Church Slavonic Typography in Unicode

Unicode Technical Note #41

Aleksandr Andreev[1] Yuri Shardt Nikita Simmons

Table of Contents

[1] This Technical Note was authored jointly under the general editorship of Aleksandr Andreev (St. Petersburg Theological Academy, St. Petersburg, Russia; `aleksandr.andreev@gmail.com`). Fonts and other software created by the authors for Church Slavonic typography may be downloaded from `http://sci.ponomar.net/`.

1 Introduction

Unicode is a computing standard for the consistent encoding and representation of text as expressed in a wide variety of writing systems. Among the scripts included within the Unicode Standard are several blocks of Cyrillic characters. As of version 8.0, the repertoire of Cyrillic characters (some still in the Pipeline) allows for the correct typesetting of texts from a wide timeframe written in the Church Slavonic language.

While the Unicode Standard documentation is sufficient for correct encoding, certain technical difficulties may arise when working with actual texts. A standard methodology for addressing such difficulties is necessary so that Church Slavonic typography can be portable and flexible; hence, the need for the present document. This Unicode Technical Note is not a supplement *in addition to* the Unicode Standard but rather represents a consensus on how Church Slavonic typography can be implemented *within* Unicode. The next three subsections explain this in detail. Following these sections, the remainder of the document discusses the necessary repertoire of characters, principles of font design and correct typography, as well as the correct handling of collation, input methods, numerals, and transliteration for Church Slavonic texts.

1.1 What is Church Slavonic?

There is some overlap in the use of the terms "Old Church Slavonic", "Church Slavonic", "Old Slavonic", etc. As defined in this document, Church Slavonic (also called "Church Slavic") is a highly codified, literary language used by the Slavs. In the earliest days of Slavic literature, Church Slavonic functioned as a high literary language, existing alongside with vernacular dialects like Russian (Uspensky, 1987, pp. 14-20). Presently, the Church Slavonic language is used only as a liturgical language by the Russian Orthodox Church, as well as by other local Orthodox Churches, Russian Old Ritualist communities and Greek Catholic churches in Croatia, Slovakia, and Ukraine.

The purpose of this document is not to discuss the syntax or morphology of Church Slavonic, or its differences from Russian and other modern Slavic languages.[2] Rather, we shall focus here solely on issues associated with Church Slavonic typography. We define Church Slavonic typography broadly to mean the process of representing and manipulating Church Slavonic characters using the computer. Specifically, three distinct purposes can be identified, the first two of which are typography *par excellence*: the typesetting (layout) of modern Church Slavonic texts used in the worship and theology of the Orthodox and Byzantine Catholic churches on a medium (paper or screen), which we shall term "work with modern liturgical texts"; the typesetting (layout) of historical Church Slavonic texts, either in whole or as part of academic treatises in the fields of linguistics, palæography, liturgiology, and related disciplines, which we shall term "academic work"; and the electronic storage and associated search and computer-aided analysis of Church Slavonic texts, which we shall term "building a Slavonic corpus." Related to these tasks, but not directly addressed in this document, are issues associated with the typesetting and storage of liturgical music in Church Slavonic, including that in Palæoslavic and Kievan Square musical notations.[3]

Most linguists agree that early Church Slavonic was written in what is now called the Glagolitic alphabet, often attributed to the brothers Sts. Cyril (Constantine) and Methodius. However, around the late 9th century in Bulgaria, Glagolitic characters were replaced by what is now called the Cyrillic alphabet, which

[2] For a general introduction to modern Church Slavonic grammar in English, see Archb. Alypy (Gamanovich).

[3] Generally speaking, the storage of music is outside the scope of encoding and requires the use of higher-level protocols (such as music notation software). Nonetheless, the characters required for Kievan (Synodal or "square") Notation have been added to the repertoire of Unicode, primarily for use in line with text (see Andreev et al. (2012)). Kievan Notation is also natively supported by the open source music engraving software LilyPond. Presently, no standardized method exists for the correct representation of the various Palæoslavic Musical Notations (chiefly Znamenny, but also Demestvenny and Put). The authors are preparing a proposal and relevant technologies for work with these notations; the details are discussed in a forthcoming paper (Andreev and Simmons, 2015).

gradually came to be used by most of the Eastern and Southern Slavs. Though some usage of Glagolitic continued in various communities in Croatia until the 20th century, and though Glagolitic manuscripts are of enormous value, the present document addresses only Cyrillic typography. The typesetting of Glagolitic texts presents a different set of challenges that are beyond the scope of this document.[4]

In order to better understand the typographical needs of the Church Slavonic language as written in the Cyrillic script, it is important to identify the distinctive eras of its development, each of which includes a different set of grammatical rules and forms, syntax, character repertoire, vocabulary, orthography, and repertoire of literature. Broadly, we identify five distinct forms of Church Slavonic script that should be discussed in this document: the earliest form of ustav writing; subsequent poluustav writing and type; Slavonic Incunabula; the Synodal Era type; and skoropis (semi-cursive). These distinct forms we shall term "recensions".[5]

Ustav Script

The earliest recension is the uncial script, also known as *ustav*. The origin of this script is clearly Greek uncial writing from about the 9th century, which was used at the time in the Greek liturgical books of the Byzantine Empire. The earliest known ustav manuscript dates to AD 933 (Karsky, 1979, p. 160). Ustav writing is characterized by the straight shape of the script. The letters are formed by straight lines, half-circles and right angles. The size of the characters is usually quite large and the spacing between characters is equal throughout. In general, one can tell that the manuscripts were written with great diligence and care, reflecting their intention for liturgical use. Ustav writing is well documented in the standard introductions to Slavonic palæography; see, for example, Karsky (1979, p. 169) and Shchepkin (1999, p. 117). Though ustav manuscripts may be found up to the 17th century, ustav writing flourishes mainly from the beginning of Church Slavonic writing in Cyrillic and up to the 14th century. Ustav typography is thus of interest for two purposes: building a Slavonic literary corpus and academic work in linguistics, palæography, liturgiology, and related disciplines.

Figure 1 presents a passage of ustav writing from the Ostromir Gospel, one of the oldest Slavonic Cyrillic manuscripts, which originated in Novgorod around AD 1056. The ability to correctly typeset ustav Slavonic script would be necessary for the proper digital storage of the text of the Ostromir Gospel and similar manuscripts.

Poluustav Script and Type

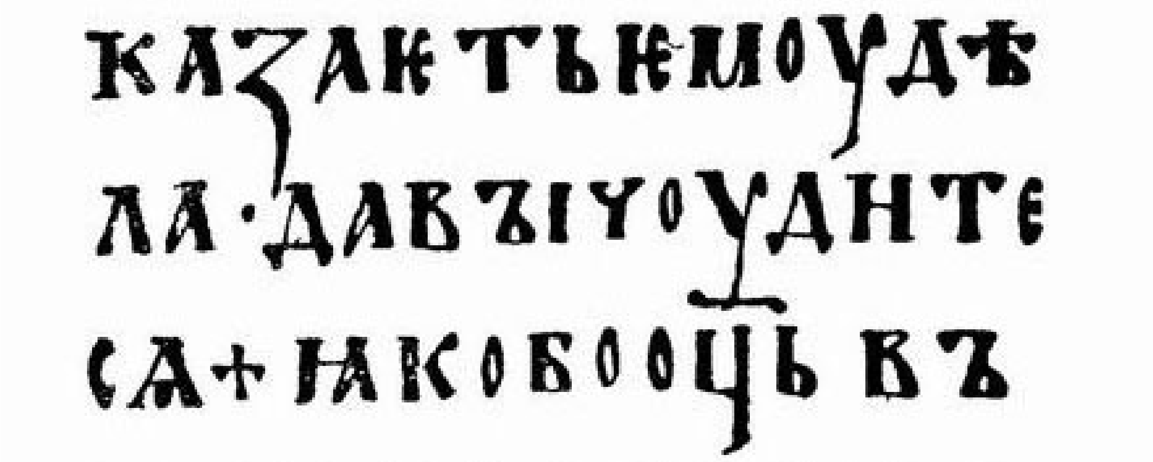
The ustav script gives way to the semi-uncial or poluustav script, which begins to flourish around the middle of the 14th century. Poluustav writing no longer reveals the kind of diligence and care used by earlier scribes – letters show some degree of sloppiness, straight lines become curved and angles are now oblique. Poluustav writing is also characterized by a greater number of abbreviations and by the emergence of accent marks, breathings and other diacritics, some of which have no linguistic meaning, but were simply adopted to imitate Greek texts.

Russian poluustav script can be broadly classified into an early and a late form, which differ in the repertoire of their characters and the style of the script. Thus, for example, early poluustav continues to use the letter ѥ (Iotified E); often uses ѹ or ү but rarely ꙋ; or, uses the ꙑ form of Yery. Later poluustav

[4] The authors provide a discussion of some of the issues in Glagolitic typography in their proposal to encode Glagolitic combining letters (see Andreev et al. (2014)).

[5] We should point out that we are using the term *recension* to indicate differences in the script. The recensions of Church Slavonic script are not the same as the recensions of the language itself. For example, modern Church Slavonic as used by the Serbian Orthodox Church could be classified as a different recension of Church Slavonic than the language used by the Russian Orthodox Church because the two "dialects" have differences in orthography and orthoepy. However, for typographical purposes they are nearly identical, and thus both classified as the "Synodal recension" [of writing].

Figure 1: Example of Church Slavonic ustav writing. Source: *Ostromir Gospel*, Novgorod, Russia, *c.* 1056.


КАꙀАѤТЬѤМОУДѢ
ЛА·ДАВЪІЧОУДИТЕ
СѦ+ꙖКОБООЦЬ ВЪ

is characterized by the use of only the є , ꙋ, and ꙑ forms, as well as by a preponderance of diacritical marks, many borrowed from Greek. Nonetheless, despite some evolution in the repertoire of characters, all of these forms constitute a distinct poluustav recension of writing (Karsky, 1979, p. 172). Poluustav handwriting continued to flourish until the late 17th century, at which time it gave way to a handwritten form which was more "blockish" or squared-off, which may still be found in Old Ritualist manuscripts.

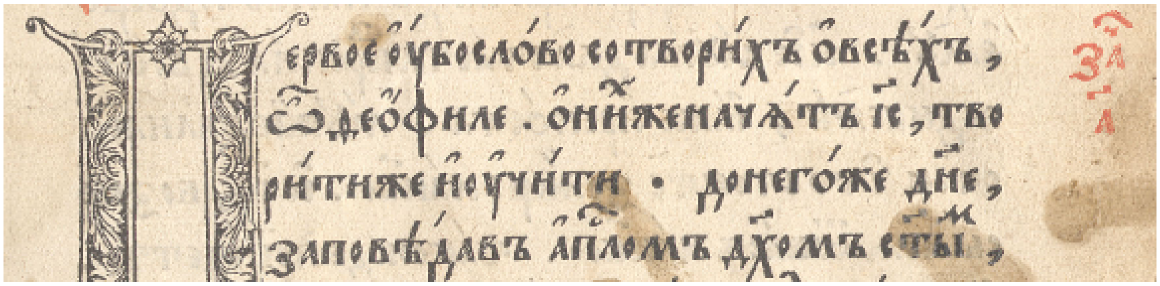
The advent of the printing press brought about both standardization and diversity to the field of Slavonic typography. The fonts designed for the printing presses in Russia and in the Polish-Lithuanian Commonwealth in the 16th and 17th centuries were modeled on poluustav script. This is the script that, for example, appears in the famous *Apostol* of Ivan Fedorov (printed in 1563), one of the earliest books printed in Moscow, and presented in Figure 2. While the poluustav typeface used by Ivan Fedorov in his publications became the standard letter-form for most printed books, other variant typefaces emerged as well, which were appropriate for different types of books, such as Altar Gospels, standard service books for parish use, private prayer books, collections of homilies, and the Bible.

Nonetheless, the form of poluustav type used by Ivan Fedorov in the *Apostol* became more or less the standard for typed texts in 17th century Moscow. A poluustav print tradition used by the Moscow Print Yard (*Pechatny Dvor*) thus emerged, and was used for the production of liturgical books in Russia under the first five Patriarchs of Moscow (late 16th – mid 17th century). Of these, especially important are the books printed under Patriarchs Joasaph I (1634–1641) and Joseph (1642–1652), because their typeface and typographical and orthographic conventions continue to be largely imitated by the Russian Old Ritualist communities even up until today.[6]

In addition to the books printed in Moscow, book printing was also active in the Polish-Lithuanian Commonwealth, and included the production of a number of extremely important texts such as the Ostrog Bible (printed by Ivan Fedorov in 1581); the Trebnik (Euchologion) (1646, see Figure 3) of Metropolitan Peter (Mogila); and the many other titles produced by presses in Lwów, Wilno, Zabłudów, Ostróg and Stratyn. Because book printing in the Polish-Lithuanian Commonwealth was decentralized and largely

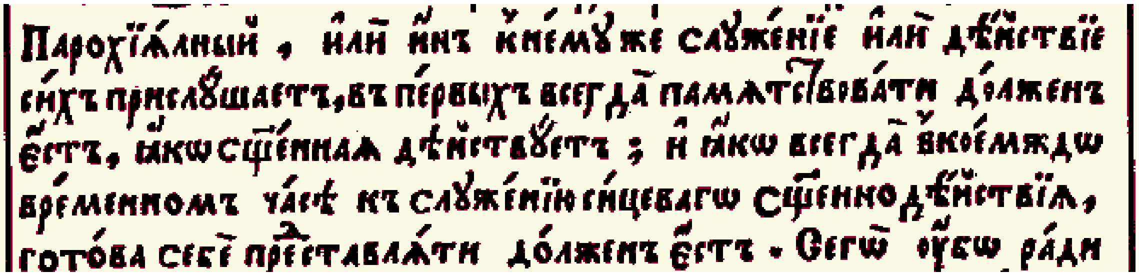
[6] Old Ritualists (sometimes called Old Believers) did not accept the liturgical and philological reforms under Patriarch Nikon (1652–1666) and Patriarch Joasaph II (1667–1672), and continue to use the liturgical texts printed in the Russian Orthodox Church prior to these reforms. In the Russian Empire, book printing by Old Ritualists was illegal, and thus only resumed in 1905; since then, some reprint editions have appeared. In addition, some Old Ritualists, called Yedinovertsy (adherents of Yedinoverie, literally "the one faith movement") reunited with the mainline Russian Orthodox Church, but continue to use the pre-reformed liturgical practices; they were able to print new books, which, however, largely imitated the typographic traditions of the poluustav era. A comprehensive study of the content and context of the reforms themselves, and their impact on the Church Slavonic language and typography, is yet to be undertaken; for an overview, see the introduction section and bibliography in Krylov (2009).

Figure 2: Example of Church Slavonic poluustav type. Source: *Apostol* of Ivan Fedorov, Moscow, 1563.



Первое ѹбо слово сотворихъ ѡ всѣхъ,
ѡ Деофиле. о нихже начатъ Іс҃, тво
рити же и ѹчити. донегоже дне,
заповѣдавъ апл҃омъ дх҃омъ ст҃ы,

Figure 3: Example of Church Slavonic poluustav type. Source: *Trebnik* of Metropolitan Peter (Mogila), Kiev, 1646.



Парохї́ѧлный, или́ инъ к немꙋже слꙋже́ніе или дѣ́йствіе
сихъ прислꙋшаетъ, въ пе́рвыхъ всегда̀ памѧтствова́ти до́лженъ
е́стъ, ѩ́кѡ сщ҃е́нная дѣ́йствꙋетъ; и ѩ́кѡ всегда̀ вкое́мꙋждѡ
вре́менномъ часѣ́ къ слꙋже́нію сицевагѡ сщ҃еннодѣ́йствїѧ,
гото́ва себѣ̀ предстаблѧ́ти до́лженъ е́стъ. Сегѡ̀ ѹбѡ ра́ди

unregulated by either civil or ecclesiastic authorities, it revealed a greater diversity in repertoire, typeface and orthography. However, it too is broadly classified as part of the poluustav recension. In sum, a standardized approach to poluustav typography is necessary for all three purposes of Slavonic typography – for academic work, for the creation of a corpus, and for the production of modern texts (for the use of Old Ritualists).

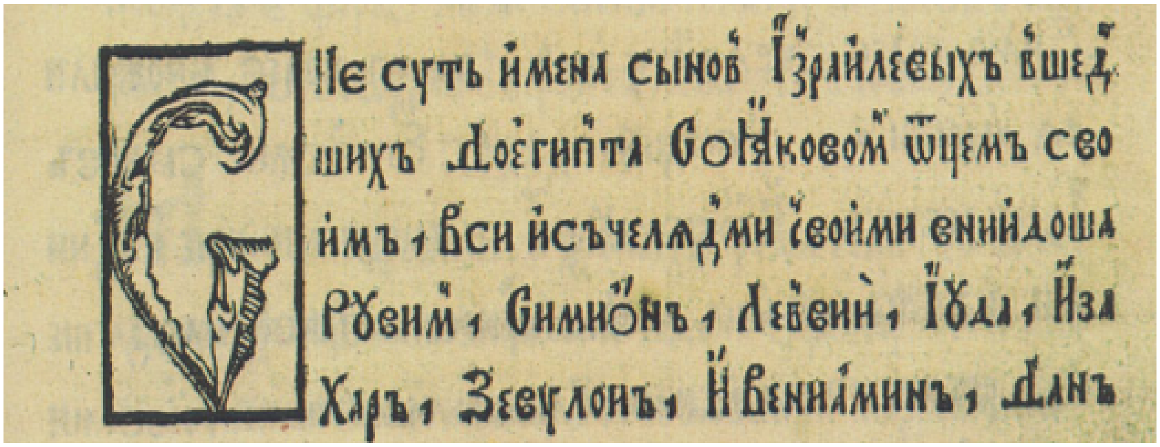
Slavonic Incunabula

While Ivan Fedorov successfully typeset Church Slavonic texts in Moscow and later in the Polish-Lithuanian Commonwealth, the first attempts at printing Church Slavonic had taken place much earlier in West Slavic and South Slavic lands. The first Church Slavonic book to be printed was the *Octoechos*, typeset by Schweipolt Fiol in Kraków in 1491 (see Figure 5). Around the same time, printing began in the Balkans, with the publication of the *Octoechos in Tone 1* in 1493 by the printing press of the Montenegrin prince Đurađ Crnojević. Fiol also printed an *Horologion* and both a Lenten and a Flowery *Triodion* while the Crnojević press produced five major titles. These printed incunabula[7] initiated a somewhat short-lived printing tradition in the South and West Slavic lands. Well known examples from this period include the books printed by Božidar Vuković, whose 1517 *Služabnik* opened the work of a Serbian press in Venice, and those by Francysk Skaryna, who between 1517 and 1519 printed the Bible in 22 volumes in Prague (see Figure 4). While Skaryna's work is not without the heavy influence of the vernacular language of his homeland (present-day Belarus), it should nonetheless be considered part of the Church Slavonic literary tradition.[8]

[7] An incunabulum (or incunable) is a book, pamphlet, or broadside printed before the year 1501 in Europe. We use the term incunabulum a bit more broadly to apply to the books printed in South and West Slavia up to the mid-16th century, since, in our opinion, they form a distinct and unique printing tradition.

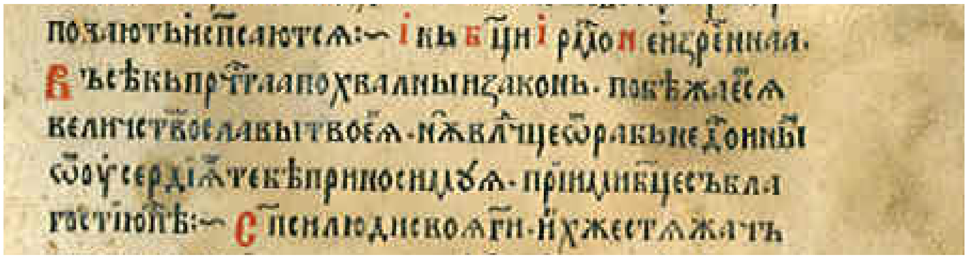
[8] In this paper, we have provided only a cursory look at the history of Slavonic typography; the interested reader may consult Nemirovsky (2003) and the other works by the same author.

Figure 4: Example of West Slavic Church Slavonic type. Source: Bible of Francysk Skaryna, Prague, *c.* 1519.



Ие суть имена сынов Ізраилевыхъ вшед
шихъ до Єгипта со Іаковом ѿцемъ сво
имъ , вси исъчелядми своими внийдоша
Рѹвим , Симиѡнъ , Левгин , Іѹда , Иза
харъ , Зевѹлонъ , И Вениаминъ , Данъ

Figure 5: Example of West Slavic Church Slavonic type. Source: *Octoechos* of Schweipolt Fiol, Kraków, 1491.



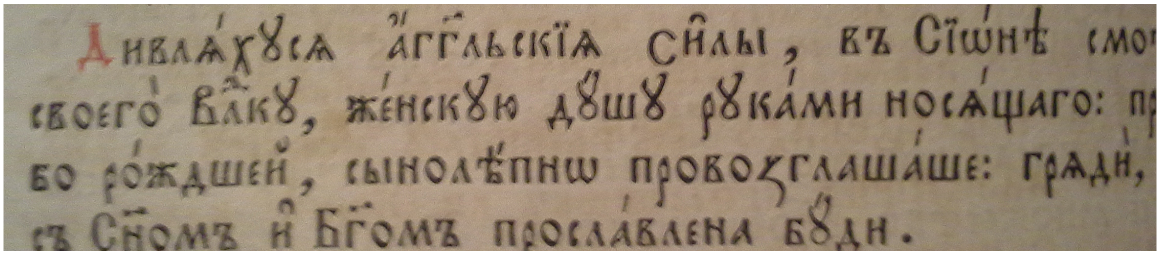
позлютьнеспсаютсѧ:~ і кь б цні рмон енеурѣнная.
Въсѣкыпрѣтлапохвалныизаконь. покѣжаѥсѧ
величьствославытвоѥѧ. нжвалѧщеѡракынезонны
ѡоусердіѧтекѣприносиндѹѧ. пріидинкцесьвла
гостинонѣ:~ Спсилюдискоагы. ихжестѧжатъ

While in the West and South Slavic Incunabula one finds a noticeable influence on the shapes of the characters from the contemporary poluustav manuscript tradition (described above), the unique shapes were primarily based on the ustav script. The letter-forms were quite crude and frequently disproportionate, since, at this infant stage of book printing, many modern typesetting conventions and techniques had not yet been introduced or standardized. A systematic analysis of the repertoire, conventions and orthography of these publications remains to be undertaken; we can only present some generalizations in the tables below. Following the introduction of poluustav letter-forms, South and West Slavic printing presses abandoned the crude Incunabula typefaces and adopted the more refined poluustav style of typography.

Synodal Slavonic

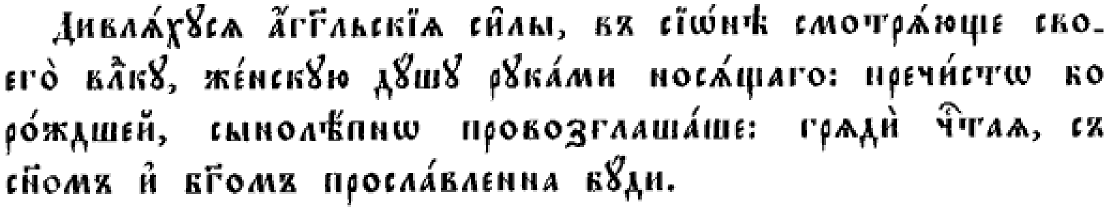
The liturgical and ritual reforms of Patriarch Nikon (culminating with the Moscow Council of 1666–1667) and his successor Patriarch Joasaph II, as well as the influence of Western European principles and methods of typography, affected the tradition of Slavonic typography and led to the emergence of a highly codified and standardized tradition of Church Slavonic – the Synodal recension. This recension is so named after the period in the history of the Russian Orthodox Church (1721–1917) during which it was governed by a body called the "Holy Governing Synod", which operated also the Synodal presses in Moscow and St. Petersburg. The Synodal recension is distinguished by the highly standardized nature of its orthography and repertoire of diacritical marks and combining letters, as well as by the distinctive typefaces. An example of text of this recension may be seen in Figure 7. In addition, books were also printed in Kiev by the Laura of the Kiev Caves (see Figure 6); these Kievan editions present a somewhat distinct typeface and repertoire of characters from the Moscow and St. Petersburg editions (and sometimes also contain orthographic differ-

Figure 6: Example of Synodal Church Slavonic from a Kievan edition. Source: *Menaion* for August, Laura of the Kiev Caves, Kiev, *c.* 1875.



Дивлѧ́хꙋсѧ а҆́гг҃льскїѧ си̑лы, въ Сїѡ́нѣ смо
своего̀ Влⷣкꙋ, же́нскꙋю дꙋ́шꙋ рꙋка́ми носѧ́щаго: пр
во ро́ждшей, сынолѣ́пнѡ провоꙁглаша́ше: гра́ди,
съ Сн҃омъ и҆ Бг҃омъ просла́влена бꙋ́ди.

Figure 7: The same text as in Figure 6, but from a Moscow edition. Source: *Festal Menaion*, Moscow, Synodal Publishing House, 1901.



Дивлѧ́хꙋсѧ а҆́гг҃льскїѧ си̑лы, въ сїѡ́нѣ смотрѧ́юще сво-
его̀ влⷣкꙋ, же́нскꙋю дꙋ́шꙋ рꙋка́ми носѧ́щаго: пречи́стѡ во
ро́ждшей, сынолѣ́пнѡ провозглаша́ше: градѝ чⷭ҇таѧ, съ
сн҃омъ и҆ бг҃омъ просла́вленна бꙋ́ди.

ences). In particular, the Kievan editions can be immediately identified by the use of the variant forms ꙁ (a form of з), ᲁ (a form of д), and ᲃ (a form of с).[9] Nonetheless, these editions are also classified as part of the Synodal recension. In addition, due to the influence of the Russian Empire on the Slavic world during this time period, books of the Synodal recension also came to be used by the other Slavic Orthodox Churches, notably in Serbia and Bulgaria, though some differences in orthoepy continue to be maintained to this day.

In terms of grammatical, orthographic and typographic rules, Synodal Church Slavonic, with minor variation, remains the main liturgical language of the Russian Orthodox Church today. Indeed, the liturgical books of the Russian Church have remained largely unchanged since the printing of the Elizabeth Bible. This text, a revision of previous Slavonic Bibles, was published under Empress Elizabeth in 1751, and continues to be the "authorized version" of the Russian Orthodox Church to this day. An attempted revision of the Slavonic liturgical books on the eve of the Revolution of 1917 remained mostly unnoticed, and focused anyway more on the content of the books than their typography.[10] The publishing activities of the Russian Orthodox Church ceased during the period of religious persecution and state-sponsored atheism in the early years of the Soviet Union; when they were allowed to resume, books of the Synodal period were reprinted without much critical work.[11] In terms of typeface, the Synodal type has largely become the standard; the Kievan typeface has fallen out of use and is mostly of historical value, though some originals and photocopied reprints of Kievan editions may still be found in choir lofts, especially in the Russian diaspora. Thus, the correct encoding of Synodal Church Slavonic is primarily of interest for the typesetting of modern liturgical texts, but also for computer-aided analysis of the Church Slavonic corpus.[12]

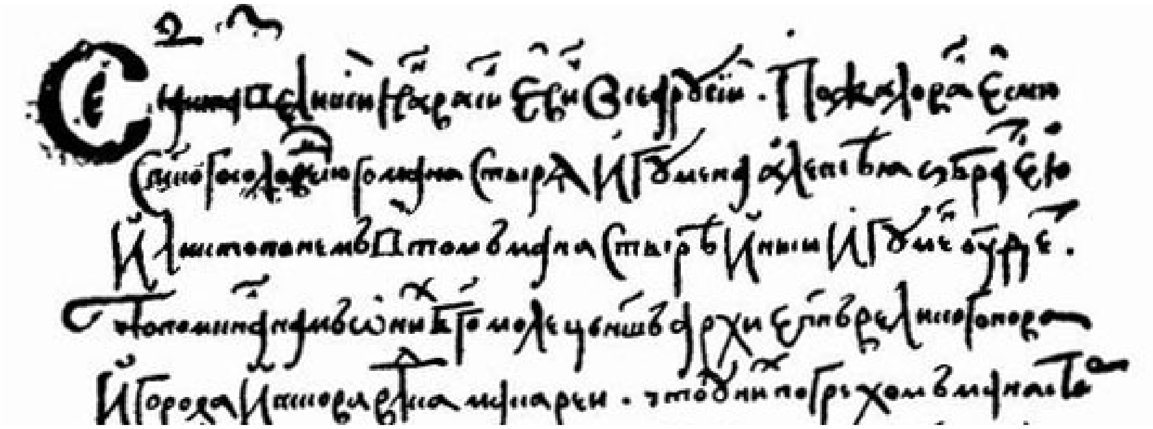
[9] For a description of the rules governing the use of these variant forms, see the Character Variants section below as well as Andreev et al. (2014).

[10] For an overview of this and other attempts to revise the liturgical books in recent history, see Sove (1970).

[11] Perhaps the only exception are the so-called *Green Menaia*, a compendium of services for every day of the year that drew on various sources, including those not published in the Synodal books. However, these liturgical texts, though in Church Slavonic, were typeset in modern Russian characters and orthography; thus, they belong to the realm of Russian, and not Church Slavonic, typography.

[12] For more on the topic of Church Slavonic book publishing in recent history, see Kravetsky and Pletneva (2001, pp. 224ff).

Figure 8: Example of Church Slavonic skoropis writing. Source: Gramata issued by Ivan IV to the Solovetski Monastery, 1539.



Skoropis

Finally, some mention ought to be made of skoropis (lit., “swift writing”), a form of Slavonic semi-cursive script that emerges around the same time as poluustav and spreads primarily to secular documents, where it can be found up through the 18th century, at which point it came to be replaced by vernacular cursive writing. As the name implies, the purpose of skoropis was to allow scribes to write quickly, a technique of key importance to many government functions, where speed of communication rather than aesthetic feel and adherence to tradition were of primary value. As can be seen from Figure 8, skoropis is characterized by wide strokes of the feather, the rounded shape of its letters, and the presence of many combining superscripts. Strictly speaking, skoropis primarily reflects the vernacular (Russian, Ukrainian, etc.) written tradition, as secular documents were rarely produced in the high, literary Church Slavonic. However, we do also find the script used in certain ecclesiastical works, primarily in collections of patristic homilies, didactic material, and scriptural commentaries, rather than liturgical texts. The forms of skoropis are diverse, and depend on time period, provenance of the manuscript, and scribal traditions; for the purposes of constructing the charts below, we select a font that mimics the skoropis tradition of Moscow around the time of the reign of Peter I, but this is intended to be purely demonstrative in value.

Broadly, then, we have identified five distinct forms of Church Slavonic script or type that should be discussed in this document: the earliest form of ustav writing; subsequent poluustav writing and type; South and West Slavic Incunabula; the Synodal poluustav type; and skoropis. Below, we will discuss the repertoire associated with each recension. First, however, a few words are warranted about why a common standard is necessary and how standardization is achieved by Unicode.

1.2 The Unicode Standard

As long as one is working in a completely closed system – that is, as long as the correct representation of a document is not of interest beyond the original user and the life expectancy of the software and hardware used to produce it – Unicode is not necessary. Indeed, in a closed environment, one can use any *ad hoc* codepage. However, as soon as documents become available for exchange over multiple systems and software implementations, the existence of a unified standard for the encoding of Church Slavonic text from all epochs becomes essential. The typesetting of any text, either in academic work or in modern liturgical editions, is conditional on one additional factor: the presence of adequate Church Slavonic fonts to represent the typeface or script of the given epoch. However, the design and development of such fonts again necessarily requires a common standard. Fonts that conform to the requirements of the Unicode standard we shall call “conformant fonts.”

In the past, attempts to create a common standard for Church Slavonic text on the basis of existing 8-bit encodings had been put forth. One such attempt that became quite popular is the Universal Church Slavonic (UCS), an approach that uses an 8-bit encoding and an *ad hoc* codepage. Since 8-bit encodings are limited to 255 characters, UCS is only designed for working with Synodal Church Slavonic, though it may be extended to represent texts of other Slavonic recensions by using a 16-bit encoding and extending the codepage.

The second approach is the use of an *ad hoc* markup language along side an 8-bit codepage. The most popular of these approaches is the Hyperinvariant Presentation (HIP) format, which uses various markup codes to represent Church Slavonic characters within the Windows 1251 codepage. The approach certainly offers considerable flexibility, since the use of markup allows both encoding and formatting to be handled at the same level. However, the term “hyperinvariant” is a misnomer: while the text is certainly invariant for storage purposes, it cannot be used for typesetting without a converter into some other standard, for example, into Universal Church Slavonic or into Unicode.

In fact, both HIP and UCS are attempts to use the CP 1251 to do something which it was not intended to do, namely, to encode Church Slavonic text. Neither of these attempts is a standard for encoding Church Slavonic, since neither is regulated by any national or international standards body and neither is supported by default in software. These approaches – though clever and functional solutions to an existing problem – are simply *ad hoc* agreements between a limited set of users. There is no compelling reason for software manufacturers to support either of these schemes, or any schemes of this type. This state of affairs leads to a number of problems. The first of these is the well-known problem of the dependence on fonts: in the absence of a specific font (and sometimes an additional add-on or extension program) the user sees nonsense characters. But even in the presence of adequate fonts, such private agreements cannot be used as a reliable data storage method since they lack any guarantee of stability and since the continued existence of the agreement and fonts that conform to it depends solely on the good will of a limited number of developers. In addition, it is difficult to use these approaches for working with multilingual texts, since distinctions between writing systems have to be made not at the encoding level but at the level of fonts. Finally, since most modern computing software in many contexts interprets a text stream as Unicode, and since Unicode provides character properties in addition to encoding, these private approaches lead to a number of curious failures when the character properties indicated in Unicode are not appropriate for the characters mapped in the private agreement.[13]

Unicode has quickly become the industry-wide standard for encoding text. In addition to the fact that it and the parallel ISO/IEC 10646 are maintained by international standardization bodies and supported by software manufacturers, Unicode also offers a number of advantages for users engaged in Church Slavonic typography. First, unlike with private agreements, the user is able to work with Slavonic text on any platform and in any software, without special tools, add-ons, converters, and the like (but not without fonts and standard support for modern font features). This also allows the user to work with the text in its immediate form and to perform computer based analysis of Slavonic texts, such as querying, parsing, and the like. Second, the Unicode Standard is sufficient for representing the full repertoire of Church Slavonic text from all of the different recensions and sources. Moreover, the same character used in different recensions is encoded identically. Third, since Unicode is stable, it is a natural solution for storing textual data.

[13] For example, suppose that a certain “private standard” for Church Slavonic text maps the acute accent to the codepoint used in Unicode to encode the numeral 1. Most modern computing software, when tasked with collation, line breaking and similar operations, will treat this character as a numeral, no matter what grapheme has been mapped to this codepoint in the font. This will lead to unexpected results for Church Slavonic text, where the user expects the character to be interpreted as a diacritical mark. This example demonstrates that all “private standards” should be mapped in the Private Use Area, not in already allocated sections of the Unicode codespace.

Finally, since Unicode encodes all of the world's writing systems, it makes it trivially simple to work with multilingual documents, which include data in Church Slavonic and in other languages.

Unicode assigns a unique codepoint – a number – to any given character while leaving the graphical representation of this character to higher level protocols (fonts and rendering systems). This abstraction allows for the proper unification of characters across recensions of the textual tradition while preserving the ability to make graphical distinctions at the font level. Thus, all characters used in the Cyrillic writing system have been unified not only across the various recensions of Church Slavonic but also across all languages – both Slavic and non-Slavic – that use the Cyrillic script, and have been encoded as one coherent system. We should, however, note what Unicode *does not* do – namely, it does not address issues of typography beyond encoding and certain properties governing the interaction of characters. In particular, the Unicode standard does not provide specifications dictating glyph shaping, presentation, and formatting. The present document demonstrates how correct glyph shaping and text flow can be achieved for Church Slavonic with a combination of Unicode, modern font technologies, and conformant software.

To explain the approach to Cyrillic that has been taken in the Unicode standard, we point out a past attempt to standardize the encoding of Church Slavonic in Unicode. Kostić et al. (2009) presented a proposal to encode an entirely new block in the Unicode standard – an Old Slavonic Cyrillic script. The proposed standard would have included all required characters for Church Slavonic of the ustav and South Slavic Incunabula recensions (including those with analogs in modern Cyrillic languages); two versions of each combining superscript letter, one with a titlo and one without; precomposed Cyrillic numerals; and required diacritical marks. In addition, the authors contemplated the idea of encoding some 200 ligatures. Elsewhere, Kostić (2009) raises the following justification for the encoding of such an addition to the Unicode Standard: difficulties associated with the correct placement of diacritical marks over base letters; differences in the meaning of characters used in Church Slavonic and modern Slavic languages; the proliferation of combining marks and ligatures in Church Slavonic texts; and difficulties associated with the correct implementation of sorting (collation), given the somewhat disorganized nature of the current Cyrillic blocks of Unicode.

Given the Unicode Standard's approach to Cyrillic, however, such a radical revision of the support for Church Slavonic as had been proposed by Kostić et al. (2009) is entirely unnecessary. First, there is no reason to encode in the standard certain characters that, while perhaps theoretically possible, have no attestation (for example, a combining Iotified Yat). If these theoretical characters are necessary for some users, they – being non-standard – can be encoded by font developers in the Private Use Area (PUA). Second, the Unicode Technical Committee has adopted a policy against encoding precomposed glyphs and ligatures, except for such precomposed combinations as had been encoded previously in various national codepages. There is a different mechanism for working with ligatures within Unicode and we discuss it in its turn. Finally, we believe that disjoining historical Cyrillic (as used in Church Slavonic) from modern Cyrillic on the basis of graphical appearance goes against Unicode principles – Unicode encodes characters, not graphemes – and thus would simply lead to confusion. Finally, advances in modern font technology allow us to address the issues of diacritical mark positioning and ligature composition without the use of precomposed glyphs. Indeed, modern OpenType and SIL Graphite technologies have been successfully used for complex writing systems such as Thai, Arabic, and Devanagari; they can be used just as well for Church Slavonic Cyrillic. Issues of collation also may be addressed quite simply by an appropriate tailoring of the Default Unicode Collation Element Table (DUCET).

1.3 Guidelines of This Technical Note

As we have stated above, the Unicode Standard addresses only the repertoire of characters and their mapping to a numerical code. It does not specify how these characters shall be used or which set of characters shall be used for which language, beyond dividing characters into script-specific blocks. While the Unicode documentation does provide some additional information and the annotations to characters provide some comments, these comments are usually insufficient for a user to be able to correctly and uniquely implement a language and script using Unicode.[14] In general, then, the purpose of the present Technical Note is to document a unified encoding of Church Slavonic of all recensions within the Unicode standard. Thus, it presents documentation on how a specific language should be correctly typeset in Unicode. To do this, the document specifies two sets of rules: first, which codepoints of the Cyrillic and other blocks shall be used to encode which characters and, second, which font features shall be used to represent these characters visually.

2 Repertoire Identification

At present, the Cyrillic characters required for Church Slavonic are encoded in Unicode in various non-contiguous blocks in the BMP (Basic Multilingual Plane) and SMP (Supplemental Multilingual Plane). This makes them difficult to locate in the codecharts and documentation simply on the basis of name and graphical appearance. We present the characters in an alphabetic or systematical order, with an indication to the codepoint(s) used for their proper encoding.

It should also be noted that the names for characters used in the Unicode Standard in some instances are based on the names of modern characters used for Russian or other languages written with the Cyrillic script. In other instances, they may be based on names used for historical characters in the Balkan tradition, which may be at variance with accepted names used in Russian sources. As well, some characters come with apparent doppelgängers – that is, visually indistinguishable forms – used in the Cyrillic-based writing systems of non-Slavic languages such as Kazakh, Sakha, Tatar, or Mongolian. In some instances modern and historical characters have been mapped to the same codepoint while in other instances they have been disjoined and the codepoint for the modern character should not be used to encode the historical character (and vice versa). We have provided notes with comments on some difficult issues. In the subsequent section, titled Implementation, we discuss many of the encoding issues in greater detail. In a following section, titled Font Development, we discuss how these encoding issues should be implemented in fonts.

2.1 Church Slavonic Letters

Note: the default modern representations presented in the second column from the left are intended for demonstration purposes (they take the form provided in the Unicode codecharts), while the authentic major period representations are given in the columns on the right.

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
0410	А	Cyrillic Capital Letter A	азъ	Аа	Аа	Аа	Аа	Аа
0430	а	Cyrillic Small Letter A						
0411	Б	Cyrillic Capital Letter Be	буки	Бб	Бб	Бб	Бб	Бб
0431	б	Cyrillic Small Letter Be						
0412	В	Cyrillic Capital Letter Ve	вѣди	Вв	Вв	Вв	Вв	Вв
0432	в	Cyrillic Small Letter Ve						

Continued on next page

[14] Take, for example, the character at U+0456, Cyrillic small letter Byelorussian-Ukrainian i. From the name of the character, the user knows that this codepoint is used to represent the letter i in Ukrainian and Byelorussian. Under Annotations, the user sees the comment “Old Cyrillic i.” However, it is unclear if “Old Cyrillic” here refers to Church Slavonic or to Russian in traditional orthography; in addition, if it refers indeed to Church Slavonic, it is not immediately clear if “Old Cyrillic i” should represent the dotless form, the dotted form, or the double-dotted form of the Church Slavonic letter.

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
(1C80)	ᲀ	Cyrillic Small Letter Rounded Ve	-	-	ᲀ	ᲀ	-	-
	This is a variant form of the lowercase Vedi, which appears in the Incunabula and poluustav print editions. This character does not have an uppercase version but maps to U+0412. It has been proposed for inclusion in a future version of the Unicode Standard. See the section on Character Variants for more information.							
0413 0433	Г г	Cyrillic Capital Letter Ge Cyrillic Small Letter Ge	глаголь	Гг	Гг	Гг	Гг	Гг
0490 0491	Ґ ґ	Cyrillic Capital Letter Ghe with Upturn Cyrillic Small Letter Ghe with Upturn	-	-	Ґґ	Ґґ	-	-
	This character is used in some Incunabula and poluustav print editions published in the Polish-Lithuanian Commonwealth to indicate the letter *г* in words of a Greek origin. It is the forerunner to the modern letter *ґ* used in Ukrainian.							
0414 0434	Д д	Cyrillic Capital Letter De Cyrillic Small Letter De	добро	Дд	Дд	Дд	Дд	Дд
(1C81)	ᲁ	Cyrillic Small Letter Long-Legged De	-	-	-	ᲁ	ᲁ	-
	This is a variant form of the lowercase Dobro, called the long-legged form, which appears in poluustav and Synodal Slavonic. This character does not have an uppercase form but maps to U+0414. It has been proposed for inclusion in a future version of the Unicode Standard. See the section on Character Variants for more information.							
A662 A663	Ꙣ ꙣ	Cyrillic Capital Letter Soft De Cyrillic Small Letter Soft De	-	Ꙣꙣ	-	-	-	-
	This is a form of the letter Dobro with palatalization, which is indicated here by the addition of the character ◌҄ , which joins with the base letter to form the ligature ꙣ. See the subsection on Palatalization.							
0415 0435	Е е	Cyrillic Capital Letter Ie Cyrillic Small Letter Ie	есть	Ее	Ее	Ее	Ее	Ее
0404 0454	Є є	Cyrillic Capital Letter Ukrainian Ie Cyrillic Small Letter Ukrainian Ie	-	Єє	-	Єє	Єє	Єє
	This is the wide form of the letter Yest. In Synodal Slavonic, it appears as the initial form of yest; sometimes it is also used in the medial form as a variant to distinguish different grammatical forms. The capitalized form of the initial yest should be U+0404 and not U+0415. However, at least for Synodal Slavonic, the form U+0404 should look exactly the same as the form U+0415. This distinction is required to properly implement capitalization functions in computer software.							
042D 044D	Э э	Cyrillic Capital Letter E Cyrillic Small Letter E	э	Ээ	-	Ээ	Ээ	Ээ
	In the ustav period, this character was a rare form of the letter “Yest”, which was probably a calligraphic variation, but it most certainly cannot be treated as a separate letter of the alphabet. In the poluustav period it is used in musical manuscripts of the Put and Demestvenny chant systems. It reappears in Russia during the reforms of Peter I as part of the Russian alphabet. It is also used to typeset liturgical texts in the Sakha (Yakut) language, and so should be included in Synodal recension fonts.							
0416 0436	Ж ж	Cyrillic Capital Letter Zhe Cyrillic Small Letter Zhe	живѣте	Жж	Жж	Жж	Жж	Жж
0405 0455	Ѕ ѕ	Cyrillic Capital Letter Dze Cyrillic Small Letter Dze	зѣло	Ѕѕ	Ѕѕ	Ѕѕ	Ѕѕ	-
A642 A643	Ꙃ ꙃ	Cyrillic Capital Letter Dzelo Cyrillic Small Letter Dzelo	-	Ꙃꙃ	-	-	-	-
	This variant of Zelo exists only in the ustav tradition. The Unicode naming is somewhat unfortunate. Note that this codepoint should not be used for the Zemlya variant (U+A640 / U+A641).							

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
A644 A645	Ꙅ ꙅ	Cyrillic Capital Letter Reversed Dze Cyrillic Small Letter Reversed Dze	-	Ꙅꙅ	-	-	-	-
	This variant of Zelo exists only in the ustav tradition. The Unicode naming is somewhat unfortunate.							
0417 0437	З з	Cyrillic Capital Letter Ze Cyrillic Small Letter Ze	земля	-	-	Зз	Зз	Зз
A640 A641	Ꙁ ꙁ	Cyrillic Capital Letter Zemlya Cyrillic Small Letter Zemlya	земля	Ꙁꙁ	Ꙁꙁ	Ꙁꙁ	Ꙁꙁ	-
	This tailed form of the letter Zemlya is the dominant form in ustav manuscripts. It also appears in poluustav documents, where it coexists with the modern form (U+0437). Following the reforms under Patriarch Nikon, this character started to disappear from usage. In Synodal Slavonic, it occurs only in sources of a Kievan origin, where the tailed form is the dominant form and the modern form is rare. The naming of this character in Unicode is somewhat unfortunate given that both this character and the modern form U+0437 are two variants for the same letter.							
0418 0438	И и	Cyrillic Capital Letter I Cyrillic Small Letter I	иже	Ии	Ии	Ии	Ии	Ии
0419 0439	Й й	Cyrillic Capital Letter Short I Cyrillic Small Letter Short I	иже краткая	-	-	Йй	Йй	Йй
		canonical decomp.: 0418 0306 / 0438 0306						
	This character is not used in ustav manuscripts. In poluustav sources it is decomposable as the letter izhe (U+0418 / U+0438) and the character Combining Breve (U+0306); conformant fonts should honor this canonical decomposition.							
0406 0456	І і	Cyrillic Capital Letter Byelorussian-Ukrainian I Cyrillic Small Letter Byelorussian-Ukrainian I	и	Іı	Іı	Іı	Іı	Іı
	These codepoints should be used for the base form of the letter ı. In Church Slavonic fonts, the character at this codepoint should be **dotless**. In Synodal Church Slavonic, as a standalone, it is used only in numerals to denote the number 10. It also serves as the base for the placement of diacritical marks over the ı. Thus, Small I with Acute Accent ı́ should be implemented as U+0456 U+0301, **not** as U+0457 U+0301. In Synodal Church Slavonic, if no diacritical mark occurs over the base form (except when it is used as a numeral), two dots are placed over this letter. This is encoded explicitly using the codepoints U+0407 / U+0457, or the equivalent decompositions. (See below). A single-dotted form does not exist in modern Church Slavonic; however, it may occur in Incunabula (particularly in the editions of Schweipolt Fiol, see Nemirovsky (2003, p. 325)). When used in Church Slavonic, this single-dotted form may be encoded explicitly as U+0456 U+0307 (which becomes ı̇). Neither the Iota (U+A647 / U+A646) nor the Cyrillic Letter Palochka (U+04C0 / U+04CF) nor the Latin Letter Dotless I (U+0131) should be used to encode the dotless i of Church Slavonic. In modern Ukrainian and Byelorussian, as well as in Russian written in pre-reform orthography, the character encoded at U+0456 is by default single-dotted. When typesetting Ukrainian or Byelorussian texts using a stylized Slavonic font (as is often done in advertisements and similar settings), both the single-dotted and the dotless variants of the character may be mapped to the same codepoint using the *locl* (Localized Forms) feature in OpenType fonts.							
0407 0457	Ї ї	Cyrillic Capital Letter Yi Cyrillic Small Letter Yi	-	Її	ї	Її	Її	Її
		Canonical decomp.: 0406 0308 / 0456 0308						
	This is the double-dotted form of the letter ı, which is the form used in modern Church Slavonic when no diacritical mark is placed over ı. The two dots are the diacritical mark *kendema* (see below, on diacritical marks) and are encoded in this case as U+0308. Thus, the double-dotted i is encoded as U+0456 U+0308, which is equivalent to U+0457 (the capital form being U+0406 U+0308, equivalent to U+0407). Conformant fonts should implement this character in a way that honors the decomposition. Note that **only** U+0456 should be used as the base form for accent placement. Thus, no accents should be placed over the double-dotted form.							
A646 A647	Ꙇ ꙇ	Cyrillic Capital Letter Iota Cyrillic Small Letter Iota	-	Ꙇꙇ	-	-	-	-

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
	This is a modern academic invention for use in transliterating Glagolitic texts, used to represent the Glagolitic letter *initial izhe* (ⰹ).							
A648	Ꙉ	Cyrillic Capital Letter Djerv	джервь	Ꙉꙉ	Ꙉꙉ	-	-	-
A649	ꙉ	Cyrillic Small Letter Djerv						
	The letter *djerv* occurs in ustav manuscripts of a Balkan provenance, where it is used to record the palatal affricate *dj*. It is also used in academic work to transcribe the Glagolitic letter Ⰼ (Ivanova, 1997, p. 24). Note that this codepoint should not be used to encode the modern characters dzhe (ђ, U+0452) or tshe (ћ, U+045B) used in Serbian. Likewise, the codepoints of the modern characters should not be used to encode the archaic *djerv*.							
041A	К	Cyrillic Capital Letter Ka	како	Кк	Кк	Кк	Кк	Кк
043A	к	Cyrillic Small Letter Ka						
041B	Л	Cyrillic Capital Letter El	людіе	Лл	Лл	Лл	Лл	Лл
043B	л	Cyrillic Small Letter El						
A664	Ꙥ	Cyrillic Capital Letter Soft El	-	Ꙥꙥ	-	-	-	-
A665	ꙥ	Cyrillic Small Letter Soft El						
	This is the form of the letter Lyudie with palatalization used in some ustav manuscripts. Palatalization is indicated here by adding the mark ◌҄ , which combines with the character forming the ligature ꙥ. See the discussion of Palatalization, below.							
041C	М	Cyrillic Capital Letter Em	мыслѣте	Мм	Мм	Мм	Мм	Мм
043C	м	Cyrillic Small Letter Em						
A666	Ꙧ	Cyrillic Capital Letter Soft Em	-	Ꙧꙧ	-	-	-	-
A667	ꙧ	Cyrillic Small Letter Soft Em						
	This is the form of the letter Myslete with palatalization used in some ustav manuscripts. Palatalization is indicated here by adding the mark ◌҄ , which combines with the character forming the ligature ꙧ. See the discussion of Palatalization, below.							
041D	Н	Cyrillic Capital Letter En	нашъ	Нн	Нн	Нн	Нн	Нн
043D	н	Cyrillic Small Letter En						
04A4	Ҥ	Cyrillic Capital Ligature En Ghe	-	Ҥҥ	-	-	-	-
04A5	ҥ	Cyrillic Small Ligature En Ghe						
	This character has two usages in the history of Cyrillic scripts, which are not linguistically equivalent, but have been encoded using the same codepoint in Unicode. In Church Slavonic, this character represents the letter Nash with palatalization. The palatalization is indicated here by adding the mark ◌҄ , which combines with the character, forming the ligature ҥ. The same codepoint is also used to encode the ligature En Ghe, which occurs in the Altay, Mari, and Sakha (Yakut) alphabets.							
041E	О	Cyrillic Capital Letter O	онъ	Оо	Оо	Оо	Оо	Оо
043E	о	Cyrillic Small Letter O						
047A	Ѻ	Cyrillic Capital Letter Round Omega	онъ широкій	Ѻѻ	Ѻѻ	Ѻѻ	Ѻѻ	-
047B	ѻ	Cyrillic Small Letter Round Omega						
	This is the wide variant of the letter On, usually occurring in the initial position. In Synodal Slavonic, it may also occur in the medial position as the first letter of a compound word. The name of this character in Unicode is unfortunate, as it is clearly not an Omega but an Omicron.							
(1C82)	ᲂ	Cyrillic Small Letter Narrow O	-	ᲂ	ᲂ	ᲂ	ᲂ	-
	This is the narrow form of the letter On, which occurs mostly in poluustav manuscripts and printed texts. In old printed editions, this form accounts for about half of all occurrences of the letter On. Often, it is used when the letter On occurs without a diacritical mark – accent or breathing – and the middle form (encoded as U+043E) is used with an accent or breathing. However, there are frequent exceptions to this rule and so the usage cannot be predicted algorithmically. In modern Synodal Slavonic, this character occurs only as the first part of the digraph ᲂу. The character has been proposed for encoding in a future version of the Unicode Standard. It has no uppercase form, but cases to U+041E. See also the section on Variants.							

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
	In the manuscript tradition, including the scribal tradition used in the Slavonic Bible translated under the direction of Archbishop Gennadius of Novgorod in 1499, there are some “illustrative” or “pictograph” variations of the letter On. (These were undoubtedly the whimsical inventions of bored scribes who needed to have an occasional amusing moment to relieve the tedium of their task.) Some of the earliest printed editions of the Gospels and Epistles, as well as the Lenten Triodion and the life of St. Stephen of Perm, included these pictographs. Some may also occur in early Incunabula. Their usage is specific for a single word, including its grammatical case endings. They disappear entirely from modern Church Slavonic.							
A668	Ꙩ	Cyrillic Capital Letter Monocular O	-	Ꙩꙩ	Ꙩꙩ	Ꙩꙩ	-	-
A669	ꙩ	Cyrillic Small Letter Monocular O						
	This monocular form is used in writing the word “eye” (ꙩко). It may also be used in the oblique cases, but only in the singular number.							
A66A	Ꙫ	Cyrillic Capital Letter Binocular O	-	Ꙫꙫ	Ꙫꙫ	Ꙫꙫ	-	-
A66B	ꙫ	Cyrillic Small Letter Binocular O						
	This binocular form is used to write the word “[two] eyes” in the dual number (ꙫчеса). (Also see below.)							
A66C	Ꙭ	Cyrillic Capital Letter Double Monocular O	-	Ꙭꙭ	Ꙭꙭ	Ꙭꙭ	-	-
A66D	ꙭ	Cyrillic Small Letter Double Monocular O						
	As in the above case, this “doubled” form is used in writing the word “[two] eyes” in the dual (ꙭчеса). This form is more widespread than the above character.							
A66E	ꙮ	Cyrillic Letter Multiocular O	-	ꙮ	-	ꙮ	-	-
	This many-eyed form is used only in writing the phrase “many-eyed seraphim” (многоꙮчитїи серафімы). It has no uppercase form in Unicode.							
A698	Ꚙ	Cyrillic Capital Letter Double O	-	Ꚙꚙ	-	Ꚙꚙ	-	-
A699	ꚙ	Cyrillic Small Letter Double O						
	This “twinned” letter form appears in two different words: the initial letter of the word ꚙбо (both), and in the word двꚙе (two), as well as their various grammatical forms; it also is used in the compound words for the number 12: ꚙбанадесѧть and двꚙюнадесѧть. It was encoded in Unicode 7.0.							
A69A	Ꚛ	Cyrillic Capital Letter Crossed O	-	Ꚛꚛ	Ꚛꚛ	Ꚛꚛ	-	-
A69B	ꚛ	Cyrillic Small Letter Crossed O						
	This “crossed” form is used only in the preposition ꚛкрест, which means “around, round-about”. It has been included in Unicode 7.0.							
041F	П	Cyrillic Capital Letter Pe	покой	Пп	Пп	Пп	Пп	Пп
043F	п	Cyrillic Small Letter Pe						
0420	Р	Cyrillic Capital Letter Er	рцы	Рр	Рр	Рр	Рр	Рр
0440	р	Cyrillic Small Letter Er						
0421	С	Cyrillic Capital Letter Es	слово	Сс	Сс	Сс	Сс	Сс
0441	с	Cyrillic Small Letter Es						
(1C83)	ᲃ	Cyrillic Small Letter Wide Es	-	ᲃ	ᲃ	ᲃ	ᲃ	-
	This is the wide variant of the lowercase letter Slovo. It occurs in a wide variety of important texts, including the texts of Fiol, the *Trebnik* of Metropolitan Peter (Mogila) and the *Ostrog Bible*. In the Synodal period, it has a widespread use in the texts published by the Laura of the Kiev Caves. It has been proposed for inclusion in a future version of the Unicode Standard. This character has no uppercase form; it cases to U+0421. See the section on Character Variants for more information.							
0422	Т	Cyrillic Capital Letter Te	твердо	Тт	Тт	Тт	Тт	Тт
0442	т	Cyrillic Small Letter Te						
(1C84)	ᲄ	Cyrillic Small Letter Tall Te	-	ᲄ	-	ᲄ	-	-

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
	This is the tall variant of the lowercase Tverdo. It occurs in a wide variety of important texts, including the *Trebnik* of Metropolitan Peter and the *Ostrog Bible*. It has been proposed for inclusion in a future version of the Unicode Standard. This character has no uppercase form; it cases to U+0422. See the section on Character Variants for more information.							
(1C85)	ᲅ	Cyrillic Small Letter Three-Legged Te	-	-	ᲅ	ᲅ	-	-
	This is the "three-legged" variant of the lowercase Tverdo. This form is the dominant form in manuscripts and it is also used in a wide variety of printed editions. This form of Tverdo continued to be used in vernacular Russian publications up to the mid-19th century. It has been proposed for inclusion in a future version of the Unicode Standard. It has no uppercase form; it cases to U+0422. See the section on Character Variants for more information.							
041E 0443 1C82 0443	Оу оу	Cyrillic Capital Letter Uk Cyrillic Small Letter Uk	(оник)	Оу оу	Оу оу	Оу оу	Оу оу	Оу оу
	This character is a digraph of the letter On and the letter Uk (Ik). It is always pronounced as *u*, based on its usage in Byzantine Greek. This character should be entered as U+041E U+0443 in its uppercase form and as U+1C82 U+0443 in its lowercase form (rarely, if the first grapheme is a middle and not a narrow On, as U+043E U+443). **Note that**: we strongly suggest that the codepoints available for this character at U+0478 / U+0479 should not be used for any reason. In particular, U+0479 lacks a proper capitalized form. In addition, the use of a single glyph for this digraph makes it impossible to color only the first part of the digraph, as is often done in liturgical texts. We believe that the encoding of U+0478 and U+0479 in Unicode was erroneous.							
A64A A64B	Ꙋ ꙋ	Cyrillic Capital Letter Monograph Uk Cyrillic Small Letter Monograph Uk	ук (ик)	Ꙋꙋ	Ꙋꙋ	Ꙋꙋ	Ꙋꙋ	Ꙋꙋ
(1C88)	ᲈ	Cyrillic Small Letter Unblended Uk	-	-	ᲈ	ᲈ	-	-
	This is a variant form of the Uk (U+A64B), which occurs in a variety of texts printed in the Polish-Lithuanian Commonwealth. It has been proposed for inclusion in a future version of the Unicode Standard. This character has no uppercase form; it cases to U+A64A. See the section on Character Variants for more information.							
0423 0443	У у	Cyrillic Capital Letter U Cyrillic Small Letter U	ук (ик)	Уу	Уу	Уу	Уу	Уу
	This is the decomposed second element of the digraph оу. By itself, it is usually only used to indicate the numeral 400. However in some manuscripts it may be used instead of the character ꙋ.							
0424 0444	Ф ф	Cyrillic Capital Letter Ef Cyrillic Small Letter Ef	ферть	Фф	Фф	Фф	Фф	Фф
0425 0445	Х х	Cyrillic Capital Letter Ha Cyrillic Small Letter Ha	хѣръ	Хх	Хх	Хх	Хх	Хх
0460 0461	Ѡ ѡ	Cyrillic Capital Letter Omega Cyrillic Small Letter Omega	омега	Ѡѡ	Ѡѡ	Ѡѡ	Ѡѡ	Ѡѡ
A64C A64D	Ꙍ ꙍ	Cyrillic Capital Letter Broad Omega Cyrillic Small Letter Broad Omega	омега	Ꙍꙍ	-	Ꙍꙍ	Ꙍꙍ	-
	In Synodal typography, this character occurs only as the base in writing the accented form, which is the exclamation "Oh!", below. In other recensions, it may occur as a stylistic variant of the Omega, U+0460 / U+0461.							

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
047C	Ѽ	Cyrillic Capital Letter Omega With Titlo	-	Ѽѽ	-	Ѽѽ	Ѽѽ	-
047D	ѽ	Cyrillic Small Letter Omega With Titlo						
	The inclusion of this character, which is not a character at all, but a grapheme usually used to record the exclamation “Oh!” (ѽ, ѽлє), in Unicode is unfortunate. Furthermore, the name of the character is incorrect, since it does not contain a titlo at all. Rather, this character is a ligature of the Broad Omega variant and the *veliky apostrof* (see Section 3.3). Thus, it is decomposable as the base Broad Omega (U+A64C / U+A64D), and the combining characters Psili Pneumata (U+0486) and Combining Inverted Breve (U+0311). However, it was encoded in Unicode without a canonical decomposition. We suggest that, while the characters should not be decomposed, fonts and software should be aware of the spelling ambiguity and interpret the sequence U+A64C U+0486 U+0311 (and the corresponding lowercase analog) in the same manner as the standalone characters. In Poluustav texts, the exclamation “Oh!” may also be written with the Broad Omega and the standard *apostrof* (Psili Pneumata U+0486 and Combining Grave Accent U+0300) as ꙍ҆̀, ꙍ҆̀лє. This form has not been encoded in Unicode as a separate character and must be composed as needed.							
047E	Ѿ	Cyrillic Capital Letter Ot	отъ	Ѿѿ	Ѿѿ	Ѿѿ	Ѿѿ	Ѿѿ
047F	ѿ	Cyrillic Small Letter Ot						
	This character is a ligature of the Omega (U+0460 / U+0461) and the letter Tverdo (U+0422 / U+0442) used mostly for writing the preposition or prefix от (“from”, “out of”).							
0426	Ц	Cyrillic Capital Letter Tse	цы	Цц	Цц	Цц	Цц	Цц
0446	ц	Cyrillic Small Letter Tse						
A660	Ꙡ	Cyrillic Capital Letter Reversed Tse	-	Ꙡꙡ	-	-	-	-
A661	ꙡ	Cyrillic Small Letter Reversed Tse						
	According to Cleminson and Everson (2009), this character appears in “documents produced in Novgorod and its environs from the 11th to the 15th centuries. In the language of this area, the distinction between *ц* and *ч* had been eliminated, and [this character] replaces both these characters in the documents. It cannot be considered equivalent to either of them, and therefore neither can replace it in transcription.”							
0427	Ч	Cyrillic Capital Letter Che	червь	Чч	Чч	Чч	Чч	Чч
0447	ч	Cyrillic Small Letter Che						
0480	Ҁ	Cyrillic Capital Letter Koppa	коппа	Ҁҁ	-	-	-	-
0481	ҁ	Cyrillic Small Letter Koppa						
	In the earlier ustav period, this character was borrowed from Greek to indicate the numeral 90 (later replaced with the letter ч). It has never had a linguistic use in Slavonic, and has been obsolete in the Greek language since classical times. The uppercase form of this letter is completely undocumented.							
040F	Џ	Cyrillic Capital Letter Dzhe	дже	Џџ	-	-	-	-
045F	џ	Cyrillic Small Letter Dzhe						
	This letter occurs in the Romanian Cyrillic alphabet, in some sources of a Balkan provenance, and in modern Serbian (Ginkulov, 1840, pp. 2ff). This codepoint should not be used to encode the letter *ц*, even if it visually appears like џ.							
0428	Ш	Cyrillic Capital Letter Sha	ша	Шш	Шш	Шш	Шш	Шш
0448	ш	Cyrillic Small Letter Sha						
0429	Щ	Cyrillic Capital Letter Shcha	ща	Щщ	Щщ	Щщ	Щщ	Щщ
0449	щ	Cyrillic Small Letter Shcha						
042A	Ъ	Cyrillic Capital Letter Hard Sign	еръ	Ъъ	Ъъ	Ъъ	Ъъ	Ъъ
044A	ъ	Cyrillic Small Letter Hard Sign						
(1C86)	ᲆ	Cyrillic Small Letter Tall Hard Sign	-	-	-	ᲆ	-	-

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
	This is the tall variant of the Hard Sign, which appears in manuscripts as well as the poluustav print tradition. It has been proposed for inclusion in a future version of the Unicode Standard. This character has no uppercase form; it cases to U+042A. See the section on Character Variants for more information.							
A650 A651	Ꙑ ꙑ	Cyrillic Capital Letter Yeru with Back Yer Cyrillic Small Letter Yeru with Back Yer	еры	Ꙑꙑ	-	-	-	-
	This is a variant form that is used only in the manuscript tradition. As orthographic rules became standardized in Slavonic, this combination (and all other variants with the “hard sign”) was abandoned in favor of the “soft sign” (ь + dotless i) combination. (The Unicode name is bizarre.)							
042A 0418 044A 0438	ЪИ ъи	Digraph	-	ЪИъи	-	-	-	-
	This is a variant of yery that occurs in ustav manuscripts of a South Slavic provenance. It is best encoded and sorted as a digraph of the hard sign, U+042A (U+044A) and the octal i, U+0418 (U+0438).							
042A A646 044A A647	ЪꙆ ъꙇ	Digraph	-	ЪꙆъꙇ	-	-	-	-
	This is a variant of yery that is used in transcribing Glagolitic sources. It is best encoded and sorted as a digraph of the hard sign U+042A (U+044A) and the iota U+A646 (U+A647).							
042C 044C	Ь ь	Cyrillic Capital Letter Soft Sign Cyrillic Small Letter Soft Sign	ерь	Ьь	Ьь	Ьь	Ьь	Ьь
042B 044B	Ы ы	Cyrillic Capital Letter Yeru Cyrillic Small Letter Yeru	еры	Ыы	Ыы	Ыы	Ыы	Ыы
	This letter is usually called “yery”.							
042C 0418 044C 0438	ЬИ ьи	Digraph	-	ЬИьи	-	-	-	-
	This is a variant of yery that occurs in ustav manuscripts of a South Slavic provenance. It is best encoded and sorted as a digraph of the soft sign, U+042C (U+044C) and the octal i, U+0418 (U+0438).							
042C A646 044C A647	ЬꙆ ьꙇ	Digraph	-	ЬꙆьꙇ	-	-	-	-
	This is a variant of yery that is used in transcribing Glagolitic sources. It is best encoded and sorted as a digraph of the soft sign U+042C (U+044C) and the iota U+A646 (U+A647).							
A64E A64F	Ꙏ ꙏ	Cyrillic Capital Letter Neutral Yer Cyrillic Small Letter Neutral Yer	-	Ꙏꙏ	-	-	-	-
	This “undifferentiated” character is used in modern editions to reproduce texts where it is completely ambiguous which letter to use: the hard or soft sign. It seems to be a modern invention to accommodate for not being able to clearly read faded manuscripts.							
0462 0463	Ѣ ѣ	Cyrillic Capital Letter Yat Cyrillic Small Letter Yat	ять	Ѣѣ	Ѣѣ	Ѣѣ	Ѣѣ	Ѣѣ

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
(1C87)	ᲇ	Cyrillic Small Letter Tall Yat	-	-	-	ᲇ	-	-
	This is the tall variant of the letter Yat, which appears in manuscripts as well as the poluustav print tradition. It has been proposed for inclusion in a future version of the Unicode Standard. This character has no uppercase form; it cases to U+0462. See the section on Character Variants for more information.							
A652 A653	Ꙓ ꙓ	Cyrillic Capital Letter Iotified Yat Cyrillic Small Letter Iotified Yat	-	Ꙓꙓ	-	-	-	-
	This “iotified yat” is only known to occur in the Izbornik of 1073, a collection of the writings of various Church Fathers (Karsky, 1979, p. 205). Outside of the ustav era, it is completely undocumented.							
042E 044E	Ю ю	Cyrillic Capital Letter Yu Cyrillic Small Letter Yu	ю	Юю	Юю	Юю	Юю	Юю
A654 A655	Ꙕ ꙕ	Cyrillic Capital Letter Reversed Yu Cyrillic Small Letter Reversed Yu	-	Ꙕꙕ	-	-	-	-
A656 A657	Ꙗ ꙗ	Cyrillic Capital Letter Iotified A Cyrillic Small Letter Iotified A	азъ йотированный	Ꙗꙗ	Ꙗꙗ	Ꙗꙗ	Ꙗꙗ	Ꙗꙗ
	This “iotified a” is widely used in Church Slavonic to represent the sound *ja*. This codepoint should not be used to encode the modern Cyrillic letter я (U+042F / U+044F). Likewise, the codepoints of the Cyrillic Letter Ya should not be used to encode the iotified az or any of the yus characters. The modern letter я does not exist in Church Slavonic, though it may be provided in fonts to facilitate typesetting stylized Russian text.							
0464 0465	Ѥ ѥ	Cyrillic Capital Letter Iotified E Cyrillic Small Letter Iotified E	есть йотированный	Ѥѥ	-	Ѥѥ	-	-
0466 0467	Ѧ ѧ	Cyrillic Capital Letter Little Yus Cyrillic Small Letter Little Yus	юсь малый	Ѧѧ	Ѧѧ	Ѧѧ	Ѧѧ	Ѧѧ
A658 A659	Ꙙ ꙙ	Cyrillic Capital Letter Closed Little Yus Cyrillic Small Letter Closed Little Yus	-	Ꙙꙙ	-	-	-	-
046A 046B	Ѫ ѫ	Cyrillic Capital Letter Big Yus Cyrillic Small Letter Big Yus	юсь большой	Ѫѫ	Ѫѫ	Ѫѫ	Ѫѫ	Ѫѫ
	The Big Yus is used in manuscripts and some printed texts (for example, in the Gospels printed in Vilnius in 1575). It was used in the Bulgarian alphabet until 1945. In modern Church Slavonic, this character is only used in Paschalion tables. It should be available in all Church Slavonic fonts.							
A65A A65B	Ꙛ ꙛ	Cyrillic Capital Letter Blended Yus Cyrillic Small Letter Blended Yus	-	Ꙛꙛ	-	-	-	-
A65E A65F	Ꙟ ꙟ	Cyrillic Capital Letter Yn Cyrillic Small Letter Yn	ын	Ꙟꙟ	-	Ꙟꙟ	Ꙟꙟ	-
	The Romanian letter Yn was used in the Romanian Cyrillic alphabet, which was used in the Romanian Orthodox Church until the 1860’s. Modern Romanian uses the Latin alphabet both for secular and ecclesiastic literature. The letter was never used in Church Slavonic, but should nonetheless be included in fonts to provide support for Romanian Cyrillic texts.							
0468 0469	Ѩ ѩ	Cyrillic Capital Letter Iotified Little Yus Cyrillic Small Letter Iotified Little Yus	-	Ѩѩ	Ѩѩ	Ѩѩ	-	-
	This letter does not appear in poluustav typography. It exists only in the manuscript tradition.							

Continued on next page

Continued from previous page

Codes	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
A65C A65D	Ꙝ ꙝ	Cyrillic Capital Letter Iotified Closed Little Yus Cyrillic Small Letter Iotified Closed Little Yus	-	Ꙝꙝ	-	-	-	-
046C 046D	Ѭ ѭ	Cyrillic Capital Letter Iotified Big Yus Cyrillic Small Letter Iotified Big Yus	-	Ѭѭ	-	Ѭѭ	-	-
	This letter does not appear in poluustav typography. It exists only in the manuscript tradition.							
046E 046F	Ѯ ѯ	Cyrillic Capital Letter Ksi Cyrillic Small Letter Ksi	кси	Ѯѯ	ѯ	Ѯѯ	Ѯѯ	Ѯѯ
0470 0471	Ѱ ѱ	Cyrillic Capital Letter Psi Cyrillic Small Letter Psi	пси	Ѱѱ	Ѱѱ	Ѱѱ	Ѱѱ	Ѱѱ
0472 0473	Ѳ ѳ	Cyrillic Capital Letter Fita Cyrillic Small Letter Fita	фита	Ѳѳ	Ѳѳ	Ѳѳ	Ѳѳ	Ѳѳ
	In modern Church Slavonic, this character is used to transcribe the Greek letter *theta*. In older printed texts, the letters *fita* and *fert* are often used interchangeably since the phoneme [θ] does not exist in Church Slavonic. Note that this character should not be confused with the Cyrillic Letter Barred O (U+04E8 / U+04E9), which is used in Tatar, Kazakh, Yakut, and other non-Slavic languages. Although there is a visual similarity, the letter Barred O is not derived from the letter *fita*; the codepoints for the Barred O should not be used to encode the fita, and vice versa. The symbol for a musical Fita used in Znamenny notation is also a separate character (*vid.* Andreev and Simmons (2015)).							
0474 0475	Ѵ ѵ	Cyrillic Capital Letter Izhitsa Cyrillic Small Letter Izhitsa	ижица	Ѵѵ	Ѵѵ	Ѵѵ	Ѵѵ	Ѵѵ
0476 0477	Ѷ ѷ	Cyrillic Capital Letter Izhitsa with Double Grave Accent Cyrillic Small Letter Izhitsa with Double Grave Accent Canonical decomp.: 0474 030F / 0475 030F	ижица	Ѷѷ	-	Ѷѷ	Ѷѷ	Ѷѷ
	The inclusion of this character in Unicode is unfortunate. It is not a standalone character at all, but rather an Izhitsa with an accent mark (*kendema*). Conformant software and fonts should honor the canonical decomposition U+0474 U+030F (where U+030F is the Combining Double Grave Accent).							

2.2 Numerical Symbols

Church Slavonic uses letters to represent numbers and the particular issues related to working with numerals are discussed in a following section. Here, we list the additional characters required for numerals. Of these, only the first character (the thousands sign) is used in the print tradition. The remaining characters are used only in the manuscript tradition; however, they may be required in fonts for demonstration purposes or for use in academic literature.

Codes	Standard Representation		Representations				
	Char	Name	Ustav	Incun.	Polu.	Synod.	Skor.
0482	҂	Cyrillic Thousands Sign	҂	-	҂	҂	҂
20DD	◯	Combining Enclosing Circle	◯	-	◯	-	-

Continued on next page

Continued from previous page

Codes	Standard Representation		Representations				
	Char	Name	Ustav	Incun.	Polu.	Synod.	Skor.
	This codepoint is used for the ten thousands sign as per Unicode documentation. This and other Combining characters may be used as standalones if entered following U+00A0 NO-BREAK SPACE. See the Section below on implementation.						
0488	҈	Combining Cyrillic Hundred Thousands Sign	҈	-	҈	-	-
0489	҉	Combining Cyrillic Millions Sign	҉	-	҉	-	-
A670	꙰	Combining Cyrillic Ten Millions Sign	꙰	-	꙰	-	-
A671	꙱	Combining Cyrillic Hundred Millions Sign	꙱	-	꙱	-	-
A672	꙲	Combining Cyrillic Thousand Millions Sign	꙲	-	꙲	-	-
	This character has only been documented in combination with the letter Yeru to indicate one thousand million (one billion) as ы꙲; see Vostokov (1863, p. 9).						

2.3 Punctuation

The following Punctuation and related marks are part of the standard. See also the section on Miscellaneous Symbols and Pictographs for the more infrequent forms.

Codes	Standard Representation		Representations				
	Char	Standard name	Ustav	Incun.	Polu.	Synod.	Skor.
0020	SP	Space					
00A0	NB SP	Non-breaking Space					
	This character needs to be included in all fonts. It is used to represent combining characters as standalones. See the Section below on Spacing.						
202F	NNB SP	Narrow No-break Space					
	The non-breaking half-space is used in the poluustav script to closely tie together [proclitic] prepositions with nouns, particles with preceding words, and enclitic accents. For more information, see the section on Spacing.						
002E	.	Period	.	•	.	.	.
002C	,	Comma	,	,	,	,	,
003A	:	Colon	:	:	:	:	:
003B	;	Semicolon	;	;	;	;	;
	While this character looks like a modern semicolon (and is so called in the Unicode standard), it is actually the Greek form of the question mark, which is also the form typically used in Slavonic typography. Based on Unicode recommendations, this codepoint (U+003B) should be used for the Slavonic question mark. **Note**: neither U+037E GREEK QUESTION MARK nor U+003F QUESTION MARK should be used for the Slavonic question mark.						
(2E49)	⹉	Double Stacked Comma	-	-	⹉	⹉	-
	This character is used as a rubrical symbol in the liturgical manual of N. Syrnikov. It has been proposed for encoding in a future version of the Unicode Standard. See Andreev et al. (2015b) for more information.						
02BC	ʼ	Modifier Letter Apostrophe	ʼ	-	-	-	-
	In some manuscripts, this character is used to indicate Palatalization (Yelkina, 1960, p. 25). See the section below on Palatalization.						
0021	!	Exclamation Mark	-	-	-	!	-
003F	?	Question Mark	-	-	-	?	-

Continued on next page

Continued from previous page

Code	Standard Representation		Representations				
	Char	Standard name	Ustav	Incun.	Polu.	Synod.	Skor.
	The exclamation point and question mark are modern Western characters which were gradually introduced into the later Synodal Era publications. The Question Mark (U+003F) should not be used to encode the Slavonic Question Mark (U+003B), which looks like the modern semicolon.						
0028	(	Left Parenthesis	-	-	(	(	-
0029	)	Right Parenthesis	-	-	)	)	-
005B	[	Left Square Bracket	-	-	[	[	-
005D	]	Right Square Bracket	-	-	]	]	-
	The parentheses and square brackets are not used in ustav and poluustav texts. Their use is primarily confined to Synodal Era publications, but they are also found in many Old Ritualist reprints.						
002D	-	Hyphen-Minus	-	-	-	-	-
005F	_	Underscore	-	-	-	_	_
2010	‐	Hyphen	-	-	-	-	-
	In Synodal Slavonic, the Underscore is most often used for hyphenation, though sometimes the Hyphen is also used. However, the codepoint for the Hyphen (U+002D) should not be used to encode an Underscore. Rather, the software should be configured to use the appropriate hyphenation character. Furthermore, the preferred character for a hyphen is U+2010 HYPHEN, not U+002D. See the section below on Spacing and Hyphenation.						
00AB	«	Left-pointing Double Angle Quotation Mark	«	«	«	«	«
00BB	»	Right-pointing Double Angle Quotation Mark	»	»	»	»	»
	These characters are used in modern Russian and some other Slavic languages. They should not be used encode the Slavonic kavyka, but they need to be available in fonts because they are used by computer software.						
00B6	¶	Pilcrow Sign	¶	¶	¶	¶	¶
	This character should be included in all fonts because it is used for technical purposes.						
00AD	SHY	Soft Hyphen					
	Despite its name, this is not a hyphen, but rather an invisible format character that is used to indicate optional intraword breaks; see the Unicode Standard, p. 268. It should be included in all fonts. See the section on Spacing and Hyphenation, below.						
2011	‑	Non-breaking Hyphen	-	-	-	-	-
	This character should be included in all fonts because it is used for technical purposes. It is used to avoid word separation in specialized cases.						
2013	–	En Dash	–	–	–	–	–
2014	—	Em Dash	—	—	—	—	—
	These two characters should be included in all fonts because they are used for technical purposes. See the section below on Spacing and Hyphenation.						

2.4 Diacritical Marks

Church Slavonic uses a variety of diacritical marks that may occur above (and in rare instances, below) a character. This section provides a list of the diacritical marks. Note that the visual appearance of some diacritical marks may be different over an uppercase letter and a lowercase letter; however, this is handled via contextual substitution, as is discussed in the section below on Font Development. Other diacritical marks are compound marks, consisting of more than one combining character. The creation of compound marks as well as marks over multiple letters is discussed in the Implementation section. When discussing combining marks, they are presented over the character U+25CC DOTTED CIRCLE, which needs to be provided in all fonts.

Code	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
0301	◌́	Combining Acute Accent	oxia	◌́	◌́	◌́	◌́	◌́
0300	◌̀	Combining Grave Accent	varia	◌̀	◌̀	◌̀	◌̀	◌̀
0311	◌̑	Combining Inverted Breve	kamora	◌̑	◌̑	◌̑	◌̑	◌̑
	This character should not be used to indicate Palatalization. Rather, use U+0484 for palatalization. As well, the Kamora should **not** be encoded as U+0302 COMBINING CIRCUMFLEX ACCENT or U+0342 COMBINING GREEK PERISPOMENE.							
0486	◌҆	Combining Cyrillic Psili Pneumata	zvatelʹtse	◌҆	◌҆	◌҆	◌҆	◌҆
	This character – a breathing mark – is also called "psili". There are a number of combining characters (the Iso, the Apostrof, and the Great Apostrof) that include a combination of the Psili and another accent. See the Implementation section below. Note that U+0313 COMBINING COMMA ABOVE should not be used for the Cyrillic breathing mark.							
0485	◌҅	Combining Cyrillic Dasia Pneumata	dasia	◌҅	-	-	-	-
	The Dasia Pneumata, a "hard breathing", occurs only in ustav-era manuscripts of a Southern origin. Note that U+0314 COMBINING REVERSED COMMA ABOVE should not be used for the Cyrillic dasia.							
0306	◌̆	Combining Breve	kratkaya	◌̆	◌̆	◌̆	◌̆	◌̆
	This character is used for the Short I. It is also used for liturgical texts written in the Sakha (Yakut) language, and so should available in Synodal-recension fonts and position over any vowel.							
2E2F	ⸯ	Vertical Tilde	yerok	ⸯ	ⸯ	ⸯ	ⸯ	ⸯ
033E	◌̾	Combining Vertical Tilde	yerok	◌̾	◌̾	◌̾	◌̾	◌̾
	In some sources, this character is also called Yerik. In Synodal orthography, it is used to indicated on omitted hard sign. In other instances, it may also be used to indicate an omitted soft sign. Note that this character has an encoded non-combining form, U+2E2F.							
A67F	ꙿ	Cyrillic Payerok	payerok	ꙿ	ꙿ	ꙿ	ꙿ	-
A67D	◌꙽	Combining Cyrillic Payerok	payerok	◌꙽	◌꙽	◌꙽	◌꙽	-
	This character is not used in Synodal typography. In some poluustav printed and manuscript texts, it is used to indicate an omitted hard sign or an omitted soft sign. These codepoints should **not** be used to encode the Yerok (U+2E2F) or Combining Yerok (U+033E). **NB**: the Yerok (Yerik) and Payerok are two different characters and should not be interchanged or confused.							
A67E	꙾	Cyrillic Kavyka	kavyka	꙾	-	꙾	꙾	꙾
A67C	◌꙼	Combining Cyrillic Kavyka	kavyka	◌꙼	-	◌꙼	◌꙼	◌꙼
	The kavyka is used to indicate a marginal reading of a word or phrase. This character should not be confused with U+0306 COMBINING BREVE, which is used to indicate a short vowel sound.							
0307	◌̇	Combining Dot Above	-	◌̇	◌̇	◌̇	◌̇	◌̇
	This codepoint can be used to explicitly provide a dot over the i.							
0308	◌̈	Combining Diaeresis	kendema	◌̈	◌̈	◌̈	◌̈	◌̈
	This codepoint is used to produce the two dots over the i.							
030F	◌̏	Combining Double Grave Accent	kendema	◌̏	◌̏	◌̏	◌̏	◌̏
	The Kendema is used over the Izhitsa as well as in some other instances, for example, to indicate the dual of some words, or as a scripture verse divider, as in the *Ostrog Bible*.							
030B	◌̋	Combining Double Acute Accent	okovavy	◌̋	◌̋	-	-	-
	For the usage of this character, see Karsky (1979, p. 239).							
1DC1	◌᷁	Combining Dotted Acute Accent	-	◌᷁	-	◌᷁	-	-
	This symbol is used in the ustav manuscript tradition and in the *Ostrog Bible* to indicate marginal references or comments.							
1DC0	◌᷀	Combining Dotted Grave Accent	-	◌᷀	-	-	-	-
	Undocumented, but included by Kostić et al. (2009).							
0302	◌̂	Combining Circumflex Accent	perispomeni	◌̂	-	-	-	-

Continued on next page

Continued from previous page

Code	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
		Undocumented, but included by Kostić et al. (2009). Note that this is not a Kamora, which should be encoded at U+0311.						
0484	◌҄	Combining Cyrillic Palatalization	-	◌҄	-	◌҄	-	-
		Palatalization marks are not used in modern Church Slavonic. This codepoint should not be used for the Kamora, which does not indicate Palatalization. Use U+0311 for the Kamora. See the discussion of Palatalization below.						
0313	◌̓	Combining Comma Above	-	◌̓	-	-	-	-
0314	◌̔	Combining Reversed Comma Above	-	◌̔	-	-	-	-
		While these two characters are used for breath marks in Greek, they should not be used to encode the breathing marks in Cyrillic. See above for the proper way to encode the Cyrillic Psili and Dasia. However, these characters can be used to encode the Comma and Reversed Comma that are used as combining marks in some manuscripts to indicate palatalization. See the discussion of Palatalization below.						
0358	◌͘	Combining Dot Above Right	-	-	-	◌͘	◌͘	-
		This character is used in the system of Typicon symbols of N. Syrnikov. See Andreev et al. (2015b) for more information.						
(1DF6)	◌᷶	Combining Kavyka Above Right	-	-	-	◌᷶	◌᷶	-
(1DF7)	◌᷷	Combining Kavyka Above Left	-	-	-	◌᷷	◌᷷	-
(1DF8)	◌᷸	Combining Dot Above Left	-	-	-	◌᷸	◌᷸	-
(1DF9)	◌᷹	Combining Wide Inverted Bridge Below	-	-	-	◌᷹	◌᷹	-
		The above four characters are used in the system of Typicon symbols of N. Syrnikov. They are proposed for encoding in a future version of the Unicode Standard. See Andreev et al. (2015b) for more information.						
0483	◌҃	Combining Cyrillic Titlo	titlo	◌҃	◌҃	◌҃	◌҃	◌҃
A66F	◌꙯	Combining Cyrillic Vzmet	vzmet	◌꙯	-	◌꙯	-	-
0487	◌҇	Combining Cyrillic Pokrytie	pokrytie	◌҇	◌҇	◌҇	◌҇	◌҇
0305	◌̅	Combining Overline	-	◌̅	-	-	-	-
033F	◌̿	Combining Double Overline	-	◌̿	-	-	-	-
		In historical texts (including the manuscript tradition) the use of titlo, vzmet, and pokrytie is somewhat interchangeable and the distinction is purely typographic. A straight supralineation (like the Longa or even the Double Longa) may also occur. Lettered titloi can occur with and without a pokrytie (or vzmet, more rarely, titlo or longa). Either the titlo or the vzmet can occur in nomina sacra, sigla (scribal abbreviations), or numerals. The distinction between titlo and vzmet is purely typographical; the vzmet is not used in Synodal typography. The codepoint of the titlo should not be used to encode a vzmet character and vice versa.						
FE2E	◌︮	(Combining Titlo Left Half)	titlo	◌︮	-	-	-	-
		This character is used as the left-most component of a titlo when a titlo balances over multiple letters. It has been encoded in Unicode 8.0. See the discussion of diacritical marks over multiple characters in the Implementation section.						
FE2F	◌︯	(Combining Titlo Right Half)	titlo	◌︯	-	-	-	-
		This character is used as the right-most component of a titlo when a titlo balances over multiple letters. It has been encoded in Unicode 8.0. See the discussion of diacritical marks over multiple characters in the Implementation section.						
FE26	◌︦	Combining Conjoining Macron	titlo	◌︦	-	-	-	-
		This character is used as a middle component of a titlo when a titlo balances over multiple letters. See the discussion of diacritical marks over multiple characters in the Implementation section.						

2.5 Combining Letters

In the Unicode standard, the combining letters appear in their uncomposed form, that is, without a titlo or pokrytie. In Synodal typography, the use of combining letters is completely standardized, that is, certain letters appear only under a Pokrytie and certain letters appear only without a pokrytie. In earlier recensions of Slavonic, the usage is less standardized – the combining letters may appear with or without a pokrytie, or, less frequently, titlo, or vzmet. In any case, whenever a combining letter needs to be with a pokrytie, the pokrytie must be specifically encoded following the combining letter. At the font level, the combination is handled via mark-to-mark positioning or via glyph composition. This is discussed in the Implementation section. In no circumstances should font manufacturers encode precomposed combining letters with a pokrytie at the codepoints listed below. Note that in the Unicode codecharts, the modern appearances of the letters are used, but period-specific representations should be provided in fonts.

Code	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
2DF6	◌ⷶ	Combining Cyrillic Letter A	az	◌ⷶ	-	◌ⷶ	-	◌ⷶ
2DE0	◌ⷠ	Combining Cyrillic Letter Be	buki	◌ⷠ	◌ⷠ	◌ⷠ	◌ⷠ	◌ⷠ
	This letter occurs in Synodal Slavonic only in the abbreviation сꙋⷠ҇ (суббота).							
2DE1	◌ⷡ	Combining Cyrillic Letter Ve	vedi	◌ⷡ	◌ⷡ	◌ⷡ	◌ⷡ	◌ⷡ
2DE2	◌ⷢ	Combining Cyrillic Letter Ge	glagol	◌ⷢ	◌ⷢ	◌ⷢ	◌ⷢ	◌ⷢ
	This letter may occur in Synodal Slavonic with or without a Pokrytie.							
2DE3	◌ⷣ	Combining Cyrillic Letter De	dobro	◌ⷣ	◌ⷣ	◌ⷣ	◌ⷣ	◌ⷣ
	This letter may occur in Synodal Slavonic only without a Pokrytie.							
2DF7	◌ⷷ	Combining Cyrillic Letter Ie	yest	◌ⷷ	◌ⷷ	◌ⷷ	-	◌ⷷ
A674	◌ꙴ	Combining Cyrillic Letter Ukrainian Ie	yest	◌ꙴ	◌ꙴ	◌ꙴ	-	◌ꙴ
2DE4	◌ⷤ	Combining Cyrillic Letter Zhe	zhivete	◌ⷤ	◌ⷤ	◌ⷤ	◌ⷤ	◌ⷤ
	This letter occurs in Synodal Slavonic only in the forms триⷤ (трижды) and дваⷤ (дважды).							
2DE5	◌ⷥ	Combining Cyrillic Letter Ze	zemlya	◌ⷥ	◌ⷥ	◌ⷥ	◌ⷥ	◌ⷥ
	This letter occurs in Synodal Slavonic only in the forms праⷥ҇ (праздник) and роⷥ҇ (розводъ (фиты)).							
A675	◌ꙵ	Combining Cyrillic Letter I	izhe	◌ꙵ	◌ꙵ	◌ꙵ	-	◌ꙵ
	This character should not be confused with the Kendema, which is encoded as the Double Grave Accent, U+030F. While the Kendema is a diacritical mark, this character indicates an omitted Letter I.							
A676	◌ꙶ	Combining Cyrillic Letter Yi	i	◌ꙶ	-	◌ꙶ	-	◌ꙶ
2DF8	◌ⷸ	Combining Cyrillic Letter Djerv	dzherv	◌ⷸ	-	-	-	-
2DE6	◌ⷦ	Combining Cyrillic Letter Ka	kako	◌ⷦ	◌ⷦ	◌ⷦ	◌ⷦ	◌ⷦ
2DE7	◌ⷧ	Combining Cyrillic Letter El	lyudie	◌ⷧ	◌ⷧ	◌ⷧ	◌ⷧ	◌ⷧ
	This letter occurs in Synodal Slavonic only in the forms ѱаⷧ҇ (псаломъ) and нлⷣ҇ѧ (недѣля).							
2DE8	◌ⷨ	Combining Cyrillic Letter Em	myslete	◌ⷨ	◌ⷨ	◌ⷨ	◌ⷨ	◌ⷨ
	This letter occurs in Synodal Slavonic only in the form риⷨ ((къ) Римляномъ).							
2DE9	◌ⷩ	Combining Cyrillic Letter En	nash	◌ⷩ	◌ⷩ	◌ⷩ	◌ⷩ	◌ⷩ
	This letter occurs in Synodal Slavonic only in the form солꙋⷩ҇ ((къ) Солуняномъ).							
2DEA	◌ⷪ	Combining Cyrillic Letter O	on	◌ⷪ	◌ⷪ	◌ⷪ	◌ⷪ	◌ⷪ
2DEB	◌ⷫ	Combining Cyrillic Letter Pe	pokoy	◌ⷫ	◌ⷫ	◌ⷫ	◌ⷫ	◌ⷫ
2DEC	◌ⷬ	Combining Cyrillic Letter Er	rtsy	◌ⷬ	◌ⷬ	◌ⷬ	◌ⷬ	◌ⷬ
2DED	◌ⷭ	Combining Cyrillic Letter Es	slovo	◌ⷭ	◌ⷭ	◌ⷭ	◌ⷭ	◌ⷭ
2DEE	◌ⷮ	Combining Cyrillic Letter Te	tverdo	◌ⷮ	◌ⷮ	◌ⷮ	-	◌ⷮ
	In Synodal typography, this character is not used. The letter ѿ is a ligature and is not decomposable to the letter ѡ and the combining te (◌ⷮ).							
2DF9	◌ⷹ	Combining Cyrillic Letter Monograph Uk	uk	◌ⷹ	◌ⷹ	◌ⷹ	-	◌ⷹ
A677	◌ꙷ	Combining Cyrillic Letter U	u	◌ꙷ	-	◌ꙷ	-	◌ꙷ

Continued on next page

Continued from previous page

Code	Standard Representation		Name	Representations				
	Char	Standard name		Ustav	Incun.	Polu.	Synod.	Skor.
A69E	◌ꚞ	Combining Cyrillic Letter Ef	fert	◌ꚞ	-	◌ꚞ	-	-
	This letter occurs in the *Ostrog Bible* and several other sources. It has been encoded in Unicode 8.0.							
2DEF	◌ⷯ	Combining Cyrillic Letter Ha	kher	◌ⷯ	◌ⷯ	◌ⷯ	◌ⷯ	◌ⷯ
	This letter occurs in Synodal Slavonic only in the forms сⷯ (стихъ) and варꙋⷯ (Варухъ).							
A67B	◌ꙻ	Combining Cyrillic Letter Omega	omega	◌ꙻ	-	◌ꙻ	-	◌ꙻ
2DF0	◌ⷰ	Combining Cyrillic Letter Tse	tsy	◌ⷰ	-	◌ⷰ	-	◌ⷰ
2DF1	◌ⷱ	Combining Cyrillic Letter Che	cherv	◌ⷱ	◌ⷱ	◌ⷱ	◌ⷱ	◌ⷱ
	This letter occurs in Synodal Slavonic only in the form ꙁⷱ҇л (зачало).							
2DF2	◌ⷲ	Combining Cyrillic Letter Sha	sha	◌ⷲ	-	◌ⷲ	-	◌ⷲ
2DF3	◌ⷳ	Combining Cyrillic Letter Shcha	shcha	◌ⷳ	-	◌ⷳ	-	◌ⷳ
A678	◌ꙸ	Combining Cyrillic Letter Hard Sign	yer	◌ꙸ	-	◌ꙸ	-	◌ꙸ
A67A	◌ꙺ	Combining Cyrillic Letter Soft Sign	yer′	◌ꙺ	-	◌ꙺ	-	◌ꙺ
A679	◌ꙹ	Combining Cyrillic Letter Yeru	yeru	◌ꙹ	-	◌ꙹ	-	◌ꙹ
2DFA	◌ⷺ	Combining Cyrillic Letter Yat	yat	◌ⷺ	-	◌ⷺ	-	◌ⷺ
2DFB	◌ⷻ	Combining Cyrillic Letter Yu	yu	◌ⷻ	-	◌ⷻ	-	◌ⷻ
2DFC	◌ⷼ	Combining Cyrillic Letter Iotified A	iotified az	◌ⷼ	-	◌ⷼ	-	◌ⷼ
A69F	◌ꚟ	Combining Cyrillic Letter Iotified E	iotified yest	◌ꚟ	-	◌ꚟ	-	-
2DFD	◌ⷽ	Combining Cyrillic Letter Little Yus	little yus	◌ⷽ	-	◌ⷽ	-	◌ⷽ
2DFE	◌ⷾ	Combining Cyrillic Letter Big Yus	big yus	◌ⷾ	-	-	-	-
2DFF	◌ⷿ	Combining Cyrillic Letter Iotified Big Yus	iotified big yus	◌ⷿ	-	-	-	-
2DF4	◌ⷴ	Combining Cyrillic Letter Fita	fita	◌ⷴ	◌ⷴ	◌ⷴ	◌ⷴ	◌ⷴ
	This letter occurs in Synodal Slavonic only in the form коріⷴ҇ ((къ) Коринѳяномъ).							
2DF5	◌ⷵ	Combining Cyrillic Letter Es-Te	-	◌ⷵ	◌ⷵ	◌ⷵ	-	◌ⷵ
	This character is a ligature of the Combining Es (U+2DED) and the Combining Te (U+2DEE). The character at this codepoint should not be used, but instead the ligature should be encoded as the sequence U+2DED U+2DEE (see the discussion of combining marks in the Implementation section). The encoding of this character in Unicode was erroneous.							

2.6 Miscellaneous Symbols and Pictographs

This section describes symbols used in the Typicon, various ornamental punctuation forms, and other pictographs. Note that we do not present here the more complex Typicon marks used in the Typicon-at-a-Glance (*Окозрительный устав*) of Archbishop Gennadius of Novgorod and later sources. The authors plan to address these more elaborate systems of rubrical Typicon Symbols in a separate document.

Codes	Standard Representation		Representations				
	Char	Standard name	Ustav	Incun.	Polu.	Synod.	Skor.
1F540	🕀	Circled Cross Pommee	-	-	🕀	🕀	-

Continued on next page

Continued from previous page

Code	Standard Representation		Representations				
	Char	Standard name	Ustav	Incun.	Polu.	Synod.	Skor.
	The Typicon symbols were not used in the ustav period. Their earliest appearance is in the *Regulations* of Nikon of the Black Mountain, an 11th century Greek document, but the forms here are characteristic of the Slavonic documents. This symbol always appears in red.						
1F541	🕁	Cross Pommee with Half-Circle Below	-	-	🕁	🕁	-
	This symbol always appears in red.						
1F542	🕂	Cross Pommee	-	-	🕂	🕂	-
	This symbol always appears in red.						
1F543	🕃	Notched Left Semicircle With Three Dots	-	-	🕃	🕃	-
	[Placed in a left outer page margin] - Great Doxology Feast symbol (red ink), or Special Simple (Hexasticheraric) Service symbol (black ink).						
1F544	🕄	Notched Right Semicircle With Three Dots	-	-	🕄	🕄	-
	[Placed in a right outer page margin] - Great Doxology Feast symbol (red ink), or Special Simple (Hexasticheraric) Service symbol (black ink). In modern usage, the two different directions of this symbol can potentially be used in single-color printing to differentiate between the more traditional red and black forms of this symbol (which indicate different usages in the Typicon). This character has been encoded in Unicode 7.0.						
1F545	🕅	Symbol for Marks Chapter	🕅	-	🕅	🕅	-
	This glyph represents the Chapters of Mark the Monk (Марковы главы), which are included in the Menaia, Triodia, and the Typicon. These chapters provide instructions for important feasts when they coincide with Sundays or other feasts or important commemorations. The symbol is placed in the margin to draw the eye to the reader. This character has been encoded in Unicode 7.0. Variants of this symbol also exist and are handled at the font level; see the discussion of Stylistic Alternatives, below.						
(1F900)	🤀	Circled Cross Formee with Four Dots	-	-	-	🤀	-
(1F901)	🤁	Circled Cross Formee with Two Dots	-	-	-	🤁	-
(1F902)	🤂	Circled Cross Formee	-	-	-	🤂	-
(1F903)	🤃	Left Half Circle with Four Dots	-	-	-	🤃	-
(1F904)	🤄	Left Half Circle with Three Dots	-	-	-	🤄	-
(1F905)	🤅	Left Half Circle with Two Dots	-	-	-	🤅	-
(1F906)	🤆	Left Half Circle with Dot	-	-	-	🤆	-
(1F907)	🤇	Left Half Circle	-	-	-	🤇	-
	The above eight characters are used as rubrical symbols in the Byzantine Catholic Typicon of Isidore Dolnitsky. They have been proposed for encoding in a future version of the Unicode Standard. See Andreev et al. (2015b) for more information.						
(2E45)	⹅	Inverted Low Kavyka	-	-	⹅	⹅	-
(2E46)	⹆	Inverted Low Kavyka with Kavyka Above	-	-	⹆	⹆	-
(2E47)	⹇	Low Kavyka	-	-	⹇	⹇	-
(2E48)	⹈	Low Kavyka with Dot	-	-	⹈	⹈	-
29DF	⧟	Double-Ended Multimap	-	-	⧟	⧟	-
(1F908)	🤈	Downward Facing Hook	-	-	🤈	🤈	-
(1F909)	🤉	Downward Facing Notched Hook	-	-	🤉	🤉	-
(1F90A)	🤊	Downward Facing Hook with Dot	-	-	🤊	🤊	-
(1F90B)	🤋	Downward Facing Notched Hook with Dot	-	-	🤋	🤋	-
	The above nine characters are used as rubrical symbols in the liturgical manual of N. Syrnikov. The characters with codepoints in parentheses have been proposed for encoding in a future version of the Unicode Standard. See Andreev et al. (2015b) for more information.						
2626	☦	Orthodox Cross	☦	-	☦	☦	-

Continued on next page

Continued from previous page

Code	Standard Representation		Representations				
	Char	Standard name	Ustav	Incun.	Polu.	Synod.	Skor.
	The Orthodox Cross was a symbol that occasionally was used as decoration in ustav manuscripts, but it was rare. By the time of the poluustav tradition, it was informally considered forbidden to print the Cross by means of movable type (although we do see the Maltese Cross in some of the early editions of the Apostol and Gospel). Only after the reforms of Patriarch Nikon (the Synodal Era) do we see the Orthodox Cross (and iconographic illustrations) in printed books.						
2720	✠	Maltese Cross	✠	-	✠	✠	-
	The "Maltese Cross" glyph was used in several early printed books for a variety of reasons. In many Gospels it was used like an asterisk to indicate instructions in the margins concerning how to introduce the liturgical readings. In the introduction to one of the early Psalters of Ivan Fedorov, it was used to compare and contrast different spellings of words. It is also used in the *Trebnik* of Metropolitan Peter (Mogila).						
A673	꙳	Slavonic Asterisk	꙳	-	꙳	꙳	꙳
	This character, and **not** the Asterisk (U+002A) should be used in liturgical texts to indicate breaks into melodic lines for chanting the text to an automelon melody. This character has some peculiar kerning properties that need to be implemented at the font level.						
	The following symbols are used for punctuation in ustav manuscripts.						
00B7	·	Middle Dot	·	·	·	-	-
2E34	⸴	Raised Comma	⸴	-	-	-	-
2022	•	Bullet	•	-	-	-	-
2E43	⹃	Dash with Left Upturn	⹃	-	⹃	⹃	-
	This character has been encoded in Unicode 8.0. It is used at the end of paragraphs in some manuscripts and printed sources, and is perhaps related to U+2E0F PARAGRAPHOS.						
2053	⁓	Swung Dash	⁓	-	-	-	-
223C	∼	Tilde	∼	-	-	-	-
	This character is just like the Swung Dash, but narrower.						
223D	∽	Reversed Tilde	∽	-	-	-	-
223B	∻	Homothetic	∻	-	-	-	-
2241	≁	Not Tilde	≁	-	-	-	-
205D	⁝	Tricolon	⁝	-	⁝	-	-
2056	⁖	Three Dot Punctuation	⁖	-	⁖	-	-
10FB	჻	Georgian Paragraph Separator	჻	-	჻	-	-
2E2B	⸫	One Dot over Two Dots Punctuation	⸫	-	⸫	-	-
2E2A	⸪	Two Dots over One Dot Punctuation	⸪	-	⸪	-	-
205E	⁞	Vertical Four Dots	⁞	-	-	-	-
2058	⁘	Four Dot Punctuation	⁘	-	⁘	-	-
2E2C	⸬	Squared Four Dot Punctuation	⸬	-	⸬	-	-
2059	⁙	Five Dot Punctuation	⁙	-	⁙	-	-
2E2D	⸭	Five Dot Mark	⸭	-	⸭	-	-
2056 2E43	⁖⹃	Three Dot Punctuation + Spear	⁖⹃	-	⁖⹃	⁖⹃	-
	This character is encoded as a digraph, perhaps with some kerning.						
003A 2E43	:⹃	Colon + Spear	:⹃	-	-	-	-
	This character is encoded as a digraph, perhaps with some kerning.						
2052	⁒	Commercial Minus Sign	⁒	-	-	-	-
	The Unicode annotation for this codepoint states: "may also be used as a dingbat to indicate correctness". For the combining version of this character, see the above section on Diacritical Marks.						
203B	※	Reference Mark	※	-	※	※	-

Continued on next page

Continued from previous page

Code	Standard Representation		Representations				
	Char	Standard name	Ustav	Incun.	Polu.	Synod.	Skor.
	This can be used as a final section marker or as a reference mark (a secondary asterisk).						
205C	⁜	Dotted Cross	⁜	-	⁜	⁜	-
	This can be used as a final section marker or as a reference mark (a secondary asterisk).						
2020	†	Dagger	†	†	†	†	†
	This character should be included in all fonts because it is used for technical purposes.						

2.7 Ecphonetic Notation

Ecphonetic notation is neumatic musical notation placed in liturgical books to indicate cantillation (musical recitation of sacred texts). Ecphonetic notation of the Byzantine kind occurs in some early Slavonic manuscripts; in this section, we present the repertoire of characters that occurs in the Ostromir Gospel (Остромирово евангелие) and the Fragments of Kupriyanov (Куприяновские листки), both of which date to the 11th century. The ecphonetic notation in these manuscripts has been recently studied by Zagrebin (2006). Since this is the same notation as used in Byzantine manuscripts, the symbols encoded in the Byzantine Musical Notation block of Unicode should be used. For more information on working with Byzantine Musical Notation in Unicode, see Nicholas (2006). Since the characters are only found in early manuscripts, they only need to be provided in fonts that imitate ustav-era writing. Note that all of the Byzantine Musical Notation characters encoded in Unicode are base characters, not combining marks. In plain text, there is no method to position any of these marks over a base character in order to render ecphonetic notation above a line of text. This rendering issue is relegated to markup language or other higher-level protocols, for example, TEX or music notation software.

Name	Codepoint	Representation
Oxeia	U+1D003	𝀃
Vareia	U+1D005	𝀅
Double Vareia	U+1D006	𝀆
Kathisti	U+1D007	𝀇
Syrmatiki	U+1D008	𝀈
Paraklitiki	U+1D009	𝀉
Hypokrisis Dipli	U+1D00B	𝀋
Kremasti	U+1D00C	𝀌
Teleia	U+1D00F	𝀏
Kentimata	U+1D010	𝀐
Apostrofos	U+1D011	𝀑
Perispomeni	U+1D002	𝀂
The three additional characters indicated below are peculiar to the Ostromir Gospel. They contain other notational symbols that were added to the Telia. According to Zagrebin (2006, p. 146), these additions were done at a later time by a different scribe. Some of the combining marks occur in red, but the issue of rendering color is beyond the scope of this paper. Since combining versions of the Perispomeni and the Syrmatiki have not been encoded in Unicode, we suggest that these symbols be encoded using similar characters found in the Combining Diacritical Marks block.		
Teleia with Syrmatiki	U+1D00F U+0303	𝀏̃
Teleia with Bridge and Syrmatiki	U+1D00F U+032A U+0303	𝀏̪̃
Teleia with Syrmatiki and Perispomeni	U+1D00F U+0311 U+0303	𝀏̑̃

3 Implementation

This section documents issues of implementation; that is, if the previous section listed which codepoints are to be used to encode which Church Slavonic characters, this section describes *how* these codepoints are used. In many instances, the proper usage is obvious – thus, for example, when we are encoding individual characters. Here, any potential pitfalls have been discussed above, in the listing of codepoints (for example, in our discussion of the letter оу, we mention that the codepoint U+0478 should not be used). Thus, in this section we discuss only certain complex issues of implementation.

3.1 Combining Diacritical and Enclosing Marks

Church Slavonic uses a wide variety of superscript (combining) diacritical marks. In the Unicode Standard, the term "diacritic" is defined very broadly to include accents as well as other nonspacing marks (Unicode Standard, p. 55). The majority of relevant diacritical marks are located in the Combining Diacritical Marks block. Per Unicode specifications, these marks can be used with any language or script (though this may cause pitfalls for designers of multi-language fonts). In addition, there are a number of script-specific combining diacritical marks in the Cyrillic block (notably, the titlo, psili, and pokrytie) and in the Cyrillic Extended-B block (the vzmet, kavyka, and payerok). As well, combining superscript letters are encoded in the Cyrillic Extended-A and Cyrillic Extended-B blocks. In the Unicode character charts, as well as in the charts in this Technical Note, a combining character is presented over the Dotted Circle (U+25CC).

All combining diacritical marks and letters are used in sequence following the base characters to which they apply. This is a convention of the Unicode Standard and is also consistent with modern font technologies (*ibid.*). We demonstrate the behavior in the following example:

а + ◌́ + о ⟶ а́о (*not* ао́)

In addition to combining superscript marks, Church Slavonic uses a set of combining enclosing marks (though these are not used in Synodal-era typography). These marks are used to create the numerals for ten thousand and higher. The behavior of these marks is the same as for combining superscript marks; to write the numeral for ten thousand, for example, we enter:

а + ◌⃝ ⟶ а⃝

Note that font designers need to take care that proper contextual kerning is implemented in the font for these instances. In addition, the character may need to be "shrunk" in order to fit into the combining mark, so it may be necessary to provide some numerals as precomposed glyphs.

3.2 Combining Marks in Isolation

In some cases it may be necessary to display the combining superscript mark or enclosing mark in isolation, that is, not applied to any character. For example, this may be done in a discussion of the mark in a primer or grammar reference. The mark may be displayed in isolation by applying it to U+00A0 (No-Break Space). For this purpose, conformant fonts must include U+00A0 in their character repertoire (see the section below on Spacing for more information). Note that prior to Version 4.1 of the Unicode Standard, the character U+0020 (Space) was used for displaying combining marks in isolation; however this is no longer recommended "because of potential conflicts with the handling of sequences of U+0020 (Space) characters in such contexts as XML" (Unicode Standard, p. 60).

Note also that a number of Cyrillic diacritical marks have non-combining versions encoded in the Unicode Standard: Combining Yerok has a non-combining form U+2E2F Vertical Tilde; Combining Kavyka has a non-combining form U+A67E Cyrillic Kavyka; and Combining Payerok has a non-combining form U+A67F Cyrillic Payerok. These diacritical marks are sometimes encountered as non-spacing diacritics and sometimes as spacing characters.

3.3 Multiple Combining Marks

In some instances, more than one diacritical mark is encountered over one base character. In principle, the number of diacritical marks that may be used is without limit, though in practice we rarely encounter more than two marks in Church Slavonic; these usually interact and follow specific typographic rules. It is important to note that in this case – when the marks interact – the display is strictly dependent on the order in which the marks are entered, whereas when the marks do not interact, the order of entry is irrelevant, but the display is governed by what is called "canonical ordering".

By default, marks that do not interact typographically are stacked vertically and positioned from the base glyph outward. Marks that do interact are positioned side by side or form ligatures, overriding default stacking behavior. It is recommended that font designers should strive to support vertical stacking behavior, because, among other uses, it provides users with a visual indicator that a sequence of combining marks has potentially been entered in the wrong order.

The correct order is particularly relevant for three widely used combining marks in modern Church Slavonic: the *iso* (a soft breathing with acute accent), *apostrof* (a soft breathing with grave accent), and *veliky apostrof* (a soft breathing with inverted breve). As the following examples demonstrate, the correct order of entry for these marks is base glyph + breathing mark + accent mark:

а + ◌҆ + ◌́ ⟶ а҆́ (*correct*)
а + ◌҆ + ◌̑ ⟶ а҆̑ (*correct*)
а + ◌́ + ◌҆ ⟶ а́҆ (*incorrect*)
а + ◌̑ + ◌҆ ⟶ а̑҆ (*incorrect*)

Note that, despite visual appearance, the second part of the *veliky apostrof* digraph is the inverted breve (U+0311), not a pokrytie (U+0487), since this character is a type of accent, related to the Greek breathing with circumflex accent (for example, in the Greek interjection ὦ). In Synodal Slavonic, the *veliky apostrof* is only used in the interjection ѽ, though in earlier recensions, it may be found over other characters. It is also important to note that the character for ѽ has been encoded in the Unicode Standard at U+047D (U+047C for the uppercase version) with the erroneous name CYRILLIC LETTER OMEGA WITH TITLO. The diacritical mark is, in fact, a *veliky apostrof*, and not a titlo, and, despite the fact that Unicode does not provide for a canonical decomposition for this character, it is linguistically and typographically equivalent to the sequence U+A64D CYRILLIC LETTER BROAD OMEGA (U+A64C for the uppercase form); U+0486; U+0311. We suggest that font designers implement both versions, allowing the sequence U+A64D U+0486 U+0311 to map to U+A67D via Glyph Composition / Decomposition and that implementers be keenly aware that the presence of two potential encodings for this character may cause problems which are best treated with a proper collation specification (see below).

In addition to these commonly encountered combinations of combining marks, there may be other, less frequent combinations that are used in the ustav recension of Church Slavonic. Whenever such a combination is supported in the manuscript or typographic tradition, the rendering system should override the default stacking behavior. At the font level, this is achieved either via the use of mark-to-mark positioning (where possible) or precomposed ligatures (see the section on font design below). In Table 8, we present some commonly encountered combinations of combining marks in Church Slavonic of the ustav recension. Note that the implementation of some of these combinations is dubious. For example, it is not clear, either from the linguistic or from the typographic standpoint, if two acute accents should be treated the

Table 8: Combinations of combining marks used in Church Slavonic

		Second character								
		Dot Above	Double Dot Above	Oxia	Varia	Psili	Dasia	Kamora	Perispomeni	Payerok
First character	Dot Above			◌̇́	◌̇̀					
	Double Dot Above			◌̈́	◌̈̀					
	Oxia			◌́́						
	Varia				◌̀̀					
	Psili			◌҆́	◌҆̀	◌҆҆	◌҆҅	◌҆̑	◌҆͂	
	Dasia			◌҅́	◌҅̀		◌҅҅	◌҅̑	◌҅͂	
	Kamora									
	Perispomeni									
	Payerok			◌꙽́	◌꙽̀					

same as U+030B COMBINING DOUBLE ACUTE ACCENT, which is used to encode the *okovavy*, and two grave accents – the same as U+030F COMBINING DOUBLE GRAVE ACCENT, which is used to encode the *kendema*. Again, implementers should be aware of possible encoding ambiguities when performing string search and comparison operations.

Each Unicode character has a number of properties and one of these properties is the *combining class* property. All base glyphs have a combining class property of zero. Combining marks have varying combining class properties that are indicative of their expected position vis-à-vis the base character. So far, all of the marks that we have discussed in this section have a combining class of 230, meaning that they are positioned directly above the base character. Note that the U+033E COMBINING VERTICAL TILDE (used to encode the yerok) and U+A67D COMBINING CYRILLIC PAYEROK (used to encode the payerok) both also have combining class properties of 230, although by typographic convention they position on the right shoulder of the base character, as in ◌꙽.

Marks that have the same combining class property may interact to form combining ligatures, but when one considers marks of different combining classes, the issue of *canonical ordering* must be kept in mind. Canonical ordering is a mechanism to ensure the canonical equivalence of Unicode strings that should have the same graphical representation but may have been entered differently at the codepoint level. Canonical ordering is designed in such a way so that combining marks of the same canonical class may interact while combining marks of different combining classes do not interact. In modern Church Slavonic, since all of the combining marks used in the writing system have a combining class property of 230, this is not an issue. However, it may become an issue in working with certain non-standard texts, for example, with the Typicon symbols used by Syrnikov (1910, f. 18ff). In particular, consider the following symbol: ◌҃͘. Since the combining class of U+0358 COMBINING DOT ABOVE RIGHT is 232 and the combining class of U+0483 COMBINING CYRILLIC TITLO is 230, when normalization is applied, the Combining Dot is ordered after the Combining Titlo and, as required by the Unicode Canonical Ordering algorithm, the following character sequences produce the same visual representation:

◌҃ + ◌͘ ⟶ ◌҃͘ (*normal order*)
◌͘ + ◌҃ ⟶ ◌҃͘ (*canonically equivalent*)

To retain the desired order of the Combining Dot and the Titlo and prevent canonical reordering, one can insert U+034F COMBINING GRAPHEME JOINER in the sequence between the Dot and the Titlo. Since the Combining Grapheme Joiner has a combining class property of zero, it prevents the canonical reordering of the diacritical marks and assures that under normalization the desired typographical representation is preserved:

◌̇ + ◌҃ ⟶ ◌҃̇ (*canonically reordered*)
◌̇ + ◌͏ + ◌҃ ⟶ ◌̇͏҃ (*desired representation*)

Font designers should provide U+034F COMBINING GRAPHEME JOINER in their fonts because of its use as a mechanism to control canonical reordering and rendering of certain sequences. The CGJ is classified as a nonspacing combining mark, and should be treated as such at the font level. Note that despite its name, U+034F COMBINING GRAPHEME JOINER is not used to join graphemes, and its function should not be confused with that of U+200D ZERO WIDTH JOINER (see below). Its other purpose is to affect collation. See the Unicode Standard, p. 817f, for more information.

Finally, a few words are warranted about one important class of combining marks – combining superscript letters, which may occur with or without a pokrytie. Note that in the Unicode Standard, the combining Cyrillic superscript letters are encoded *without* a pokrytie and have a combining class property of 230. Thus, the character encoded at U+2DED COMBINING CYRILLIC LETTER ES should be represented as ◌ⷭ and *not* as ◌ⷭ҇. In the Synodal recension of Church Slavonic, the use of combining characters is completely standardized – some occur only with the pokrytie while others occur only without the pokrytie. (This is not absolutely the case; for example, the Combining Cyrillic Letter Ge may occur without the pokrytie as ◌ⷢ in some editions and with the pokrytie in other editions as ◌ⷢ҇.) In other recensions of Church Slavonic, the combining letters may occur by themselves or with a pokrytie (U+0487), vzmet (U+A66F), or other form of supralineation. Whenever it is necessary to represent a combining letter with supralineation, the correct encoding sequence is: "base glyph; combining superscript letter; combining supralineation character", as in the following example:

а + ◌ⷢ ⟶ аⷢ
а + ◌ⷢ + ◌҇ ⟶ аⷢ҇

Font designers may support the correct positioning of the supralineation character (pokrytie or vzmet) over the combining letter either via the use of mark-to-mark positioning or via the use of precomposed glyphs (ligatures). Since, as we have already stated, the use of combining characters is completely standardized in Synodal orthography, we list in Appendix B all of the forms commonly encountered. In addition to aiding font developers, a complete list of Slavonic abbreviations is useful for the development of tools for language processing and morphological analysis.

In working with poluustav texts, and especially with manuscripts, multiple combining letters occurring over a single base character may be encountered. In such instances, two combining letters typically form a composite combining letter, where the components are stacked side-by-side or ligated. This deviates from the normal vertical stacking behavior of combining characters in the Unicode Standard. Font designers may choose to provide some of these horizontally-stacked combining letters or ligatures as precomposed glyphs. Thus, the following behavior is typically expected:

а + ◌ⷣ + ◌ꙵ ⟶ аⷣꙵ

On occasion, one may encounter combining characters that stack vertically. The Unicode Standard does not provide for any explicit mechanism to control the stacking behavior, relegating desired representation to the font level. Where advanced font features or markup cannot be used – such as in plain text contexts – the character U+034F COMBINING GRAPHEME JOINER may be inserted between the combining letters to force vertical stacking behavior. If encountered in the data, the CGJ may prevent the two combining letters from forming a ligature. In other situations where its usage is not supported or required, the CGJ should be ignored. See the Unicode Standard, p. 316, for more information.[15]

3.4 Combining Marks over Multiple Base Characters

In some instances, it is necessary to position a combining mark (a titlo) over two or more base glyphs. Although this does not occur in Synodal or poluustav type, such features occur frequently in iconography, for example in the inscription М︮Р︯ Ѳ︮Ꙋ︯ (a Slavonic rendition of the Greek Μήτηρ του Θεού, *Mother of God*) or І︮С︯ Х︮С︯ (І҆исꙋ́съ Хрїсто́съ, *Jesus Christ*). In addition, with some frequency in ustav, poluustav, and skoropis manuscripts, the titlo may be found to balance over two, three, or more letters, as in the abbreviation ц︮р︦ь︯ д︮в︦д︦ъ︯ (Ца́рь Даві́дъ, *King David*). Reproducing such features of the orthography may also be necessary in academic publications.

First, it should be noted that existing Unicode codepoints intended for positioning diacritical marks over *two* base characters, for example, the Combining Double Macron (U+035E) and the Combining Double Tilde (U+0360), should **not** be used for this purpose. The Unicode Standard distinguishes between a Titlo and a Tilde or Macron, and the Tilde or Macron are not correct methods for encoding a Titlo. In addition, the Double combining marks have a combining class property of 234, not 230, which one must keep in mind for purposes of canonical reordering.

Instead, the Unicode Technical Committee has agreed that "Triple [and in our instances, higher order – *eds.*] marks should be handled by the mechanism associated with two-part diacritics in the Combining Half Marks block at U+FE20" (Irish NB and German NB, 2011). The characters encoded in the Combining Half Marks block are "presentation forms … used to visually encode certain combining marks that apply to multiple base letterforms" (Unicode Standard, p. 335). These marks are implemented in a way such that "a discontiguous sequence of the combining half marks corresponds to a single combining mark" (*ibid*). One common use for these marks is for supralineation in Coptic (Unicode Standard, p. 312).

Following this recommendation and given the existing implementation for Coptic, the UTC has accepted (see Andreev et al. (2013)) a similar approach for Cyrillic supralineation as used in Church Slavonic. In particular, two characters have been encoded in Unicode, a Cyrillic Titlo Left Half (U+FE2E) and a Cyrillic Titlo Right Half (U+FE2F). When a Titlo needs to balance over two letters, this is encoded in the following manner:

◌ + ◌︮ + ◌ + ◌︯ ⟶ ◌︮◌︯

When a Titlo needs to balance over three or more letters, the middle component or components are encoded using the character Combining Conjoining Macron (U+FE26), already used for Coptic supralineation. Thus, the words ц︮р︦ь︯ and д︮в︦д︦ъ︯ are encoded as follows:

ц + ◌︮ + р + ◌︦ + ь + ◌︯ ⟶ ц︮р︦ь︯
д + ◌︮ + в + ◌︦ + д + ◌︦ + ъ + ◌︯ ⟶ д︮в︦д︦ъ︯

The correct implementation of supralineation at the font level presents a serious challenge for the font developer. Presently, there is no simple approach to supralineation in OpenType in such a way that takes into account the different widths and heights of Church Slavonic glyphs, or the possible presence of other combining diacritics. We see two possible approaches.

In the first approach, the sequence of combining marks beginning with U+FE2E and ending with U+FE2F can be replaced with a single glyph for a double, triple, quadruple (or longer) titlo. This substitu-

[15] The authors would like to thank Laurențiu Iancu for helpful comments on this topic.

tion can take place via the *ccmp* feature in OpenType (in the substitution table, the "Ignore base glyphs" flag needs to be set) or via an appropriate substitution rule in SIL Graphite. Under this approach, problems occur with the positioning of the composed titlo glyph, as correct positioning needs to take into account both the different width and height of the base glyphs. In SIL Graphite, it is possible to write positioning rules that take into account the horizontal and vertical glyph metrics of the base glyphs and would thus correctly position the composed combining glyph. In OpenType, to our knowledge, this is not possible. Rather, it would be necessary to write contextual positioning rules that would determine the horizontal and vertical position of the composed glyph on the basis of the sequence of base glyphs. In practice, this becomes quite tedious as the number of glyph classes becomes large.

The second implementation approach is to create precomposed glyphs of the base characters with titlo halves of the appropriate height and width for each of the possible combinations of base characters. The correct precomposed glyph is then selected via the use of contextual substitution rules. This approach has the advantage that the order of glyphs is preserved. Since the precomposed glyph of the base letter and half mark can be given an appropriate glyph name in the font in accordance with Adobe Glyph List conventions, the correct codepoints in the correct order will be preserved under such operations as copying from a PDF document. Under the first approach, the correct order is not preserved, since, at the glyph level, all of the half marks are eliminated and replaced with a single mark that combines with the first base character. On the other hand, this approach is by far even more tedious than the first approach, as it requires the creation of several precomposed glyphs for at least every single letter of the Church Slavonic Cyrillic alphabet.

The authors feel that in the future, an extension to OpenType should be considered that allows the use of existing technologies for the correct "joining" of combining marks, for example, via the use of the *curs* (cursive attachment) feature. Presently, this is not possible. However, if this feature were extended to combining marks over different base glyphs, it would allow half marks to be joined together visually without resorting to glyph substitution and complex contextual rules. This would greatly simplify the implementation of half marks used in Church Slavonic or Coptic supralineation, as well as in other settings.

3.5 Palatalization

Palatalization is the phenomenon of softening certain consonants by pronouncing them with part of the tongue moved closer to the hard palate. Historically, certain consonants in Slavic languages have had palatal articulations and in ustav-era Slavonic manuscripts, one encounters graphical ways of indicating their softness. Graphical indicators of palatalization are absent from later Slavonic texts, and thus this topic is of interest only to scholars studying early manuscript, and to font developers, who should ensure that the correct implementation is used in fonts designed for academic work with historical texts.

Three methods of graphically indicating palatalization have been identified by scholars:[16] the use of iotified vowels (such as ꙗ, the iotified analog of а); the use of soft consonants (such as ꙥ, the soft analog of л); and the use of diacritical marks above or next to characters. All of the iotified vowels documented in Cyrillic sources have been encoded in Unicode and are identified in Section 2 (above). Some graphical variants of these iotified vowels have not been encoded as standalone characters and are properly treated using the methods discussed in the section below on Glyph Variants.

The diacritical mark that is used to indicate palatalization in ustav-era texts most often has the appearance of an inverted breve (◌̑), a comma (which may also be turned horizontally), or a hook-like character (◌҄). The character may appear directly over the letter or balance somewhat to the right (Golyshenko, 1987, pp. 36ff). For the purpose of representing this character, the Unicode standard includes U+0484 COMBINING CYRILLIC PALATALIZATION. This character should be provided in fonts as either an inverted breve or a hook or half-breve opening to the right. Though the usage of early Cyrillic diacritical marks has not

[16] The authors would like to thank Heinz Miklas for helpful comments on this topic.

been adequately researched and documented, one should, nonetheless, distinguish between the graphical character used for indicating palatalization and the kamora (the Cyrillic circumflex accent), which is encoded at U+0311 COMBINING INVERTED BREVE. We suggest that U+0311 should not be used to indicate palatalization; furthermore, the character at U+0484 should never be used to encode the kamora.

In addition to the character encoded at U+0484, a comma placed above or to the right of a character has also been identified by researchers as a graphical indicator of palatalization (Golyshenko, 1987, p. 38). When the comma needs to be placed above a character, the codepoints U+0313 COMBINING COMMA ABOVE and U+0314 COMBINING REVERSED COMMA ABOVE should be used (as in н̓ and н̔). Note that while U+0313 and U+0314 are used to indicate aspiration (breath marks) in Greek texts, the Cyrillic breath marks zvatelʹtse (psili) and dasia pneumata have been encoded separately at U+0486 and U+0485, respectively. The characters U+0313 and U+0314 should only be used to encode combining palatalization marks and should never be used to encode breath marks. (Note that, generally speaking, breath marks occur over vowels while palatalization marks occur over consonants). When the comma occurs to the right of a character, the spacing mark U+02BC MODIFIER LETTER APOSTROPHE should be used to encode this symbol (ʼ; see Yelkina (1960) for examples). This character should therefore be provided in all Slavonic fonts. A character of this shape may also be used in ustav manuscripts to indicate an omitted hard sign or soft sign, in which case it acts like a spacing yerik (ⸯ) or payerok (ꙿ), but has a different graphical appearance (Yelkina, 1960, p. 25).

Four Cyrillic soft consonants have been identified by researchers and three of these have been specifically encoded in Unicode. For the fourth (the soft En), a modern Cyrillic character – the Ligature En Ghe used in Altay, Mari and Yakut – should be used. While the Unicode Standard provides both lowercase and uppercase forms of these characters (in keeping with its convention for Cyrillic as a bicameral script), only the lowercase forms are attested. We summarize these characters in Table 9; note that ꙣ usually occurs only as part of the digraph жꙣ. While some fonts provide additional palatalized consonants of an analogous appearance (for example, a Soft Er or Soft Es), these are not known to exist in Slavonic writing, though they may be used by scholars in academic publications as hypothetical constructs (Golyshenko, 1987, p. 44). Because any additional such characters are not native to the writing system, we believe that they are best encoded in the Private Use Area of fonts intended for academic work (see below).

There is some disagreement between researchers about the origin of the soft consonants. Some scholars (for example, Kozlovsky (1885, p. 20) and Bulatova (1973, pp. 90ff)) consider the soft consonants to be ligatures of the main character and the combining palatalization mark (in other words, ꙥ = л + ◌҄) while others (for example, Golyshenko (1987, p. 42)) insist that they are standalone characters that should not be decomposed. In any case, these characters have been encoded in the Unicode Standard as standalones lacking any decomposition, and fonts and software implementations should not decompose these characters. In other words, the strings л҄ and ꙥ should be treated as two different character sequences, though under collation they may be treated as being different only at the second or third level of comparison (see below). It should also be pointed out that although ҥ has been equated in Unicode with the Ligature En Ghe used in some non-Slavic languages recorded with the Cyrillic script, neither this character nor any of the other soft vowels as used in Slavonic are ligatures consisting of г as a component. They should be treated as standalone characters, and difficulties arising in string comparison operations should be resolved with an appropriate collation tailoring.

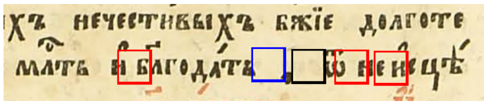
3.6 Spacing and Hyphenation

This section explains how various spacing characters available in Unicode should be used in typesetting Slavonic text. As can be seen in Figure 9, in poluustav typography, a full space is inserted both before and after punctuation, making all punctuation marks (including the Slavonic Asterisk) to appear to be balanced

Table 9: Palatalized Consonants used in Church Slavonic

Base Character			Soft Character		
Name	Codept.	Rep.	Rep.	Codept.	Name
Cyrillic Letter De	U+0434	д	ꙣ	U+A663	Cyrillic Letter Soft De
Cyrillic Letter El	U+043B	л	ꙥ	U+A665	Cyrillic Letter Soft El
Cyrillic Letter Em	U+043C	м	ꙧ	U+A667	Cyrillic Letter Soft Em
Cyrillic Letter En	U+043D	н	ҥ	U+04A5	Cyrillic Ligature En Ghe
Characters commonly used with U+0484 Combining Palatalization: г҄ к҄ л҄ н҄ р҄ х҄					

Figure 9: Different kinds of spaces used in Slavonic texts. Note the usage of the Space (boxed in black), the No-break Space (boxed in blue), and the Narrow No-break Space (boxed in red). Source: Menaion, Moscow, 1645.



between two words. There is almost never an instance in the authentic poluustav-period typography when no space is inserted before punctuation. On the other hand, in Synodal typography, just as in modern Russian text, spacing before punctuation is generally not used. The only exception to this rule is the Slavonic Asterisk, which is also usually separated from the word before it by a full-width space.

The common space character used in typography is encoded in Unicode at U+0020 SPACE. This is the standard spacing character that should used in the vast majority of instances. It is a “breaking” character, as it allows text to flow to the next line. For this reason, U+0020 should not be inserted before punctuation marks or the Slavonic Asterisk, as it would incorrectly allow the punctuation character to flow onto the next line. The correct character to use before the Slavonic Asterisk or before a punctuation mark in poluustav typography is U+00A0 NO-BREAK SPACE (NBSP). This character has the same width as the character U+0020 but does not allow a line break at the point of its placement in the text. We note that besides including U+00A0 in fonts (and making it the same width as U+0020), there is no special functionality that font developers need to assign to the character; the line breaking behavior is specified in the Unicode character properties, and text rendering systems should honor the specifications of the Unicode Line Breaking Algorithm. Finally, in addition to its function as a non-breaking space, U+00A0 is also used for the display of combining marks in isolation (see above).

In addition to the use of a full-width, non-breaking space, poluustav typography also makes use of a narrower, also non-breaking, space character. This character is used either by typographic convention as a space-saving device or as described by orthographic rules to assist in the syntactical flow and cohesion of sentence structure by linking together words and fragments (for example, before some pronouns or after some unaccented particles). This is similar to spacing conventions in Mongolian typography. The codepoint used for this narrow space is U+202F NARROW NO-BREAK SPACE (NNBSP). This character should also be provided in all fonts, and should have about one-half or one-third of the width of U+0020 SPACE.[17] It is also treated by the Line Breaking Algorithm as a non-breaking space.

It is difficult to formulate exact rules for the use of the narrow no-break space since there is always a certain degree of idiosyncratic usage in poluustav typography. However, some general patterns of usage may be observed, and we recommend that they should be followed in preparing digital texts for re-publication. We summarize these patterns of usage in Table 10.

[17] No clear rules for the width of this character are specified in Slavonic books; one-half width is a general rule-of-thumb and one-third is the standard width of the *espace fine insécable* used around punctuation characters in French typography.

Table 10: Rules for the usage of non-breaking spaces in poluustav typography

Rules	Examples
1. between an abbreviated word ending with a letter titlo and all punctuation marks (including the Slavonic Asterisk). This is a space-saving convention.	на́шеⷢ҇ ., глаⷭ҇ ,
2. between a Slavonic numeral and a punctuation mark. Note that numerals are generally set off by punctuation marks before the numeral, and a narrow space and period following the numeral. The space before the numeral should be full-width, non-breaking space.	д҃ ., глава̀ , ѕ҃ .
3. before unaccented clitics and pronoun particles. Examples: всѧ́ же, вельмѝ бо. However, if the particle is accented, then it is not preceded by a NARROW NO-BREAK SPACE, as the expression спасѝ ны̑.	Clitic particles: бо бы же ли либо; Unaccented pronouns: ми мѧ ти тѧ си сѧ ны вы
4. after unaccented enclitics and particles, particularly between a preposition and the word it modifies. Prepositions as a rule had no indicated stress in poluustav-era Slavonic, unless they borrowed the stress from the following word as a syntactical feature. Examples: въ ни́хъ , со сла́вою , до конца̀. (Note that the conjunctions и, но, а, или are not included in this category, while the archaic да [when used as “and”] is included. However, when the narrow і҆ is used as a space-saving device, it is followed by narrow no-break space by convention: да бг҃ъ , і҆ посе́мъ)	безъ , без꙽ , беⷥ҇ , безо , въ , в꙽ , во , да , до , за , (і҆), и҆зъ , и҆з꙽ , и҆, и҆зо , къ , к꙽ , ко , на , надъ , над꙽ , наⷣ , надо , не , ни , ѡ҆ , ѡ҆бъ , ѡ҆б꙽ , ѡ҆бо , ѿ , ѡ҆тъ , ѡ҆то , по , подъ , под꙽ , поⷣ , подо , предъ , пред꙽ , преⷣ , предо , при , съ , с꙽ , со , ᲂу҆ , чрезъ , чрез꙽ , чреⷥ҇ , чрезо
5. The NARROW NO-BREAK SPACE is also inserted whenever an accent is displaced from one word to another for grammatical reasons or to form an idiomatic expression. This occurs most frequently with prepositions that take their stress from the noun they modify, but other examples may likewise be found.	ко́ г҇дꙋ́ , со́ страхомъ, по всѧ́ дни

In order to facilitate correct typography of poluustav texts, standard keyboard input methods for Church Slavonic should provide easy access to these spacing characters. In the keyboard layouts we discuss below, the no-break space and the narrow no-break space are provided as ⇧ Shift Space and AltGr Space, respectively.

In addition to these characters, the Unicode Standard encodes a number of other spacing characters. These include the various fixed-width space characters U+2000 through U+2006 as well as U+2009 THIN SPACE and U+200A HAIR SPACE. These characters only differ in their width and provide standard spacing characters as used in traditional (hot lead) typography. The width of these characters is determined by the font and typically is not adjusted by text justification algorithms. Since modern (Synodal) Slavonic typography can take advantage of algorithmic justification, there is no need to use these characters in modern texts. However, they may be used in reproductions of poluustav-era texts where spacing needs to be strictly defined at the encoding level, and thus we recommend that fonts designed for poluustav typography provide these characters. Finally, U+200B ZERO WIDTH SPACE may be used to indicate word boundaries in ustav-era text that does not use visible inter-word spacing in order to allow for line-breaking and morphological analysis; this character should also be included in fonts, and should be provided as a zero-width character with no visible glyph. Note that U+200B is a breaking character, and its presence will allow a line-break between the two words. In situations where two words appear next to each other without a visible space and a line break is not desired (for example, between a clitic and a word it modifies), the character U+2060 WORD JOINER, which prohibits a line break, should be used instead.

A few words are warranted about Slavonic hyphenation. Liturgical texts are most often typeset with justified alignment and when text justification is used, hyphenation (the transfer of part of a word to the next line) is a modern typographical norm. This issue is particularly significant in the digital presentation of Slavonic text, especially on mobile devices that have small effective screen size. Long compound words (which frequently occur in Slavonic) may need to be hyphenated in these instances to make the presentation of text aesthetically pleasing. Generally speaking, the typesetting of modern (Synodal-era) texts should take advantage of algorithmic hyphenation, a process where the software automatically determines the appropriate hyphenation points on the basis of language-specific rules (patterns). The topic of algorithmic hyphenation of Slavonic is beyond the scope of this paper and is discussed by the authors in a separate, forthcoming study.

In typesetting poluustav-era texts, on the other hand, a more cautious approach to hyphenation should be observed. As a general rule, in the early texts, hyphenation was not used and in reproducing such texts, hyphenation should be avoided. In some of the later texts, hyphenation does exist, and a part of the word may be transferred to the next line, even though no hyphen character is printed at the end of the line, since hyphen characters were generally not used in poluustav typography. In our view, because the emerging hyphenation of these texts is of scholarly interest in and of itself and because potential users of such typeset text may have a requirement that the typeset edition of a sacred text reproduce the autograph as closely as possible, existing hyphenation should be preserved, but no new hyphenation (including any algorithmic hyphenation) should be introduced. To indicate existing hyphenation in a text, the character U+00AD SOFT HYPHEN should be inserted at the hyphenation point. The Soft Hyphen is not a hyphen character, but rather a control code that is used to indicate to software an intraword break at this point. Font developers should provide U+00AD SOFT HYPHEN in their fonts as a zero-width character with no visible glyph. Text rendering software may break a word at the Soft Hyphen if necessary and specialists studying hyphenation can easily analyze break points so indicated. On the other hand, one should not simply place a line break in the middle of a hyphenated word as this practice breaks the flow of text and prevents word searching and other algorithms from correctly identifying the broken word. For more information on spacing and hyphenation characters available in Unicode and their properties, consult Section 6.2 of the Unicode Standard (pp. 266ff).

3.7 Glyph Variants

Many Church Slavonic characters may have several visual presentation forms (“glyphs”). The choice of which glyph is selected to present a given character may be governed by typographic convention, by rules of orthography and semantics, or it may be haphazard. In the manuscript tradition, one finds a greater amount of diversity of presentation forms than in the later print tradition, since rules governing the usage of glyphs became more standardized with time. Generally, modern Synodal Slavonic has very rigid typographic and orthographic norms and uses fewer variant glyph forms than earlier recensions.

As a policy, the Unicode Standard encodes characters, not individual glyph shapes. The term “character” is defined in Unicode as the “smallest unit of a writing system that has semantic value.” Nonetheless, this definition is somewhat vague, as the term “semantic value” can be interpreted quite broadly. In fact, a number of variant glyph forms have been encoded in the Unicode Standard as standalone characters and are thus considered “characters” as far as encoding is concerned. For some of these, usage in the manuscript tradition is governed by scribal whim or complex grammatical rules (for example, word etymology), and so it cannot be predicted algorithmically on the basis of contextual rules. In such instances, the desired glyph must be explicitly selected at the codepoint level. This is the case with various palæographic variants, like the glyph ‘ꙃ’ used in ustav writing, or with the use of the broad form of Omega (ꙍ) in words of a Greek origin in poluustav typography.

In other instances, usage may in fact be governed by contextual rules, at least to some degree or in some sources. For example, in Synodal Slavonic, the Narrow On (ᲂ) occurs only before у; in Kievan Synodal-era editions, the form of the letter Dobro with long legs (ᲁ) occurs only in medial position. However, in other settings (for example, in poluustav type), these later orthographic conventions are not observed, and the usage there is not entirely predictable. Thus, these glyph forms, which also have been encoded in Unicode as standalone characters too must be selected explicitly at the encoding level. To put it simply, one must not use contextual substitution to change the character о (U+043E) to the character ᲂ (U+1C82) in the digraph ᲂу; rather, whenever the first component of the digraph is the Narrow On (which is almost always the case), the digraph must be explicitly encoded as U+1C82 U+0443. From the standpoint of encoding, U+043E and U+1C82 are two completely different characters. On the other hand, in comparing two Slavonic strings, these characters may, in fact, need to be treated as being closely related (or even identical) in many instances; the issue of string comparison is addressed via the use of collation (see below).

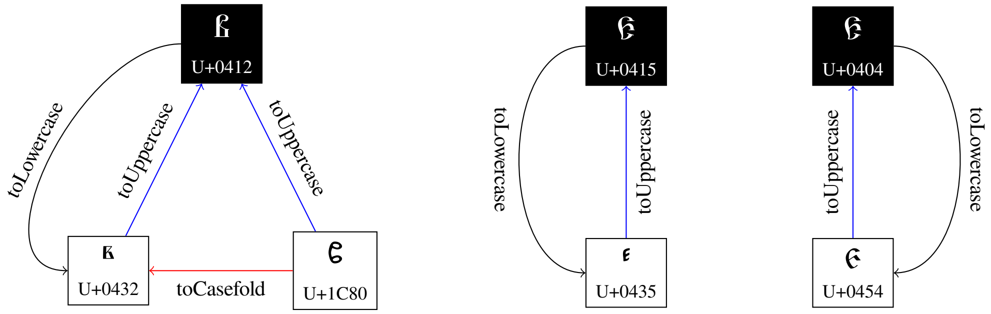
While contextual behavior must not be used to select the glyph form at the font level, contextual behavior may be used by the input method to select the correct character for a text stream generated by keystrokes. In this instance, for example, the same keystroke may be mapped to U+0434 in some instances and to U+1C82 in other instances; the mapping may be based on contextual rules. To provide for a method to override the default mapping, a special key may be set up to override the default input method behavior in settings where it yields undesired results. While such functionality is not available in basic input method systems, it is possible to implement in more advanced systems, such as `IBus`; however, it is beyond the scope of this paper.

Case Folding

Note that since Cyrillic is a bicameral script, the Unicode Standard has, generally speaking, encoded both an uppercase and a lowercase form of each character, even for those characters that exist only in ustav-era manuscripts where casing distinctions are not well-defined. There are, however, some exceptions. Some variants that do not have attested uppercase forms resolve under casing to the uppercase form of the related “base” character. When this uppercase form is converted back to lowercase, the “base” form, not the variant, is the result. This process is called “case folding”, a mechanism designed for the caseless matching of strings. Under this mechanism, some characters that are considered variants of each other are specifically designed to fold to the same character under the `toCasefold(·)` operation, as visualized in the left pane of Figure 10. The character U+1C80 CYRILLIC SMALL LETTER ROUNDED VE does not have an uppercase representation but folds to the “base” character U+0432 CYRILLIC SMALL LETTER VE. In the right pane of Figure 10, we see two characters that are variants of each other, but each of which has its own uppercase form. Note that the uppercase form of U+0454 CYRILLIC SMALL LETTER UKRAINIAN IE (used to represent the wide Yest in Church Slavonic) is U+0404 CYRILLIC CAPITAL LETTER UKRAINIAN IE and not U+0415 CYRILLIC CAPITAL LETTER IE, even though in fonts designed for Synodal Slavonic, U+0404 and U+0415 should have exactly the same visual appearance.

The case folding mechanism allows for string comparison involving variants without implementing collation, though since not all variant forms used in Church Slavonic fold onto each other, this may not be particularly useful. Note that such treatment of variants is not unique to Cyrillic; for example, in Greek the character U+03C2 GREEK SMALL LETTER FINAL SIGMA (ς) folds onto U+03C3 GREEK SMALL LETTER

Figure 10: Casing relationships between Cyrillic variants in Unicode



SIGMA (σ) and in the Latin script, the character U+017F LATIN SMALL LETTER LONG S (ſ) folds onto U+0073 LATIN SMALL LETTER S (s). However, one should keep in mind that because of the way that the Cyrillic characters have been encoded in the Unicode standard, the implementation of casing is not consistent. Note also that the character U+A66E CYRILLIC LETTER MULTIOCULAR O (ꙮ) is encoded as completely caseless.

Contextual Forms and the PUA

The glyph variants that have been encoded as standalone characters in Unicode are presented in Table 11. Additional presentation forms do exist that have not been encoded. Generally, this is true in cases where the usage of a glyph is completely predictable on the basis of context and thus can be relegated to the rendering system, usually via contextual substitution rules in OpenType as is discussed in the section on Font Design and Development. For example, whenever a superscript letter occurs over the character Uk (ѹ) in Synodal Slavonic, the character changes its shape to the truncated form ꙋ (as in ꙋⷯ). Since in this case, this change is governed entirely by contextual rules, no separate ꙋ character has been encoded in Unicode. The selection of the correct glyph is performed by the rendering system on the basis of glyph shapes and writing system rules available in a "smart" font that implements OpenType or SIL Graphite features. Nonetheless, instances may arise when this glyph form needs to be selected outside of its normal context. For example, the authors had to present the shape ꙋ as a standalone character in writing the present sentence. It may also be necessary to select the desired glyph shape in settings where use of "smart" font technologies or OpenType-aware rendering systems is not possible or not practical, either on legacy systems or in certain computer programming situations where software needs to access glyphs directly. For these reasons, we highly recommend that font manufacturers make these contextual glyph forms available in the Private Use Area of Unicode.

The PUA is a set of blocks in Unicode that are not presently allocated to any characters and are guaranteed to remain unallocated by the Unicode Consortium. Instead, they are specifically intended for use by third parties to be assigned to characters used internally. Providing access to glyph variants in the PUA allows for them to be used in instances as described above without conflicting in any way with the existing encoding scheme. We emphasize that selecting characters from the PUA by codepoint is not a desired practice, and generally speaking, users should not place any PUA characters in documents or data intended for interchange. The correct selection of a glyph shape should always be relegated to the rendering system whenever this is possible. Use of the PUA should only take place in situations where such glyphs need to be accessed in a software / platform setting where advanced font features are not available or where glyphs need to be manipulated directly by computer software (and not by the end user) in ways that do not rely on advanced font features. In any case, since all possible glyph shapes are mapped in fonts anyway (typically outside of the Unicode range), we suggest mapping them to codepoints in the PUA to make accessing these glyphs at the codepoint-level possible. The PUA is also the place where various "special" characters should be encoded. These include nonce glyphs, hypothetical constructs used by specialists, or any other glyphs that are not deemed part of the Cyrillic writing system and not intended to be transmitted in data exchange and thus will not be encoded in Unicode.

While the Private Use Area is by definition "private" and thus not subject to standardization, nothing prevents industry members from reaching an internal consensus concerning its allocation, though there is no explicit mechanism in place for this to be achieved. However, the authors have adopted a Private Use Area Policy, which specifies which glyph forms are mapped to which codepoints in the PUA for fonts manufactured by the authors, and propose that other manufacturers of Slavonic fonts adopt or build upon this policy to ensure compatibility. In addition to glyph forms, the authors also encode in the PUA certain characters that have not yet been formally proposed for addition to the Unicode repertoire. Providing places in the PUA for these characters allows data interchange to take place in a somewhat standardized manner while the necessary documentation and consensus is gathered for the final encoding of these characters.

One important point to keep in mind: following the Adobe Glyph List standard, the glyph names for characters encoded in the PUA should not be the codepoint-based names but rather the name of the base glyph, followed by a dot, followed by an identifier. Thus, the above-mentioned variant for Uk (ꙋ), encoded according to the authors' PUA Policy at U+E0E4, should be named `unia64b.short` and not `unie0e4`. This naming convention facilitates correct behavior of fonts within PDF documents and other software relying on the Adobe Glyph List. See the section on Font Design, below, for more information. For a comprehensive listing of the additional presentation forms, as well as ligatures and other glyphs not encoded in Unicode, and other policies related to PUA allocation and usage, we refer the reader to Andreev et al. (2015a).

3.8 Ligatures

Broadly defined, a ligature is a glyph created by joining two or more characters. In modern writing systems, ligatures sometimes occur as discretionary forms; for example, in English it is not uncommon to represent the letters f and i side-by-side as the ligature ﬁ. With the exception of some digraph combining marks, there are no ligatures in Synodal Church Slavonic. However, ligatures abound in the poluustav tradition, where they are used as space- or ink-saving devices or decorative features. These ligatures are usually formed by joining two characters; less often – three or more characters.

Since ligatures are not standard in Church Slavonic writing (that is, no letters are ligated by default, as in Arabic or Devanagari), they are appropriately handled via the use of U+200D ZERO WIDTH JOINER. This format character, which should be included in fonts as a zero-width, invisible character, requests that two adjoining characters, which do not usually form a ligature, be formed into a ligature. At the font level, the rendering is handled via the use of such features as the Glyph Composition / Decomposition feature of OpenType. The following examples are illustrative:

л + ѵ ⟶ лѵ (standard behavior)
л + ‍ + ѵ ⟶ л‍ѵ (ligature)

In some instances (mostly for demonstration purposes), it may be required to represent the (contextual) ligature forms of characters side-by-side without forming a connected ligature. In the case of Arabic and Devanagari, Unicode specifies the encoding sequence ‍‌‍, where ‌ is the format code U+200C ZERO WIDTH NON-JOINER, which invokes the contextual forms of the characters without connecting them into a ligature. However, this approach does not work for Cyrillic since there are no predefined contextual

Table 11: Glyph variants encoded in Unicode

A dagger (†) by a codepoint indicates that this character casefolds to the base form. An asterisk (*) indicates that this character does not have case.

Base Form	Variant Forms
U+0432 (в)	U+1C80 (ᲀ)†
	In incunabula, the Rounded Ve form may occur haphazardly or following some contextual rules (for example, with certain diacritical marks). Either it or the base form may be the predominant form in the text.
U+0434 (д)	U+1C81 (ᲁ)†
	The long-legged form occurs in texts of a Kievan provenance in the medial position while the base form occurs in initial position. However, in some instances the respective forms are used outside of context and in poluustav text, the usage may be haphazard.
U+0435 (е)	U+0454 (є) U+044D (э)
	In Synodal Slavonic, U+0454 is used whenever the letter Yest occurs in initial position, but also in non-initial position to make grammatical distinctions. In poluustav and ustav recensions, the usage may be haphazard. U+044D is used in some early texts as a reversed variant of Yest.
U+0455 (ѕ)	U+A654 (ꙅ)
	The reversed variant U+A654 occurs only in ustav manuscripts.
U+0437 (з)	U+A641 (ꙁ) U+A643 (ꙃ)
	The form of Zemlya encoded at U+A641 is used in Kievan editions in non-initial position. Despite the fact that in Unicode, a capital form U+A640 has been encoded, such a capital form is not used in the print tradition. The form encoded at U+A643 occurs only in ustav manuscripts.
U+043E (о)	U+047B (ѻ) U+1C82 (ᲂ)† U+A669 (ꙩ) U+A66B (ꙫ) U+A66D (ꙭ) U+A66E (ꙮ)* U+A69B (ꚛ) U+A699 (ꚙ)
	The wide form U+047B is used in Synodal Slavonic and in late poluustav recension texts in initial position or in medial position as the first letter of a compound word. The narrow form U+1C82 is used in Synodal Slavonic only in the digraph ᲂу, but occurs frequently in poluustav text, usually when the letter is not accented. The other variants are graphical embellishments that may be found in manuscripts (see Section 2 for more information).
U+0441 (с)	U+1C83 (ᲃ)†
	The wide form U+1C83 is used in Kievan editions and in some poluustav texts in initial position to indicate that a word refers to God.
U+0442 (т)	U+1C84 (ᲄ)† U+1C85 (ᲅ)†
U+A64B (ꙋ)	U+0443 (у) U+1C88 (ᲈ)†
	In Synodal Slavonic, the form у is used only as part of the digraph ᲂу or as a numeral. In earlier recensions, it may be used as either a variant of ꙋ or as a variant of ѵ. The "unblended" form U+1C88 occurs in some poluustav texts of a Polish-Lithuanian provenance. This letter also has a large number of contextual variant forms, which have not been encoded in Unicode as standalone characters.
U+0461 (ѡ)	U+A64D (ꙍ)
	In Synodal Slavonic, the broad form U+A64D is only used as the base in writing the exclamation ѽ, but the character for ѽ has been encoded in Unicode separately at U+047D. In poluustav texts, the broad form was also used in medial position in words of a foreign origin.
U+0446 (ц)	U+A661 (ꙡ)
	The reversed variant U+A661 occurs only in ustav-era texts.
U+044A (ъ)	U+1C86 (ᲆ)†
U+044B (ы)	U+A651 (ꙑ)
	The variant form U+A651 occurs only in ustav-era texts.
U+0463 (ѣ)	U+1C87 (ᲇ)†
U+044E (ю)	U+A655 (ꙕ)
	The reversed variant U+A655 occurs only in ustav-era texts.

forms. Rather, the necessary glyph must be selected explicitly at the codepoint level. In certain instances, the contextual form has not been encoded as a standalone character and thus the glyph would need to be selected from the Private Use Area, as discussed above.

т + ZWJ + к → ᲄк (ligature)
ᲄ + ZWJ + к → ᲄк (graphically equivalent)
ᲄ + к → ᲄк (unligated contextual forms)
т + ZWJ + ZWNJ + ZWJ + к → тк (incorrect)

Consult also Section 23.2 of the Unicode Standard for more information (Unicode Standard, p. 814f).

4 Font Design and Development

The Unicode Standard addresses the issue of character encoding, listing the repertoire of graphical elements used in a script and providing numerical codes for their representation, but users work with actual characters, not with numerical codes. Thus, once we have defined and implemented an encoding model for Church Slavonic, we are in need of a font that allows working with characters within the model's framework. The present section addresses some issues associated with the design of such "conformant fonts".

At the basic level, a font consists of glyphs that have been drawn by the font designer and rules that inform the rendering system how these glyphs are to be manipulated. We first discuss the selection and design of glyphs, which follows directly from the character repertoire identified in Section 2, and then the programming of writing system rules, which follows in turn from the discussion of implementation in Section 3.

4.1 Font Design and Distribution

A font is defined as a quantity of sorts (glyphs) composing a complete character set of a single size and style of a particular typeface. Note that by definition, fonts are character set and typeface specific. Therefore, one cannot create a single Church Slavonic font.[18] Rather, a font should only provide a typeface that reproduces one particular recension of Church Slavonic writing. A typeface may attempt to reproduce the features of one of the recensions identified in Section 1: ustav writing, poluustav writing or type, Incunabula type, Synodal type, or skoropis. Note that for purposes of typeface design, the poluustav recension can be further subdivided into manuscript poluustav, Old-style book poluustav (the vertically expansive printed type style found in the early editions of a Polish-Lithuanian provenance) and New-style book poluustav (the more vertically conservative type style of the early editions of Ivan Fedorov and the Moscow Printing Court throughout the pre-Nikonian era). In addition to these recensions, we can also identify two further categories: decorative Church Slavonic fonts intended for use for *bukvitsy* (drop caps) and *vyaz´* (Cyrillic calligraphy) and modern Cyrillic typefaces. The design of decorative fonts adds a number of complexities because of the abundance of character variants, stylistic alternatives, and ligatures, and so this topic is also beyond the scope of this document. As for modern Cyrillic fonts, they should provide glyphs (including ancient glyphs) in the typeface of modern Russian or other Slavic languages. Obviously, this is anachronistic, but is nonetheless necessary for use in academic publications, computer software, and other settings. Such modern fonts should provide, by convention, a serif, sans-serif, and monospace typeface.

[18] Strictly speaking, the use of stylistic sets and other advanced features allows the inclusion of multiple typefaces into a single font, but this is beyond the scope of this document.

We have identified the character repertoire of each recension of Church Slavonic in Section 2. This is the basic repertoire necessary for a recension-specific font. The inclusion of other characters not relevant for a particular recension (for example, the inclusion of a glyph for the modern Cyrillic Letter Ya in a Synodal-era font) may be desirable, for example, for use in creating ornamented Russian or Ukrainian text, but is in no way required for a conformant font. On the other hand, conformant fonts must provide the entire repertoire of characters associated with the recension for which they are designed. Thus, for example, a Synodal-era font lacking a Big Yus could not be considered a conformant font. Note that the shapes presented in the Unicode codecharts are usually the modern character shapes, but in the tables in Section 2 we provide the period-specific glyph shapes.

Font designers need to clearly specify the recension of Church Slavonic for which their font is intended. This may be done either in the font name by including the recension (for example, an ustav-era font may be called "*Name* Ustav", where *Name* is the name of the typeface) or in accompanying documentation. In addition, font designers need to address in the documentation, as a minimum, the following issues: who is the author of the typeface; under which license (or licenses) is the font distributed; what is the repertoire of glyphs available in this font; and which advanced typographic features are used by the font. This is to address possible misunderstandings about fonts and their distribution. A font, like any piece of software, is intellectual property that may be covered by applicable copyright law. Many users incorrectly believe that a font may be freely downloaded, opened, edited, and distributed; but, in fact, some of these actions may be copyright violations. Just because a font's binary files may be loaded into font editing software does not imply that it is legal or ethical to do so. On the other hand, many font developers incorrectly believe that simply publishing their font's binary files on a webpage implies that users are free to edit and redistribute the font. However, designers who wish to provide their fonts as free software must explicitly do so by indicating in the documentation that the font is distributed under an appropriate "free software" license.

Two such free software licenses are currently popular: the SIL Open Font License (in version 1.1 as of this writing) or the GNU General Public License (in version 3 as of this writing). Distributing their font under one (or both) of these licenses means that font designers not only contribute to the noble cause of free software but also are able to guarantee recognition of their intellectual rights (via attribution) and protection from potential liability should a defective font result in data loss or other damage. If distributing fonts under these licenses, font authors should also distribute the "source code" files (for example, the FontForge `sfd` file), where appropriate, as these files contain additional data useful for modifying the font. Also, it is necessary to keep in mind that these licenses have some stipulated conditions. For example, the GNU General Public License does not permit embedding the font in a document unless that document is also licensed under the GPL. This prohibits the use a font so licensed in a PDF document that uses font embedding. If a font designer wishes to allow font embedding, he must specify that the font is licensed with the GPL Font Exception.[19] The SIL Open Font License does allow font embedding, but indicates that if a font is sold, it must be packaged with computer software.[20] The authors strongly recommend the use of one or both of these licenses for font distribution, however, since the authors are not legal experts and this technical paper is not a legal document, none of these statements should be construed as legal advice.

4.2 Character Repertoire

In addition to the relevant repertoire for the appropriate recension of Church Slavonic, as given in Section 2, conformant fonts need to include the following characters for proper rendering and compatibility

[19] The documentation on the Free Software Foundation's website is useful in understanding these, and other, font copyright issues.

[20] See the SIL International website for more information. A *Hello, world!* program may be sufficient to fulfill this condition.

with various software applications, including for reverse compatibility with previous font or encoding standards.

For the purposes of correct implementation of the Church Slavonic typographic features described above, fonts need to include glyphs for the Zero Width Joiner (U+200D); Zero Width Non-Joiner (U+200C); Combining Grapheme Joiner (U+034F); No-Break Space (U+00A0); Narrow No-Break Space (U+202F); Soft Hyphen (U+00AD); and Dotted Circle (U+25CC). For Zero Width Joiner, Zero Width Non-Joiner, Combining Grapheme Joiner, and Soft Hyphen, fonts should have zero-width, non-visible glyphs. The Dotted Circle should posses a complete set of anchor points required to correctly implement positioning of diacritical marks for demonstration purposes. The character No-Break Space should have the same right extent as the character for Space (U+0020); it is used for the rendering of diacritical marks in isolation. The inclusion of other space characters discussed in Section 3 is also recommended.

For the purposes of compatibility with Apple's original TrueType specification, fonts should contain the following four glyphs as the initial four glyphs of the font. A glyph named *.notdef* at glyph id 0 (Unicode value undefined); a glyph named *.null* at glyph id 1 (Unicode value U+0000); a glyph named *CR* (carriage return) at glyph id 2 (Unicode value U+000D); and a glyph named *space* at glyph id 3 (Unicode value U+0020). Failure to include these glyphs may result in bizarre behavior in certain applications. As well, the *.notdef* glyph is important in providing information to the user that a given glyph is not available in the font. It is recommended that the shape of the *.notdef* glyph be either an empty rectangle, a rectangle with a question mark inside of it, or a rectangle with an "X". Font designers should not use creative shapes or leave this glyph without an outline as the user may not recognize such a glyph as an indication that a desired glyph is missing from the font. In addition to these glyphs, it is highly recommended that fonts contain all of the ASCII control codes encoded in Unicode between U+0000 and U+001F, inclusive. Microsoft Corporation makes the following additional recommendations concerning these glyphs:

- Glyph 1 (*.null*) should have no contours and zero advance width.
- Character U+000D (carriage return) should map to a glyph with a positive advance width.
- Characters U+0001 through U+001F (miscellaneous ASCII control codes) and U+007F (delete) should be mapped to glyph 0 (with some exceptions noted below).
- Characters U+0000 (*.null*), U+0008 (backspace) and U+001D (group separator) should map to glyph 1.
- Characters U+0009 (horizontal tabulation), U+0020 (space) and U+00A0 (no-break space) should map to a glyph with no contours and a positive advance width.
- Characters U+0009 (horizontal tabulation) and U+0020 (space) should map to a glyph with the same width (Microsoft Corporation, 2010).

It is also suggested that fonts include zero-width, blank glyphs for Left To Right Mark (U+200E) and Right To Left Mark (U+200F). While Church Slavonic is written exclusively from left to right, the presence of these glyphs in a font will facilitate correct behavior in multilingual applications.

As well, it is highly desirable for fonts to present the basic repertoire of Latin characters. At a minimum, all OpenType fonts for Church Slavonic should include the characters between Unicode codepoints U+0020 (space) and U+003F (question mark). These characters provide glyphs for European numerals and Latin punctuation required for many technical tasks. As well, evidence strongly suggests that certain software requires the presence of these characters for proper typography. Font designers can either design glyphs that stylistically resemble the Church Slavonic glyphs of the font or include glyphs from a standard font licensed under a licensing scheme compatible with the new font; in either case, these glyphs should not be left blank or mapped to the *.notdef* or *.null* glyph.

While the inclusion of all 95 graphical ASCII characters is desirable, it is probably not required for proper functionality. However, Microsoft Corporation does recommend the inclusion of the following glyphs for compatibility with Microsoft Office products: the Euro (U+20AC); all glyphs in the range U+0020 to U+003F; and the glyphs U+00A0 (No-break Space), U+00A4 (Currency Sign), U+00A7 (Section Sign), U+00AC (Not Sign), U+00B0 (Degree Sign), U+00B6 (Pilcrow Sign), U+00B7 (Middle Dot), U+2022 (Bullet) and U+2219 (Bullet Operator). In general, it is advisable to follow Microsoft recommendations to guarantee proper functionality with Microsoft products (Microsoft Corporation, 2010). We add to this list the symbols for the Russian Ruble (U+20BD) and the Ukrainian Hryvnia (U+20B4), since many font users will be using systems localized for the Russian Federation or other Eastern European countries.

Finally, fonts should not use Unicode codepoints to provide any characters other than the characters encoded in Unicode at those codepoints and must not define or use variation sequences not specifically defined in the Unicode Standard. Note that variation sequences are not a general encoding extension mechanism but rather are also subject to standardization. Any characters or variants available in a font that are not mapped in the Unicode Standard should be encoded outside of the Unicode range and accessible via advanced typographic features or should be placed in the Private Use Area (see Andreev et al. (2015a) for more information).

4.3 Glyph Names

Font designers need to adhere to a set of rules developed by Adobe Systems governing the naming of the glyphs in a font. Failure to follow this Adobe Glyph Naming convention will result in improper functionality in a variety of contexts, particularly when working with Portable Document Format (PDF) files.

The Adobe Glyph Naming convention is a mapping for "converting the sequence of glyphs to plain text while preserving the underlying semantics". For example, a glyph for "A" and a glyph for "small capital A" and a glyph for a swash variant of "A" will all be mapped to the same Unicode value. This is useful in copying text in some environments, and is useful for doing text searches that will match all glyphs in the original string that mean "A" (Adobe Systems, 2007). This is particularly relevant for Church Slavonic text, which contains glyph variants and ligatures.

In general, a glyph in a font should be named in one of the following ways. If the glyph is listed in the Adobe Glyph List, then the Adobe Glyph List (AGL) name may be used. Otherwise, the glyph should named using the prefix *uni* and a sequence of four hexadecimal digits (0 though F) representing the Unicode codepoint or using the prefix *u* and a sequence of four to six hexadecimal digits representing the Unicode codepoint. Thus, valid glyph names for the Cyrillic Capital Letter A (U+0410) would be *Acyrillic* (the name in the AGL), *uni0410*, or *u0410*. Note that calling the glyph *A* would result in its being mapped to the Latin letter A (U+0041) while calling the glyph *Az* or *uni410* would result in its being mapped to the empty string as these are not valid glyph names. For the Combining Cyrillic Letter A (U+2DF6), valid glyph names are *uni2DF6* or *u2DF6*, as this glyph is not in the Adobe Glyph List. For the Symbol for Mark's Chapter (U+1F545), the only valid glyph name is *u1F545*. For the sake of simplicity, the authors of this paper recommend not using the names in the Adobe Glyph List, but rather using the *uniXXXX* form for all characters in the Basic Multilingual Plane (BMP) (that is, in the range U+0000 to U+FFFF) and the form *uXXXXX* for all characters in the Supplementary Multilingual Plane (SMP) (that is, U+10000 to U+1FFFF). There are presently no characters of interest to Church Slavonic in higher planes, except perhaps Private Use characters.

Variants in the Adobe Glyph Naming convention are designated by a descriptor separated from the base glyph name by use of a Period (U+002E). The glyph name mapping drops all characters starting with the first occurrence of the Period. Therefore, *uni0410*, *uni0410.variant*, and *uni0410.tall* all describe variants of the character Cyrillic Capital Letter A and will all be mapped to the same Unicode codepoint U+0410. This allows, for example, for a search query for the character U+0410 to return all variant forms in the document.

Glyphs that represent multiple characters – ligatures – are named by joining their components. For example, a precomposed glyph of Cyrillic Capital Letter A (U+0410) and Combining Acute Accent (U+0301) should be named *uni04100301*, which would map the glyph to the appropriate character string U+0410 U+0301 (in that order). Note that a precomposed glyph of the variant Cyrillic Small Letter O (U+043E) and Combining Acute Accent (U+0301) should be named *uni043E0301.variant*, not *uni043E.variant0301* as all characters starting with the period are ignored in the mapping. Alternatively, it is possible for components in the name to be separated by use of an underscore, as *uni043E_uni0301.variant*. This is the only possible approach for naming ligatures composed of characters in the SMP, for example, *u1F545_uni0301.alternate*. Ligatures encoded via the use of the Zero Width Joiner (ZWJ) should include the ZWJ as one of the components of the glyph (U+200D), for example, a ligature of Cyrillic Small Letter A and Cyrillic Small Letter U joined by the ZWJ should be named as *uni0430200D0443* or as *uni0430_uni200D_uni0443*. Note that "some current software limitations subject all glyph names to a limit of 31 characters in length" (Adobe Systems, 2007). This may result in problems when dealing with complex ligatures in skoropis fonts.

Some font developers place precomposed glyphs or ligatures into the Private Use Area (PUA), from which they may be accessed either via the PUA codepoint or via an appropriate OpenType feature. However, if these glyphs have names in accordance with their location in the PUA (e.g., *uniE000*), they will not behave as desired during search, copy, and other operations. Thus, it is recommended that precomposed glyphs be given the names that consist of their components, as described above, even if they are encoded in the PUA and not outside of the Unicode standard. See Andreev et al. (2015a) for examples and more information.

4.4 Use of Advanced Typographic Features

The above subsections described the repertoire and naming of glyphs used in Church Slavonic; this subsection describes the use of advanced typographic features required for proper Church Slavonic fontography. Briefly, fonts with advanced typographic features (so-called "smart fonts") contain not only glyph outlines but also additional instructions on how glyphs are combined and positioned. These instructions are read and processed by the rendering system, which is responsible for correctly positioning or substituting glyphs to generate the desired text stream. In order for Church Slavonic text to appear correctly, the font designer needs to specify the writing system rules and the user needs to be running a rendering system that correctly process these rules.

Presently, several technologies providing "smart" features exist – these include OpenType, Apple Advanced Typography (AAT), and SIL Graphite. These three technologies do not necessarily compete – a font may be designed to work with more than one of these technologies, thus providing the user with some flexibility. In addition, at the time of this writing, not all software applications support all of these technologies or all features of a given technology. Font designers should keep in mind the target user audience and decide appropriately which technology should be used. Because OpenType is by far the most prevalent "smart font" technology, we describe the OpenType features below.

4.5 OpenType Technology

What is OpenType?

OpenType Layout is a technology for advanced typography developed by Microsoft Corporation and Adobe Systems and based on the TrueType font format. The OpenType Layout technology was joined to the TrueType or Type 1 font formats, resulting in what is now called an OpenType font. Thus, an OpenType font may contain either TrueType or PostScript outlines. Glyphs in an OpenType font are mapped to their Unicode positions, thus rendering the format trivially Unicode compliant. In addition, however,

Table 12: OpenType features used to implement Church Slavonic typography.

OpenType feature	Description	Table
Language-based forms:		
ccmp	Glyph composition/decomposition	GSUB
Glyph positioning:		
kern	Pairwise kerning	GPOS
mark	Mark to base positioning	GPOS
mkmk	Mark to mark positioning	GPOS

OpenType fonts may include many non-standard glyphs, for example, old-style figures, small capitals, contextual or stylistic alternatives, and ligatures not mapped to Unicode codepoints. The support of these features is contingent on the availability of software that correctly implements features that allow access to these glyphs. Software that cannot access the "smart" features of the font can still access the characters mapped to the Unicode codepoints, but would not be able to correctly implement such features as glyph positioning or contextual substitution. Software that is not compliant with Unicode can only access the first 256 glyphs of the font.

Glyph Substitution

OpenType provides a variety of "features" for advanced typography, though no feature is required to be implemented in a font. Some features are always on while other features may be turned off and on by the user to achieve specific results. The OpenType specification distinguishes between "language-based forms" – that is, forms of glyphs specified by the rules of the writing system and required for correct orthography – and "typographical forms," which are conventions used in typography. In simplest terms, an example of a language-based form would be the correct selection of the initial, medial, or final form of an Arabic letter. An example of a typographical form would be the ligature fi (composition of f and i); this ligature is not a feature of the English language or Latin script, but does commonly appear as a typographical convention. Generally, the language-based forms are implemented using features that are always on while the typographical forms rely on features that may be turned off and on by the user.

Standard elements of Church Slavonic typography are language-based forms, and should be implemented using features that are always on, as listed in Table 12. Other OpenType features are intended for more complex typographical forms and may not be supported in basic software (rather, they may be turned off by default with no simple way to turn them on). The order in which the features are invoked is the same order in which they are listed in Table 12.

Glyph substitution in OpenType fonts is specified in the GSUB (Glyph Substitution) table. In simplest terms, the GSUB table defines a mapping from a glyph index (or indexes) in the CMAP table of the font to the glyph index (or indexes) of substitute glyphs allowing for substitute glyphs to be selected. There are several types of possible substitution algorithms. A *single substitution* replaces a single glyph with another single glyph. This is the type of substitution that, for example, should be used to select the alternative form of the Psili (U+0486), which in Synodal Church Slavonic is used over capital letters:

◌҆ → ◌҆

A *multiple substitution* replaces a single glyph with more than one glyph. This substitution can be used to decompose ligatures for various purposes. An *alternate substitution* provides different looking, but equivalent versions of a glyph (so-called aesthetic alternatives). This type of substitution is not required for proper Church Slavonic typography, but may used to provide access to stylistic alternatives (see below). A *ligature substitution* replaces several glyphs with one glyph. This substitution should be used to create both the digraphs that have standard presentation forms and the ligatures that are implemented via the Zero Width Joiner (see Section 3, above), as the following examples demonstrate:

◌҆ + ◌̀ → ◌҆̀ (ligature substitution using *ccmp* feature)
т + ZWJ + к → т‍к (ligature substitution using *ccmp* feature)

A *contextual substitution* substitutes glyphs given a specific context – that is, a substitution within a certain pattern of glyphs. To specify such a substitution, the developer describes one or more input glyph sequences and one or more substitutions to be performed on that sequence. This is the type of substitution that should be used to define context-specific glyph variants, such as the variant form of Uk used with a combining (superscript) letter:

ꙋ + ◌ⷯ → ꙋⷯ (contextual substitution using *ccmp* feature)

A *chaining contextual substitution* is a more powerful functionality for contextual substitution that allows the developer to specify backtrack, input and lookahead sequences. Finally, a *reverse chaining contextual substitution*, provides the same glyph substitution functionality but allows for processing of the input glyph sequence from end to start (Microsoft Corporation, 2002).

In defining the entries of a GSUB table, in addition to specifying the substitution types, the developer specifies zero, one, or more features. The two features of interest for Church Slavonic typography are *ccmp* and *liga*.

The *ccmp* feature is used to compose a number of glyphs into one glyph or decompose a glyph into a number of glyphs. This feature should be used for all language-based glyph substitutions. It is invoked before any other feature because it is designed to provide control over the shaping of glyphs following the rules of the writing system. For purposes of Church Slavonic typography, the *ccmp* feature should be used for almost all substitutions.

The *liga* feature should be used for glyph substitutions involving optional ligated forms (typographic alternatives). While the *ccmp* feature is always on, the *liga* feature, though on by default, may be turned off by the user to suppress typographic alternatives. Ligatures with more components must be stored ahead of those with fewer components in order for the GSUB rules to be invoked correctly (see more on this below). This feature is commonly used in Latin fonts to compose and decompose the fi ligature and similar typographic forms. In Church Slavonic typography, it may be used for certain ornamental ligatures (especially in decorative fonts); we still suggest using the *ccmp* for implementing ligatures joined with the Zero Width Joiner, since by explicitly entering the ZWJ, the user has already requested that two characters form a ligature.

Glyph Positioning

After carrying out glyph substitution operations, the rendering software proceeds with glyph positioning operations. Glyph positioning in OpenType is handled via the use of the GPOS (Glyph Positioning) table. The three features of GPOS relevant for proper Church Slavonic typography are *kern*, *mark* and *mkmk*. The *curs* feature may also be used to implement correct attachment of glyphs in skoropis fonts, but, for the sake of brevity, we omit discussion of issues associated with cursive attachment.

The *kern* (Pairwise Kerning) feature is used to adjust the amount of space between glyphs. Its most common application is to adjust the horizontal spacing (kerning) between glyphs so as to provide a more visually appealing result. The feature can also be used for vertical kerning, which may have some applications in the design of decorative fonts. In the below example, the *kern* feature is used to bring the glyph Cyrillic Letter Lowercase A closer to the glyph Cyrillic Letter Capital Te:

Т + а → Та (no kerning)
Т + а → Та (horizontal kerning via *kern* feature)

The *kern* feature may also be used in a contextual setting by defining a contextual kerning table. One application of contextual kerning is to provide for more or less space before or after a letter that has an attached diacritical mark.

Kerning rules (adjustments) may be stored as one or more tables defined either for pairs of left and right glyph classes or for individual glyph pairs. If both forms are used, the classes should be listed last, so as to provide a means to replace any non-ideal values that may result from the class tables. Additional adjustments may be provided for larger patterns of glyphs (that is, for three, four, five, or more glyphs) to overwrite the results of pair kerns in particular combinations. According to the specification, these should be listed before the kerning adjustments for individual glyph pairs (Microsoft Corporation, 2002).

Note that while kerning is a standard feature of modern typography and is expected to appear in modern (Synodal) Church Slavonic texts, the use of kerning in poluustav or ustav era fonts would be completely anachronistic, and, thus, should be avoided. The only exceptions are in the rendering of certain digraphs where it is required to bring the constituent glyphs closer together.

The *mark* (Mark-to-base Positioning) feature positions diacritical mark glyphs with respect to a base glyph or a base ligature glyph. This is accomplished by defining "anchor points" – coordinates at which diacritical marks attach to base glyphs – for the base glyph either for individual diacritical marks or for classes of diacritical marks. Note that it is recommended that base glyphs have anchor points for all diacritical marks, even for combinations of base glyphs and marks that do not typically exist in Church Slavonic. Failure to provide all of the necessary anchor points may result in errors in the font and improper behavior.

Finally, the *mkmk* (Mark-to-Mark Positioning) feature is used to position diacritical marks with respect to other diacritical marks. This feature should be used to correctly implement the Pokrytie over combining (superscript) letters (although in some instances, combining letters with a Pokrytie may need to be provided as precomposed glyphs, which is accomplished via the *ccmp* feature). As well, as discussed in Section 3, all combinations of diacritical marks that are not expressly defined as functional in the Church Slavonic writing system by default are expected to stack vertically away from the base character. This stacking behavior likewise is properly achieved using the *mkmk* feature.

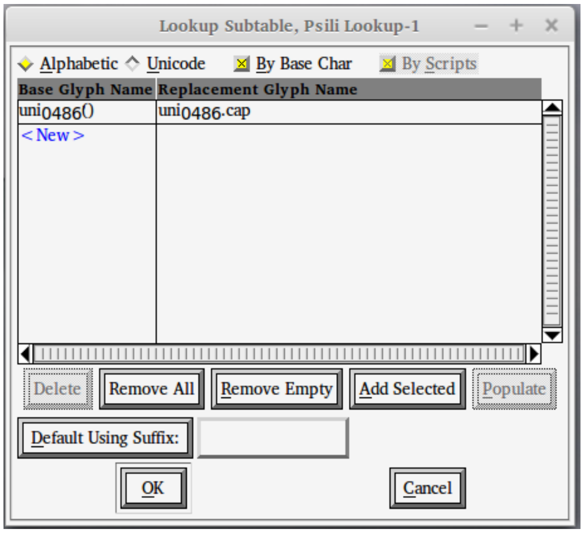
Ordering of Lookups and Specification of Script

The correct ordering of lookups in the GSUB and GPOS tables is essential for the proper behavior of the font and the rendering system. As we have noted above, it is required that in the GSUB table, the *ccmp* lookup tables be placed first, followed by the *liga* lookup tables. In the GPOS table, the correct order is *kern* followed by *mark* and then *mkmk*.

In addition, it is vital that the lookup tables implementing any given feature also be ordered correctly. For example, suppose we have two substitutions defined in a font, one of which uses the *ccmp* feature to compose a glyph for the Iso from the character sequence Psili (U+0486) and Acute Accent (U+0301) and another, which also uses the *ccmp* feature to invoke substitutions of alternate forms of diacritical marks over capital letters. In this instance, it is essential that the lookup table that composes the Iso be ordered first and the lookup table that replaces the Iso with an alternate form for use with uppercase letters be placed second. Failure to logically order the lookup tables will result in rendering problems.

Finally, anecdotal evidence suggests that software is sensitive to which scripts (writing systems) are specified for a particular lookup. It is highly recommended that all lookup tables are specified at least as applying to the default (*DFLT*), Latin (*latn*) and Cyrillic (*cyrl*) scripts. Failure to specify default and

Figure 11: Single substitution in FontForge.



Lookup Subtable, Psili Lookup-1

Base Glyph Name	Replacement Glyph Name
uni0486◌	uni0486.cap
<New>	

Latin in addition to Cyrillic may result in the rendering system's ignoring of the lookup. In principle, OpenType rules are script- and language-specific, but in practice, not all software correctly identifies the language of the text stream and passes this information to the rendering system. In addition, some software manufacturers (namely, as of the time of this writing, Microsoft Corporation) do not recognize Church Slavonic as a language and do not allow the user to specify that a given text is in Church Slavonic. Given this limitation, we recommend that font designers refrain from defining language-based rules and use script-based rules or global rules only.

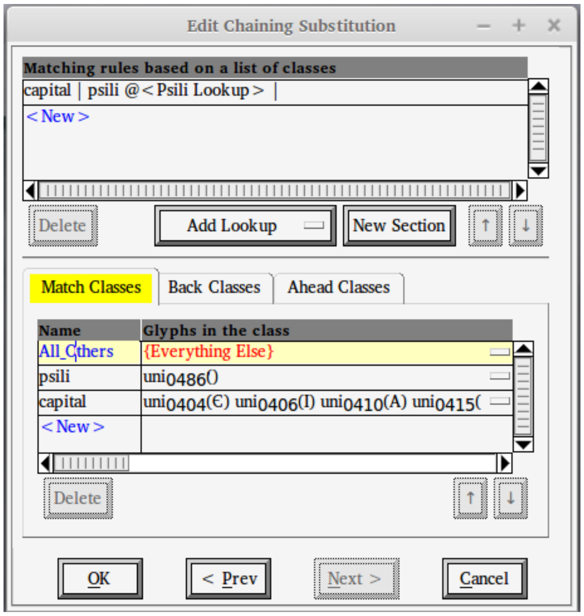
Worked Examples

It is perhaps easiest to demonstrate the features of OpenType and their proper implementation by means of worked examples. Thus, we provide two simple examples in this subsection. For our examples, we reference the free font editor FontForge, which in our opinion is the most suitable tool for design of OpenType fonts.

In the first example, we shall create a contextual lookup that substitutes the glyph for the Psili (U+0486) to an alternate form when the Psili is placed over an uppercase letter. To begin, we have two separate glyphs in our font. The first of these, the lowercase form of the Psili, is encoded at U+0486 and appropriately named *uni0486*. The second, which is the form that occurs over capital letters, is encoded outside of the Unicode range and, keeping in mind the discussion above on glyph naming, named *uni0486.cap*.

We now create in FontForge a new substitution as follows. From the `Element` menu we select `Font Info`. In the `Font Information` window, we click on `Lookups` and then, making sure that the GSUB tab is active, click `Add Lookup`. Under `Lookup Type`, we select `Single Substitution`. We leave the `Feature` list blank and under `Lookup Name` provide a descriptive name for our lookup (*Psili Lookup*); we then click `OK` to create the new table. Selecting our new table, we click `Add Subtable` and provide a name for the subtable, and click `OK`. We are now ready to define our new substitution. In the `Lookup Subtable` window, under `Base Glyph Name` we select the Psili glyph, called *uni0486*. Under `Replacement Glyph Name`, we select the alternative form, called *uni0486.cap*. This is displayed in Figure 11. Clicking `OK` on this dialog, we complete the process of defining the new substitution. Note that because our substitution has no feature enabled, it will not get invoked by the rendering system. We need another substitution that will invoke this substitution given a specific context.

Figure 12: Contextual Chaining substitution in FontForge.



Matching rules based on a list of classes

capital | psili @<Psili Lookup> |

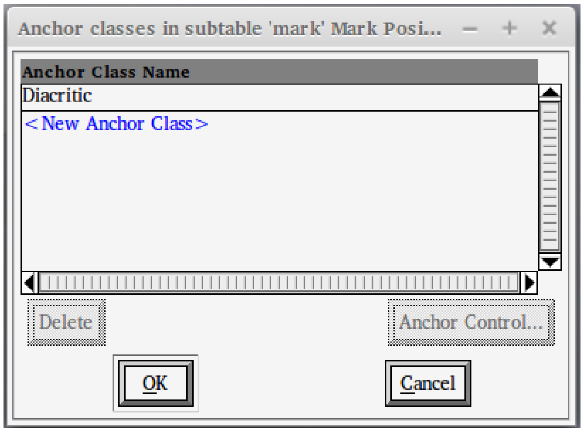
Name	Glyphs in the class
All_Others	{Everything Else}
psili	uni0486()
capital	uni0404(Є) uni0406(І) uni0410(А) uni0415(
<New>	

To this end, we click on `Add Lookup` again and this time in the `Lookup Window` under `Type` select `Contextual Chaining Substitution`. Under `Feature`, we type `ccmp` and under `Scripts`, we ensure that `DFLT` (default), `latn` (Latin), and `cyrl` (Cyrillic) are listed. After giving our new Lookup a name, we click `OK`. Selecting this new Lookup, we again click `Add Subtable` and provide a name for the Subtable. Clicking `OK` opens the `Edit Chaining Substitution` window. The simplest approach, in the opinion of the authors, is to select `By Classes` and `Simple dialog type` (the default choices) and then to click `Next`. In the new window that appears, under `Match Classes`, we create a new class with name *psili* and one glyph, *uni0486*. We create a second class with name *capital* and containing all glyphs for which we wish the substitution to be invoked (for example, *uni0410* (Cyrillic Capital Letter A), *uni0415* (Cyrillic Capital Letter Ie), and the other Cyrillic capitals). Under `Matching rules based on a list of classes`, we now enter the following rule: `capital | psili @<Psili Lookup>`. This is displayed in Figure 12. Clicking `OK` completes the creation of this substitution. Note that while FontForge allows us, in principle, to list multiple different substitutions in one Subtable and to include multiple Subtables in each lookup table, in practice, because of limitations in some software implementations, it seems to be best to define a separate lookup table for each substitution.

Next we provide an example of a GPOS (glyph positioning) substitution. Here, we will create a rule that positions the diacritical marks over base characters using the *mark* (Mark positioning) feature. In FontForge, in the `Lookups` section of the `Font Information` window we now select the `GPOS` tab. We click `Add Lookup`. Under `Type`, select `Mark to Base Position`. Under `Feature`, specify `Mark positioning`. Again, we ensure that under `Script(s) & Language(s)`, Default (`DFLT`), Latin (`latn`) and Cyrillic (`cyrl`) are listed. Under `Lookup Name`, we give the Lookup a relevant name, such as `Mark Positioning 1`, and click `OK`. Next we click `Add Subtable`, specify the name of the subtable and click `OK`. The `Anchor classes` window appears. In the `Anchor classes` window, we create a new class called `Diacritic`. This is demonstrated in Figure 13. Clicking `OK` concludes the operation of creating the new Lookup. We further click `OK` to close the `Font Information` window.

We next select and open the appropriate diacritical mark glyph. In order for the glyph to correctly work with OpenType Mark to Base positioning rules, it is vital to ensure that it has the correct glyph class.

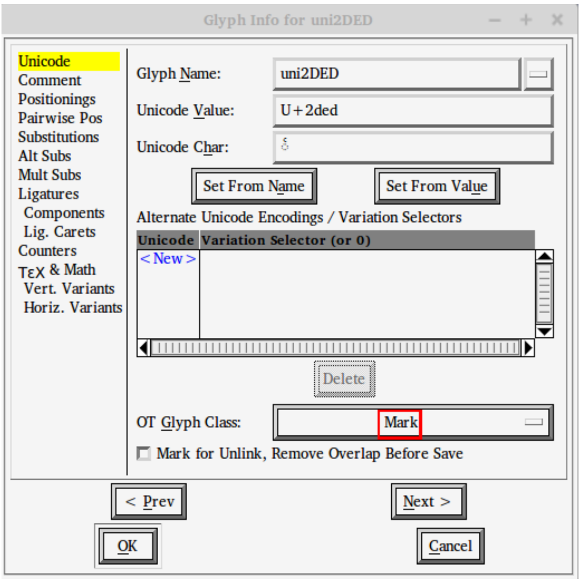
Figure 13: Anchor classes for Mark to Base Positioning in FontForge.



Anchor classes in subtable 'mark' Mark Posi...

Anchor Class Name
Diacritic
<New Anchor Class>

Figure 14: Specifying the correct glyph class in FontForge.



Glyph Info for uni2DED

Glyph Name: uni2DED

Unicode Value: U+2ded

Unicode Char:

OT Glyph Class: Mark

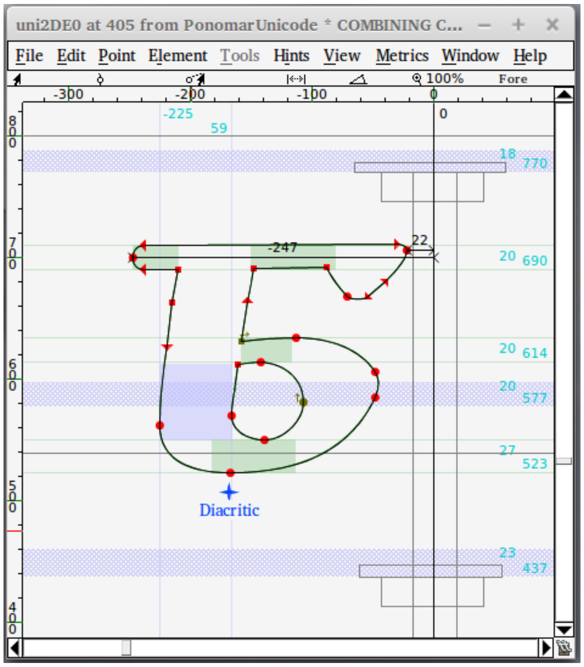
☐ Mark for Unlink, Remove Overlap Before Save

For this, pull down the Element menu and select Glyph Info. In the resulting Glyph Info dialog, set OT Glyph Class to Mark and click OK. This is displayed in Figure 14.

Next, we specify the anchor points for the positioning of this glyph. To do this, pull down the Point menu and click Add Anchor. In the Anchor Point Info window, select our Diacritic class and ensure that the Mark radio button is selected (and not the Base Glyph radio button). You can specify the coordinates of the anchor point by setting values for X and Y, or you can click OK and then use the mouse to drag the anchor point to the desired location in the glyph window. We demonstrate the latter approach in Figure 15. Next, we must add this anchor point for other relevant diacritical marks; to do this, we repeat this operation: for every diacritical mark glyph, we add the Diacritic anchor point from the Point → Add Anchor menu item, ensuring first that the OT Glyph Class is set to Mark.

Finally, we must add the relevant anchor points to the base characters. This follows exactly the same procedure. For every relevant base glyph, we add the Diacritic anchor point from the Point → Add Anchor menu item. However, here we must ensure that in the Anchor Point Info window,

Figure 15: Anchor point positioning in FontForge.



the `Base Glyph` radio button is set. As well, the `OT Glyph Class` for the base glyphs should be set to `Base Glyph` or to `Automatic`.

For additional examples and explanations, developers are encouraged to consult the FontForge documentation.[21] Even if developers are using a software package other than FontForge, the FontForge documentation can still be helpful in understanding the OpenType specification. Finally, font developers should keep in mind that not all software supports contextual substitutions. Moreover, some software may support these substitutions only for certain scripts (usually Arabic or Indic scripts) and not for others. Often, contextual substitutions may not be supported for Cyrillic. In some software, the user may need to explicitly enable some feature (usually called "display ligatures") to turn on processing of OpenType rules. Such limitations should be viewed as bugs in the software, and developers should alert the maintainers of the software package at fault. As well, it is best to test the font in a software that is known to support the basic OpenType features without limitations; we recommend XƎTEX and the Mozilla Firefox web browser.

4.6 SIL Graphite technology

Graphite is a powerful "smart font" technology developed by SIL International. Graphite and OpenType are not competing technologies; rather, font developers may choose to support both technologies simultaneously. 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

[21] In particular, as of the time of this writing, the following Tutorial was extremely helpful: `http://fontforge.org/editexample6-5.html`.

supported widely: in addition to SIL's own WorldPad editor, Graphite is supported in LibreOffice (starting with version 3.4), Mozilla Firefox (starting with version 11), and XƎTEX (starting with version 0.997).

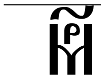
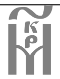
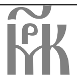
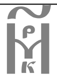
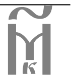
Graphite features are written in a C-like programming language called Graphite Description Language (GDL), a rule-based programming language that is used to describe the behavior of a writing system. Then, the TrueType font file is compiled against the GDL file by using a Graphite compiler, a free software package available from SIL International. The resulting TrueType font file contains additional tables that are used by the Graphite engine of a Graphite-aware application. Applications that are not Graphite-aware simply ignore these additional tables. Note that, as of the time of this writing, Graphite tables may only be added to TrueType-formatted fonts; it is not possible to add Graphite tables to OpenType-CFF fonts.

While all of the features of Church Slavonic writing can be successfully implemented by using only OpenType features, a developer may wish to also add Graphite features to his font for various reasons. Furthermore, since Graphite features are written in a separate GDL file and then compiled into a font, they are not font-specific; thus, a developer may quickly develop many fonts with the same Graphite features and using only one GDL file. Second, developing a font with both OpenType and Graphite features will allow the font to be usable in a greater number of settings. We recommend that font developers implement both OpenType and Graphite features. Because Graphite features are not predefined, font documentation should, however, clearly describe the Graphite features that are available in a given font.

SIL International has developed a set of Perl scripts available in the `Font::TTF::Scripts` module on CPAN. We have not tested these scripts under Windows, but under the GNU/Linux operating system, they automate much of the initial "busy work" involved with creating a GDL file. To add Graphite features, it is simplest to start with a FontForge SFD (Spline Font Database) file containing already the OpenType features (notably, the attachment points for the *mark* and *mkmk* features) and a generated binary font in TrueType format. The utility `sfd2ap` will convert the SFD file to an XML-based anchor point database, which contains a listing of the glyphs in the font and their attachment points. Next, the utility `make_gdl` can be used to create a skeleton GDL file from the XML attachment point database and the binary TrueType font. The developer then needs to edit the GDL file by hand, adding the necessary code for the Graphite rules. For the syntax in constructing the Graphite tables we refer the reader to the GDL Documentation (Hosken et al., 2011). Finally, the GDL file is compiled against the TrueType font by invoking the `grcompiler` program. For testing purposes, we recommend testing the resulting font in SIL's own WorldPad editor first, as this should avoid the possibility of bugs with the rendering system. Developers are encouraged to consult the GDL Documentation and online GDL Tutorials for helpful information.

4.7 Stylistic Alternatives

In addition to contextual glyph forms, which are automatically selected by the rendering system on the basis of the rules of the writing system, font designers may wish to provide other alternative glyphs that may be selected by the user at will. Typically, these are glyphs that differ from the base glyph only in graphical appearance and the use of these glyphs does not follow any language-based or typography-based rules, but rather is just an embellishment. For example, the 1641 edition of the Typicon (*Oko Tserkovnoye*) contains different glyphs for the Symbol for Mark's Chapter (encoded at U+1F545) and a font designer may wish to provide all of these glyphs in a font:

U+1F545	Alternative Glyphs			
				

Access to such alternative glyphs may be provided in OpenType via the use of the *salt* (Stylistic Alternatives) feature. This feature replaces the default glyph of a character with the stylistic alternative. To implement this feature, the same procedure is followed as above to create a *salt* lookup table, which maps the glyph id of the default form to one or more glyph ids for its stylistic alternatives. The lookups may be either one-to-one (GSUB lookup type 1) or may suggest an entire list of potential replacement glyphs (GSUB lookup type 3). Note that in the latter case, when the replacement is a set of glyphs, the feature requires active user interaction, and so the application must provide a means for the user to select the desired alternative. In practice, this is not well supported in most software implementations. In addition, the *salt* feature is inactive by default and must be explicitly enabled by the user.

In addition to, or instead of, providing Stylistic Alternatives of individual glyphs (via the *salt* feature), font developers may provide alternative glyphs for entire ranges of the character set. For example, this could be an alternative set of uppercase Cyrillic letters for decorative titling or a set of lowercase Cyrillic letters where all diacritical marks position in a particular manner. Generally speaking, all of the glyphs in the set are designed to have a similar visual appearance or to otherwise interact with each other. As an extreme example, one could provide sets of glyphs representing different recensions of Church Slavonic within the same font.[22] Access to an entire set of alternative glyphs in OpenType is provided by the "stylistic sets" features, a set of twenty features named *ss01* through *ss20*. Font developers implement these sets in a similar manner, providing a mapping from the set of base glyphs to the set of replacement glyphs, via the *ssXX* table. Each stylistic set lookup table uses one-to-one (GSUB lookup type 1) substitutions. Note that after the user has selected a stylistic set, he may still wish to apply glyph-specific features, (for example, *salt*), to individual glyphs. It is thus essential to correctly order lookup tables in the font to provide the desired results. We refer font developers to the OpenType documentation for more information (Microsoft Corporation, 2010).

The functionality of *salt* and stylistic sets may also be provided in SIL Graphite. Since Graphite does not provide predefined features, the developer simply creates the relevant substitution rules and provides a feature for turning these rules on. Graphite features may have discrete values that may be used to select the desired alternative. Thus, for example, we can create a Graphite feature called `mark` with possible values of `1`, `2`, `3`, or `4` for selecting the alternative glyphs for the Mark's Chapter Symbol depicted above. The default value of `0` means that the base form is selected.

We note that while *salt*, stylistic sets, and the corresponding SIL Graphite approach, are useful ways to provide alternative glyphs, they have their limitations. As of this writing, not all software allows access to Stylistic Alternatives or Stylistic Sets; for example, there is no standardized methodology to turn on such advanced typographic features in HTML for use in webpages. Second, and more importantly, while the Unicode standard is a universal standard for encoding characters, there is no universal standard for font features. Font developers may choose to implement OpenType features, SIL Graphite features, both, or neither in their fonts. There is no requirement that developers implement the same features and there is no naming conventions for these features (as we have pointed out, the name of the SIL Graphite feature is entirely up to the developer). There is no mechanism for requiring developers to map a specific alternative form to a specific feature. While it is recommended that alternatives present in more than one font face be ordered consistently across a family of fonts, there is no mechanism to require font developers to do so. Relying on these features forces the desired representation of a text stream to be dependent on the choice of font. Thus, such features should only be used for handling graphical embellishments and other situations that have no semantic meaning; they are not an alternative to encoding and their use should be avoided in any documents intended for data interchange. In addition, to provide access to such alternatives in implementations that do not support advanced typographic features, font developers may choose to encode alternative glyphs in the Private Use Area of Unicode, as discussed by Andreev et al. (2015a).

[22] We do not recommend using this approach, as it can quickly lead to complex fonts that are difficult to use effectively.

5 Church Slavonic Collation

Collation is the process and function of determining the sorting order of strings of characters. Though at first glance it may appear that the topic of determining the correct sort order of Church Slavonic strings is only of academic interest, in fact, defining the collation of a writing system is the second step required for correct implementation, first only to defining the repertoire of characters. This is because a collation is required for the work of text processing systems, not only for purposes of sorting textual data but also for string comparison and matching operations. While Church Slavonic uses the Cyrillic script, it is important to note that collation varies according to language and culture. For example, Germans, French and Swedes sort the same characters differently, although all three use the Latin alphabet. In the same way, the collation rules for Church Slavonic texts will be different from the collation rules for Russian or any other language written with Cyrillic characters.

Collation is not the same thing as Unicode code point order. The fact that characters in the Cyrillic blocks of Unicode are encoded "not in the correct order" from the standpoint of Church Slavonic (or, for that matter, any writing system using the Cyrillic script) has no bearing on collation. Instead of using the binary comparison derived from code point order, the collation of Unicode strings relies on multilevel comparison codes.

The Unicode Collation Algorithm (UCA) details how Unicode strings are compared and sorted. The UCA provides a *collation table*, which associates four numerical weights with each character, thus allowing for a four-level *comparison code*. Default comparison codes are provided via the Default Unicode Collation Element Table (DUCET). This table determines the default sort order for Unicode strings. However, its codes may be rearranged to provide for a different sort order based on language and locale. The process of reordering certain entries of the DUCET in order to provide a language- and locale-specific sort order is called *tailoring*. Such tailorings may be standardized and stored in the Common Locale Data Repository (CLDR) of Unicode, an XML-driven database that provides standardized locale data to computer applications. Below, we detail a possible tailoring for Church Slavonic.

5.1 Collation of Synodal Church Slavonic

We start with the basic characters – the letters of the Cyrillic writing system that are used in modern Church Slavonic, which are commonly given in grammar books in the following order: Аа, Бб, Вв, Гг, Дд, Ее, Єє, Жж, Ѕѕ, Зз (Ꙁꙁ), Ии, Ї, Кк, Лл, Мм, Нн, Оо, Ѻѻ, Ѡѡ (Ѽѽ), Пп, Рр, Сс, Тт, Ѹѹ, Ꙋꙋ, Фф, Хх, Ѿѿ, Цц, Чч, Шш, Щщ, Ъъ, Ыы, Ьь, Ѣѣ, Юю, (Ѫѫ), Ꙗꙗ, Ѧѧ, Ѯѯ, Ѱѱ, Ѳѳ, Ѵѵ.[23] Note that the letter ѫ is not used in Synodal Church Slavonic except as a symbol in tables used for Paschal computus, where it is placed after ю but before ѧ.

Though, at first glance, the problem of sort order appears simple enough once we agree on the alphabetical order of characters, the issue is, in fact, more complex. We must also take into account the usage of combining letters and diacritical marks as well as the fact that some characters are variants of each other.

The diacritical marks used in Synodal Church Slavonic are listed in Table 13. Note that the Breve is used only over the letter и to indicate a short [i] sound. Furthermore, for the sake of compatibility with existing Unicode specifications, the diæresis (*kendema*) is encoded in two ways: when it occurs over the letter і, it is encoded as U+0308, and when it occurs over the letter ѵ, as U+030F; nonetheless, these are

[23] See, for example, Koz′min (1903, p. 3), Archb. Alypy (Gamanovich), (p. 17), and Vorob′yeva (2008, p. 32) for this standard sequence of letters. However, Pletnyeva et al. (1996, p. 28) place ѡ and ѽ between ꙗ and ѧ, while Mironova (2008, p. 10) keeps ѡ with ѻ but places ѿ at the end of the alphabet, treating it is a ligature. In any case, these differences are not significant for our purposes, as will be seen from the discussion below. Note also that this is the alphabetical ordering in modern Church Slavonic; grammar books focusing on a treatment of Old (Church) Slavonic will propose a different order (see below).

Table 13: Diacritical Marks used in Synodal Church Slavonic

Mark Name	Traditional Name	Codepoint or Sequence	Example
Simple marks:			
Acute Accent	oxiya	U+0301	А́ а́
Grave Accent	variya	U+0300	А̀ а̀
Inverted Breve	kamora	U+0311	А̑ а̑
Breve	kratkaya	U+0306	Й й
Soft Breathing	psili	U+0486	А҆ а҆
Diæresis	kendema	U+0308 *or* U+030F	Ї ї Ѷ ѷ
Titlo	titlo	U+0483	А҃ а҃
Vertical Tilde	yerik *or* yerok	U+033E *and* U+2E2F	Д̾ д̾
Payerok	payerok	U+A67D *and* U+A67F	дꙿ, д꙽
Kavyka	kavyka	U+A67C *and* U+A67E	а꙼, ꙾а꙾
Pokrytie	pokrytie	U+0487	А҇ а҇
Complex marks:			
Breathing with Acute	iso	U+0486 U+0301	А҆́ а҆́
Breathing with Grave	apostrof	U+0486 U+0300	А҆̀ а҆̀
Breathing with Inverted Breve	veliky apostrof	U+0486 U+0311	Ѡ҆̑ ѡ҆̑

functionally equivalent. The *kavyka* provides no meaningful grammatical information: its purpose is to indicate a footnote containing a gloss (alternative reading of a word or words); thus, it should be ignored for collation. Note also that for the *kavyka*, the *yerok* and the *payerok*, both a combining and a standalone version have been encoded in Unicode.

In addition to the combining diacritical marks, Church Slavonic uses a variety of combining (superscript) letters. As we discussed in Section 2.5, in earlier recensions, any of the letters of the alphabet may have occurred as a superscript letter but in Synodal Church Slavonic, the repertoire of superscript letters is in fact limited. We have listed all of the occurrences in Appendix B. The *pokrytie* occurs over certain of these combining letters: some superscript letters are used only without a *pokrytie* and some are used only with a *pokrytie.* As the *pokrytie* by itself carries no meaning but rather serves as a stylistic embellishment, it, too, should be ignored for collation. Note also that the vertical tilde (*yerok*) indicates an omitted hard sign and thus should sort in the same way as a combining (superscript) hard sign. In some Kievan editions, the *payerok* may be used to indicate an omitted soft sign (or, rarely, also an omitted hard sign).

While in earlier recensions of Church Slavonic, the usage of character variants was largely haphazard, Synodal Church Slavonic has some rigid orthographic rules, and these rules must be taken into account. Some of the variants are merely different graphical representations, like the ꙁ used in Kievan editions instead of з; the usage of ꙁ vs. з provides no semantic information. Others are functional variants; for example, є is used whenever the [je] sound occurs in initial position and is also used in medial or final position in place of е to distinguish certain grammatical forms (see below). Similarly, о is usually used in medial position while ѻ is used in initial position, but may also occur in medial position as the first letter of a compound word. For purposes of collation and string comparison, it may be necessary to ignore the variant spelling in some instances, while taking it into account in others. The UCA provides mechanisms to achieve this flexibility.

To construct the Church Slavonic sort order, we first analyzed the order of words listed in Church Slavonic dictionaries. Unfortunately, the literature in this field is quite limited. Many of the existing dictionaries are not true dictionaries of Church Slavonic, but provide entries for Church Slavonic and

Old Slavonic words together, as well as archaic Russian words; in many instances, words are provided in modern Russian characters and according to the established Russian sort order. This is true both of the only dictionary explicitly devoted to liturgical Church Slavonic of the Synodal recension[24] and of the comprehensive *Dictionary of Church Slavonic and Russian* published by the Russian Imperial Academy of Sciences (1847). Of the dictionaries that do provide words in Church Slavonic characters, we consulted the dictionary of D′yachenko (1899) (although this dictionary is now standard reference, it also contains Old Slavonic and archaic Russian words, and is not without its limitations), the more limited lexicons of Gilterbrandt (1898), and the modern dictionary of Russian-Slavonic paronyms constructed by Sedakova (2005). Given the sort order in these dictionaries, we are able to formulate the following rules for the collation of Synodal Church Slavonic:

1. Collation should follow (phonetically), as closely as possible, the sort order of pre-reform Russian. Thus ѻ, о and ѡ should sort together, as all are pronounced as the Russian o. In this way, we have ѻ҆́ба before ѡ҆бавла́ємь before ѻ҆́дръ. Similarly, ꙗ҆зы́къ should sort next to ѧ҆зы́къ, since both cases are pronounced as *yazyk*. Since ѣ is treated as a standalone character in pre-reform Russian also (and not as a variant of є), it sorts in its proper place after ь.
2. The letters й, ѽ and ѷ should be treated as decomposable into their base glyph and diacritical mark(s). However, the letter ѿ is treated as a ligature for ѡт, not as a letter ѡ with combining (superscript) т, since this ensures that all words written with ѡт, ѿ and от sort together.
3. In keeping with the phonetic order, the digraph оу sorts as у, not as о followed by у.

Looking at the sort order implemented in dictionaries, however, is not sufficient, since for collation we must consider also the oblique cases of nouns and other wordforms that are typically not listed in a dictionary. We consider also the morphology of Church Slavonic. In particular, the letters е and ѡ are also used as variants of є and о, respectively, to indicate noun forms in the dual and plural number when the wordform is the same as the form in the singular. For example, consider рабо́мъ (instrumental singular) vs. рабѡ́мъ (dative plural) and фарїсє́й (nominative singular) vs. фарїсе́й (genetive plural). These orthographical changes may occur not only in endings but also in roots of words, as in the example of ѻ҆́трока (genetive singular) and ѻ҆́трѡка (nominative dual); see Archb. Alypy (Gamanovich), (pp. 44ff), for a discussion. The same changes may also occur for some pronouns (*op. cit.*, p. 61). In addition to using variants to indicate wordforms, accent changes are also used for this purpose; thus the inverted breve (◌̑) occurs in wordforms in the dual or plural number when the form is the same as a wordform in the singular number (for example, ра́бъ [nominative singular] and ра̑бъ [genetive plural]; рабꙋ̀ [dative singular] and рабꙋ̑ [genetive dual]). Thus we formulate the following principle:

4. The collation tailoring should allow the user to distinguish between wordforms. As far as it is possible, the singular form should sort before the dual and plural forms and the nominative case should sort before the oblique cases. Thus, we wish to have рабо́мъ before рабѡ́мъ and also ра́бъ before ра̑бъ.[25]

Generally speaking, the diacritical marks in Synodal Church Slavonic are used to indicate stress, and it is natural for these marks to sort from left to right. Thus, ве́щьми should sort before вещьмѝ. Casing is rarely used in Church Slavonic, although uppercase forms may be used in Kievan editions for *nomina sacra* and proper names. Accordingly, we formulate two additional principles:

[24] This dictionary was constructed between 1847 and 1872 by Archpriest A. Nevostruyev but remains, as of yet, unpublished in full. For more information see Davydenkova et al. (2013).

[25] Note that it is not always possible to implement such a sort order without resorting to special exceptions. For example, the letter ѧ is used after sibilants in place of а to make case distinctions (*cf.* на́ша [feminine nominative singular] and на́шѧ [masculine accusative plural]). However, it would not be correct to sort а together with ѧ in all instances, and the sort order we propose would provide на́ша before нашє́ствїє before на́шѧ.

5. Sorting of diacritical marks should be from left to right; to ensure correct sorting of abbreviations, the superscript letters should sort after other diacritical marks, but in the same order as the base letters, for example: дв҃а before два̀ before дваⷤ҇.
6. Uppercase letters should sort before lowercase letters (although in certain contexts the opposite may be desirable).

As we have already mentioned, the Unicode Collation Algorithm (UCA) provides for four levels of comparison codes. The *primary level* is typically used to denote differences between base characters, for example, that in English, *a* should sort before *b* (in ICU syntax used throughout this section, we write this as `a < b`). The *secondary level* is typically used to compare characters with and without diacritical marks; for example, in some writing system, we may require for *as* to come before *às* before *at*, requiring us to set *a* to sort before *à* at the secondary level (`a << à`). This level can also be used to compare variant forms of letters, if this is required by the writing system. The *tertiary level* is typically used to distinguish between upper and lower case characters, (for example, by setting `A <<< a`), but may also be tailored to fit the needs of the writing system. Finally, the *quaternary level* can only be used to sort punctuation or for certain operations involving Japanese text. If strings are equivalent at all four levels, a *tie-breaker level* is used to sort these strings, usually on the basis of Unicode codepoint values. This applies to strings that are inputted in canonically equivalent representations.

Because of the complexity of the Church Slavonic writing system, four levels of comparison are required; however, as we have mentioned, tailoring of the quaternary level is not allowed, so only three levels can be used practically. The UCA does allow for an optional *case level*, which is an intermediate level between Level 2 and Level 3 ("Level 2.5"). It is designed to allow differences in case to have a higher-level priority than other tertiary differences, which, strictly speaking, is not what we need; however, we can use this feature to allow case differences to be handled separately form non-case tertiary differences where this is needed.[26]

We propose the following approach to implementing Church Slavonic collation in the UCA. At the *primary level*, we distinguish between the base characters of the writing system, which are sorted in the following order:

а < б < в < г < д = ꙁ < е < ж < ѕ < з = ꙁ < и < і < к < л < м < н < о = ꙩ < п < р < с < т < у < ф < х < ц < ч < ш < щ < ъ < ы < ь < ѣ < ю < ꙗ < ѧ < ѯ < ѱ < ѳ < ѵ

This is the order found in the Church Slavonic dictionaries mentioned above. Note that in the context of Synodal Church Slavonic, ᲁ, ꙁ and ꙩ are variant forms that should be treated as equivalent at all levels to their respective base forms, since they carry no semantic information.

At the *secondary level*, we sort the diacritical marks. Of these, the following marks are treated as default (secondary) ignorable: ◌҇ (pokrytie, U+0487), ꙾ (non-combining kavyka, U+A67E) and ◌꙼ (combining kavyka, U+A67C). The remainder of the diacritical marks and combining letters are sorted in the order given below.

`& [first secondary ignorable]` = ◌҃ = ◌꙽ = ꙿ << ◌҆ << ◌́ << ◌̀ << ◌̑ << ◌҆́ << ◌̆ << ◌̈ = ◌̏ << ◌ⷶ << ◌ⷠ << ◌ⷡ << ◌ⷢ << ◌ⷣ << ◌ⷷ << ◌ꙴ << ◌ⷤ << ◌ⷥ << ◌ꙵ << ◌ꙶ << ◌ⷦ << ◌ⷧ << ◌ⷨ << ◌ⷩ << ◌ⷪ << ◌ꙻ << ◌ⷫ << ◌ⷬ << ◌ⷭ << ◌ⷮ << ◌ⷹ << ◌ꙷ << ◌ꚞ << ◌ⷯ << ◌ⷰ << ◌ⷱ << ◌ⷲ << ◌ⷳ << ◌ⸯ = ◌ꙸ = ⸯ << ◌ꙹ << ꙿ = ◌꙽ = ◌ꙺ << ◌ⷺ << ◌ⷻ << ◌ⷼ << ◌ⷽ << ◌ⷴ

As we have mentioned, accentuation in Church Slavonic, among other things, is used to indicate stress, and it is natural for words with a stress on an initial syllable to sort before words with stress on a final

[26] In our situation, we would like the case level to be a "Level 3.5", in other words, for case differences to have a lower priority than other tertiary differences, but this is not supported by the UCA. However, since casing is not very important in Church Slavonic, this limitation does not have any significant impact on our implementation.

syllable. In addition, lettered titloi – even in instances where they have been shifted from the position of the stress – should also sort from left to right. These expectations require us to specify what is called *backward secondary sorting*. Under this scheme, words are sorted at the secondary level according to the last accent difference as opposed to the first accent difference; the same specification is used in Canadian French and Latvian. The difference in the two sorting schemes can be seen from the following example:

	Secondary sorting	
	Forward	Backward
1	два̀	дв҃а
2	дваⷯ	двⷯа
3	дв҃а	два̀
4	двⷯа	дваⷯ

At the *tertiary level*, we specify the correct sort order of the variant characters used to distinguish grammatical forms. Thus, we specify that Е (U+0415) <<< Є (U+0404) and that е (U+0435) <<< є (U+0454); we specify as well that ѻ <<< о <<< ѡ, that оу (digraph uk) <<< ꙋ (monograph uk) <<< у (the letter u),[27] that ꙗ <<< ѧ, and the analogous uppercase comparisons as in the first example, *mutatis mutandis.* Note that the sort order ѻ <<< о <<< ѡ is required because the wide form ѻ sometimes occurs in the medial position (*cf.* the words і҆ѻппі́ю (1 Maccabees 10:75) < і҆о́ппы (Joshua 19:46)). As we have mentioned, the UCA also handles case differences at the tertiary level, though we can specify the use of a case level and the case-first option to force uppercase letters to sort before lowercase letters. This creates an additional, case-based level of comparison, though it gives case a higher priority than orthographic variation. Alternatively, lowercase letters may be forced to sort before uppercase letters at this case-based level, if desired.

The Synodal Church Slavonic collation rules in ICU syntax are provided in Listing 5.1. A few remarks are warranted about these rules. Observe that we have specified one contraction (a *contraction* is a sequence consisting of two or more letters that is considered as a single letter in sorting). In our instance, the sequence оу is a contraction and should always sort in the order т < оу <<< ꙋ, since the digraph is always pronounced as [u] and never as a diphthong. We note that the Unicode Standard also encodes a standalone character for the digraph, U+0479 Cyrillic Small Letter Uk (and its uppercase analog), but, for reasons explained in Section 2, those codepoints should not be used in encoding Church Slavonic. Since, nonetheless, the characters U+0478 / U+0479 are available in Unicode without a canonical decomposition, we make them equivalent to their respective decomposed sequences under collation. The same applies to the erroneously encoded grapheme ѽ (U+047D; see below) and its uppercase analog and to the combining character Es-Te (U+2DF5), which is treated as equivalent to the sequence Combining Es (U+2DED) Combining Te (U+2DEE).

Instances where a contraction should be avoided (that is, where the digraph оу should be treated as two characters for purposes of collation and not as one character) should not arise when working with standard Synodal Church Slavonic text. But if such a need does, in fact, arise, the UCA provides for a method to avoid the contraction by placing U+034F COMBINING GRAPHEME JOINER between the two characters (*e.g.*: о + ͏ + у). The CGJ does not change the visual appearance of the digraph, but its presence prevents the digraph from being treated as a contraction under collation. Note that despite its name, the Combining Grapheme Joiner does not join graphemes (in fact, its function in this instance is, in some sense, the opposite of "joining") and is not a combining character; see Section 3 for more information.

In addition to one contraction, our collation tailoring also provides for one *expansion*, that is, a letter that sorts as if it were a sequence of more than one letter. The letter ѿ should sort as ѿ = ѡт. It is true that ѿ is treated as a standalone character in computus tables, where it sorts between х and ц. However,

[27] In Synodal Church Slavonic, the letter у is only used as a numeral or as part of the digraph оу.

Listing 5.1: Collation Rules for Synodal Church Slavonic in ICU Syntax

```
[caseLevel on]
[caseFirst upper]
[backwards 2]
& [first primary ignorable] = \; = \: = \\ = \. = \- = \, = \* = – = — = \_ = \uA673 =
    \u0482 = \u20DD = \u0488 = \u0489= \uA670= \uA671= \uA672 = ☩
& [first secondary ignorable] = \u0487 = \uA67C = \uA67E << \u0485 << \u0486 << \
   u0301 << \u0300 << \u0311 << \u0483 << \u0306 << \u0308 = \u030F << \u2DF6 << \
   u2DE0 << \u2DE1 << \u2DE2 << \u2DE3 << \u2DF7 << \uA674 << \u2DE4 << \u2DE5
   << \uA675 << \uA676 << \u2DE6 << \u2DE7 << \u2DE8 << \u2DE9 << \u2DEA << \
   uA67B << \u2DEB << \u2DEC << \u2DED << \u2DEE << \u2DF9 << \uA677 << \uA69E
   << \u2DEF << \u2DF0 << \u2DF1 << \u2DF2 << \u2DF3 << \u033E = \uA678 = \
   u2E2F << \uA679 << \uA67F = \uA67D = \uA67A << \u2DFA << \u2DFB << \u2DFE <<
     \u2DFC << \u2DFD << \u2DF4
& \u2DED\u2DEE = \u2DF5
& д = ᲁ
& е <<< Е <<< є <<< Є
& ж <<< Ж < ѕ <<< Ѕ < з = ꙁ <<< З = Ꙁ
& и <<< И < і <<< І
& и = й / \u0306
& И = Й / \u0306
& і = ї / \u0308
& І = Ї / \u0308
& н <<< Н < ѻ <<< Ѻ <<< о = ᲂ <<< О <<< \u0461 <<< \u0460 <<< \uA64D <<< \uA64C
& \uA64C\u0486\u0311 = \u047C
& \uA64D\u0486\u0311 = \u047D
& Ѡт = Ѿ
& ѡт = ѿ
& т <<< Т < \u0479 = \u043E\u0443 = \u1C82\u0443 <<< \u0478 = \u041E\u0443 = \u041E\
   u0423 <<< \uA64B <<< \uA64A <<< \u0443 <<< \u0423
& э <<< Э < ѣ <<< Ѣ
& ю <<< Ю < ѫ <<< Ѫ < я <<< Я < ꙗ <<< Ꙗ <<< ѧ <<< Ѧ < ѯ <<< Ѯ < ѱ <<< Ѱ < ѳ <<< Ѳ <
    ѵ <<< Ѵ
& ѵ = ѷ / \u030F
& Ѵ = Ѷ / \u030F
& 🕀 < 🕁 < 🕂 < 🕃 < 🕄
```

treating it as an expansion gives the more natural word order in all other settings; in any case, this is implemented in all dictionaries that we have consulted, and computus has more to do with mathematics than language.

Finally, note that we also provide a tailoring for various punctuation characters, numerical symbols and Typicon symbols. The punctuation and numerical characters are set to primary ignorable (numerical methods are beyond the scope of collation and are discussed below). The standard Typicon symbols are sorted in order of decreasing solemnity, which allows for parsing and manipulation of liturgical information.

Unicode encodes a number of precomposed glyphs in the Cyrillic blocks. Some of these glyphs sort as standalone letters in various Slavic languages (for example, the Russian letters *ё* and *й*; the letter *ў* in Belarussian). In Church Slavonic, all of these letters should be decomposed to their base letter and one (or more) diacritical marks; moreover, if their canonically equivalent digraph representations are specified by the DUCET as contractions, such contractions should be suppressed. This particularly applies to the graphemes й (U+0439), ї (U+0457), ѐ (U+0450), ѝ (U+045D) and ѷ (U+0477) and their uppercase analogs. Note also that the grapheme ѿ (U+047D), which was erroneously named in Unicode as “Cyrillic Small

Letter Omega with Titlo", for purposes of sorting should be decomposed as the base Broad Omega ꙍ (U+A64D), followed by the Psili (U+0486) and Inverted Breve (U+0311). While this character and its uppercase analog do not have a canonical decomposition specified in Unicode, the potential encoding ambiguity is resolved by the collation tailoring by making the character U+047D equivalent to the decomposed sequence. In Synodal Church Slavonic, the Broad Omega is merely a graphical variant of the Omega ѡ (U+0461), but in the poluustav recension, Broad Omega does have semantic meaning, thus we specify that ѡ <<< ꙍ. Since in the Synodal recension, the Broad Omega is only used for the grapheme ѽ, such a specification does not affect the collation of Synodal Slavonic in any meaningful way.

As of this writing, the authors are in the process of submitting relevant collation data and other language data for the Church Slavonic locale to Unicode's Common Locale Data Repository (CLDR) and users are encouraged to rely on the data available through the CLDR. International Components for Unicode (ICU), a set of open source libraries for Unicode support and software internationalization, provide the basic tools necessary for implementing collation in C/C++ and Java-based settings; other implementations of the UCA are also available. For more information, consult Unicode Technical Standard #10 (Unicode Collation Algorithm) and the ICU documentation.[28]

5.2 Collation in Other Recensions of Church Slavonic

While Synodal Church Slavonic has standardized orthographic rules that make it relatively straightforward to state an expected sort order, and, therefore, to write down a collation tailoring, earlier recensions of Church Slavonic lack fixed orthographic rules. The number of graphical variants is larger – this is especially true of the manuscript tradition – and these variants include both glyphs that clearly serve as purely graphical variants (reversed forms such as ѕ and ꙅ or the variants of O with different configurations of "eyes") as well as variants that may have a semantic meaning, such as the iotated characters (ꙗ, ѩ, and so forth) or the soft consonants indicating palatalization (see Section 3.5). In addition, earlier recensions use a greater variety of diacritical marks, and the rules governing the usage of such marks neither appear to be completely standardized nor have been sufficiently well researched and documented. While there is a need for a collation tailoring that would encompass ustav and poluustav era Church Slavonic so that text from these recensions may be manipulated on the computer, the authors do not claim that the tailoring provided below is final or definitive. Rather, this should be viewed as one potential approach to implement collation for earlier recensions of Church Slavonic.

Basically, we propose to expand the same principles that were used to construct the tailoring for Synodal Church Slavonic, above. Namely, the primary level is used to make distinctions between the base characters of the writing system; the secondary level is used for ordering the diacritical marks and superscript letters; and the tertiary level is used for ordering variant characters that have a semantic, or otherwise meaningful, distinction. Since, generally speaking, character variation is of interest to palæographers, we assume that most variant characters should, in fact, be distinguished for purposes of collation. Casing may be handled at the additional case level by specifying that uppercase letters sort before their lowercase analogs (or vice-versa).

We begin by writing down the base characters of the writing system – the letters of the Cyrillic alphabet used in Church Slavonic of the ustav recension. In designing the collation tailoring for Synodal Church Slavonic, we needed for the sort order of the characters to be based on their phonetic value in Russian. But for working with earlier recensions, the more traditional order of letters given in Old Slavonic manuscripts is probably desirable. The order presented below follows the order of letters in the Cyrillic and Glagolitic writing systems based on manuscript evidence (specifically, acrostic prayers), as presented by Izotov (2007, pp. 15ff). At the primary level, we have the following order:

[28] A useful guide to collation in ICU is available at http://userguide.icu-project.org/collation.

а < б < в < г < д < е < ж < ѕ < з < и < і < ћ < к < л < м < н < о < п < р < с < т < у < ф < х < ѡ < ц < ч < ҁ < ш < щ < ъ < ы < ь < ѣ < ю < ѭ < ѧ < ѫ < ѩ < ꙗ < ѥ < ѯ < ѱ < ѳ

Note that the character ҁ is only used as a numerical symbol for 90, and does not have phonetic value. While Izotov (2007, *op. cit.*) gives it at the end of the alphabet, we put it next to the letter ч, which is used for 90 in modern Church Slavonic. In addition, observe that, while in the tailoring for Synodal Church Slavonic, we treated ѡ as a variant of о and ꙗ as a variant of ѧ, for ustav-era Church Slavonic we treat them as separate letters. The letters ꙗ and ѧ had different phonetic values in Old Slavonic, although this phonetic distinction did not exist among Eastern Slavs (Uspensky, 1987, p. 127); for working with poluustav-era text, it may make sense to treat the characters as variants of each other as is done in the tailoring for Synodal Church Slavonic.

Next, we consider the diacritical marks used in Old Slavonic. These are compared at the secondary level in the following order:

`& [first secondary ignorable]` = ◌҇ = ◌̆ = ˘ = ◌̋ << ◌҆ << ◌҅ << ◌́ << ◌̀ << ◌̑ << ◌҃ = ◌̄ = ◌꙯ << ◌̆ << ◌̈ = ◌̎ << ◌҄ << ◌ⷶ << ◌ⷠ << ◌ⷡ << ◌ⷢ << ◌ⷣ << ◌ⷷ << ◌ꙴ << ◌ⷤ << ◌ⷥ << ◌ꙵ << ◌ꙶ << ◌ⷸ << ◌ⷦ << ◌ⷧ << ◌ⷨ << ◌ⷩ << ◌ⷪ << ◌ⷫ << ◌ⷬ << ◌ⷭ << ◌ⷮ << ◌ⷹ << ◌ꙷ << ◌ꚞ << ◌ⷯ << ◌ꙻ << ◌ⷰ << ◌ⷱ << ◌ⷲ << ◌ⷳ << ◌꙽ = ◌ꙸ = ꙿ << ◌ꙹ << ꙾ = ◌꙼ = ◌ꙺ << ◌ⷺ << ◌ⷻ << ◌ⷿ << ◌ⷽ << ◌ⷾ << ◌ⷼ << ◌ꚟ << ◌ⷴ

Here the sort order defined above for Synodal Church Slavonic is taken as a starting point, to which we add the additional diacritical marks and characters not found in the Synodal recension. The *okovavy* (◌⃛) are treated as an ignorable character and the variants of the *titlo* – the overline and the *vzmet* – are treated as equivalent to the *titlo* since they are graphical variations only. We sort the rough breathing (which is encountered in rare instances) following the soft breathing and the palatalization mark before the superscript letters. The latter are given in the same order as their base forms.

Finally, we must specify the sort order for the remaining characters, which we are considering to be variants of the base characters. Some characters should be treated as decomposables (expansions) for purposes of collation. These include the ligature ѿ and the soft (palatalized) vowels, which are treated as being equivalent to the base form followed by the palatalization mark (ꙣ = д҄; see also our discussion of palatalization in Section 3.5). Other characters are treated as graphical variants of the base character and distinguished from their base forms at the tertiary level. Within the tertiary level, graphical variants of the same character are almost always sorted in Unicode codepoint order. The proposed collation tailoring for ustav-recension Church Slavonic is given in ICU syntax in Listing 5.2. We stress that this tailoring is intended more as a guideline, and that an actual tailoring may, in fact, need to take into account the specific needs of a research project and the textual data in question. For example, researchers studying poluustav-era texts, especially printed texts, may need to rely more on the sort order of modern Church Slavonic than on the sort order for ustav-era texts given in this subsection. In any case, the rules given in Listing 5.2 and the discussion in this subsection and are intended more to provide sufficient information for developers to create their own collation tailorings for earlier recensions of Church Slavonic that fit their particular needs, rather than as a definitive result.

5.3 String Comparison via Collation

In this subsection, we demonstrate how the specified collation tailoring for Church Slavonic may be used for simple string comparison without requiring complex morphological analysis. Consider, for example, the problem of matching word forms where the spelling may not be standard or when accentuation is of no interest. Simple codepoint-by-codepoint comparison would only return a true match if the two strings are identical at the codepoint level, or, at least, canonically equivalent. Thus, the strings рабо́мъ and рабѡ́мъ

Listing 5.2: Possible Collation Rules for ustav-era Church Slavonic in ICU Syntax

```
[caseLevel on]
[caseFirst upper]
[backwards 2]
& [first primary ignorable] = \; = \: = \\ = \. = \- = \, = \* = – = — = \_ = \uA673 =
    \u0482 = \u20DD = \u0488 = \u0489= \uA670= \uA671= \uA672 = ✠
& [first secondary ignorable] = \u0487 = \uA67C = \uA67E = \u030B << \u0486 << \u0485
    << \u0301 << \u0300 << \u0311 << \u0483 = \u0305 = \uA66F << \u0306 << \u0308 = \
   u030F << \u0484 << \u2DF6 << \u2DE0 << \u2DE1 << \u2DE2 << \u2DE3 << \u2DF7 << \
   uA674 << \u2DE4 << \u2DE5 << \uA675 << \uA676 << \u2DF8 << \u2DE6 << \u2DE7 << \
   u2DE8 << \u2DE9 << \u2DEA << \u2DEB << \u2DEC << \u2DED << \u2DEE << \u2DF9 << \
   uA677 << \uA69E << \u2DEF << \uA67B << \u2DF0 << \u2DF1 << \u2DF2 << \u2DF3 << \
   u033E = \uA678 = \u2E2F << \uA679 << \uA67F = \uA67D = \uA67A << \u2DFA << \u2DFB
    << \u2DFF << \u2DFD << \u2DFE << \u2DFC << \uA69F << \u2DF4
& \u2DED\u2DEE = \u2DF5
& в <<< ᲀ
& д <<< ᲁ
& д = д҄ / \u0484
& Д = Д҄ / \u0484
& Е <<< е <<< Є <<< є <<< Э <<< э
& Ж <<< ж < Ѕ <<< ѕ <<< Ꙅ <<< ꙅ <<< Ꙃ <<< ꙃ < З <<< з <<< Ꙁ <<< ꙁ
& И <<< и < І <<< і <<< Ꙇ <<< ꙇ
& и = й / \u0306
& И = Й / \u0306
& і = ї / \u0308
& І = Ї / \u0308
& л҄ = л / \u0484
& Л҄ = Л / \u0484
& м҄ = м / \u0484
& М҄ = М / \u0484
& Н <<< н < Ѻ <<< ѻ <<< О <<< о <<< ᲂ <<< Ꙩ <<< ꙩ <<< Ꙫ <<< ꙫ <<< Ꙭ <<< ꙭ <<< ꙮ <<<
   Ꚙ <<< ꚙ <<< Ꚛ <<< ꚛ
& н҄ = н / \u0484
& Н҄ = Н / \u0484
& С <<< с <<< ᲃ < Т <<< т <<< ᲄ <<< ᲅ < Оу = Оу = ОУ <<< ᲂу = оу = ᲂу <<< Ꙋ <<< ꙋ <<<
   ᲈ <<< У <<< у <<< Ѵ <<< ѵ
& ѵ = ѷ / \u030F
& Ѵ = Ѷ / \u030F
& Х <<< х < Ѡ <<< ѡ <<< Ꙍ <<< ꙍ
& \uA64C\u0486\u0311 = \u047C
& \uA64D\u0486\u0311 = \u047D
& Ѡт = Ѿ
& ѡт = ѿ
& Ц <<< ц <<< Ꙡ<<< ꙡ
& Ч <<< ч < Ҁ <<< ҁ
& Ъ <<< ъ <<< ᲆ < Ы <<< ы <<< Ꙑ <<< ꙑ < Ь <<< ь < Ҍ <<< ҍ < Ѣ <<< ѣ <<< ᲇ <<< Ꙓ
   <<< ꙓ < Ю <<< ю <<< Ꙕ <<< ꙕ < Ѭ <<< ѭ < Ѧ <<< ѧ <<< Ꙙ <<< ꙙ < Ѫ <<< ѫ
   <<< Ꙛ <<< ꙛ < Ѩ <<< ѩ <<< Ꙝ <<< ꙝ < Ꙗ <<< ꙗ < Ѥ <<< ѥ < Ѯ <<< ѯ < Ѱ <<< ѱ
   < Ѳ <<< ѳ
& ⊕ < ⛣ < ✢ < ⁘ < ⁘
```

Table 14: Results of collation-based matching against the wordform рабѡ́мъ.

Comparison level	рабѡ́мъ	рабо́мъ	рабѡмъ	рабомъ
Level 3	True	False	False	False
Level 3 ignoring Level 2	True	False	True	False
Level 2	True	True	False	False
Level 1	True	True	True	True

are unequal. It is, of course, possible to perform fuzzy string matching via the use of regular expressions such as /раб(о|ѡ)[.]?мъ/, but this has the potential to become very complicated very quickly, especially if one needs to specify all possible combinations of diacritical marks and given that not all programming languages or software correctly implement Unicode characters in regular expressions.

The alternative is to rely on the collation table for string matching and comparison. The collation-based comparison returns "true" if the two expressions evaluate to the same numerical weight at a desired level. Because we have specified our tailoring in such a way that the levels collect diacritical marks and variant letterforms, specifying the desired level of comparison allows us to easily compare strings in contexts where we may wish to allow for different orthographies of the same phoneme, for ignoring stress position, or for ignoring diacritical marks completely. The latter is especially useful when a user wishes to search Church Slavonic text for a query inputted in modern (*e.g.*, Russian) orthography, an extremely common application given that many users do not have a Church Slavonic input method installed.

The results in Table 14 demonstrate a computer program's attempt to match the Church Slavonic word рабѡ́мъ (dative plural) against itself and the wordforms рабо́мъ (instrumental singular), рабѡмъ and рабомъ (accents stripped completely) at three different levels of comparison.[29] As can be seen from the table, this approach allows for simple and customizable fuzzy matching without regular expressions or any morphological analysis and is especially useful in applied settings such as creating a search engine.

6 Input Methods

Our discussion of Church Slavonic typography cannot be complete without considering the issue of how Church Slavonic characters can be input into the computer. For most end users, this is the most important question that arises when working with a writing system in Unicode. That is, users are usually not concerned with encoding (how characters are stored internally by the system), but with simple input of characters (the availability of input methods) and the correct appearance of glyphs (the presence of adequate fonts). In this section, we discuss possible input methods for Church Slavonic.

ISO/IEC 14755 is an international standard that defines the design of input methods for Unicode. This standard describes one possible solution, called the *basic method*, which allows for Unicode characters to be inputted by their codepoint. The user presses a certain combination of keys (the *beginning sequence*), then types the numerical codepoint of the desired character, and then presses another combination of keys (the *ending sequence*). The input system interprets this combination and generates the graphic character mapped at the numerical codepoint. For example, in GNOME applications under GNU/Linux, this method is implemented by typing Ctrl ⇧ Shift U (the beginning sequence), then the hexadecimal codepoint (*e.g.*, 0430 for the Cyrillic Small Letter A), then Space or ↵ Enter (the ending sequence). Similar functionality is available in other ISO/IEC 14755-conformant graphical systems. While this method provides for a standard way to input characters not available on a keyboard, it has its obvious limitations: inputting text in this way is a slow and tedious process requiring the user to know (or to be able to readily look up) the hexadecimal codepoints of the desired characters. This method is impractical for most users.

[29] Note that the fourth level provides no additional functionality since it cannot be tailored.

ISO/IEC 14755 also defines a *screen selection entry method* popular with many users. The method uses a graphical application that provides a visual list of characters (usually organized by writing system or by Unicode block) and allows their selection with a pointing device (mouse). Examples of such an interface include the Character Map application in Microsoft Windows and the `gucharmap` GNOME application for GNU/Linux. Some common applications also provide their own mechanisms of this kind, such as Insert → Symbol in Microsoft Word or Insert → Special Character in LibreOffice. While this may be a convenient solution for inputting the occasional character, it is likewise slow. It has an additional pitfall: because Unicode encodes many characters that have a similar visual appearance but are intended for different purposes, it may be difficult to select the correct character without consulting the Unicode documentation. To prevent the unintended misuse of characters, this method can be recommended only for experienced, "Unicode-savvy" users. Furthermore, this input method can result in an additional problem: the insertion of a character from a graphical interface often breaks the text input stream in an application by beginning a new formatting block, which prevents the rendering system from executing commands given in the lookup tables of a "smart" font. As a result, the glyph substitution and glyph positioning rules for a character are not executed, and, by consequence, diacritical marks or ligatures appear to be "broken." Users typically complain to a font developer about a perceived problem with the font, but, in fact, the problem is caused by a programming deficiency in the application. Thus, the screen selection entry method should likewise be avoided in most situations.

A more satisfactory approach requires a custom keyboard layout, which allows the user to type Church Slavonic characters directly on the keyboard. Since modern keyboards send scancodes rather than characters to the operating system, a Church Slavonic keyboard layout can be implemented by designing the relevant software that converts the scancodes to the needed characters. The process of designing keyboard layout software has become fairly straightforward for most modern operating systems, and, in principle, each user can create customized keyboard layouts to suit his own specific needs. However, in practice, most users will not go to the trouble of designing their own keyboard layouts, so standard keyboard layout software should be supplied with the operating system or be readily available for download and installation. Unfortunately, no Church Slavonic keyboard layout software is currently shipped with any operating system; in this section, we consider two keyboard layouts that could be standardized and readily distributed: a Russian Extended keyboard that allows the user to enter both Church Slavonic and modern Russian characters,[30] and a Church Slavonic ("professional") keyboard that has been optimized for users typesetting large amounts of Church Slavonic text.

6.1 Russian Extended Keyboard

A keyboard contains *character keys* (such as Q) for entering characters and *modifier keys*, which modify the function of character keys but produce no characters themselves. To achieve the modified functionality, a modifier key is held down while a character key is struck. The first standard modifier key is the ⇧ Shift key, which gives access to the first set of additional characters (so-called "Level 2"). On the standard Russian ЙЦУКЕН keyboard, Level 2 contains the uppercase characters; thus, pressing the Q key produces the Russian letter й, but pressing Q while holding down either ⇧ Shift key produces Й. Note that Q represents the key that produces "q" on the QWERTY keyboard; we reference the QWERTY keyboard throughout this section since it is the most common keyboard layout for the Latin script.

To gain access to additional characters that may be required in a writing system, further modifier keys are introduced. The second standard modifier key is the AltGr (*alternate graphemes* or *secondary shift*),

[30] We discuss only the case of the Russian extended keyboard because Russian is by far the most widely used language recorded with the Cyrillic script. However, all of the approaches to designing this keyboard layout could be used to design a keyboard layout based on a Ukrainian, Bulgarian, Serbian, *etc.*, input method for users of other linguistic and ethnic backgrounds.

which gives access to a Level 3. In most operating systems, the right Alt key functions as AltGr; on some operating systems, either Alt key may be used as a secondary shift and on Apple OS X systems typically the secondary shift is the ⌥ Option key. Holding down both the AltGr and ⇧ Shift keys simultaneously gives access to an additional level of characters ("Level 4").

In addition to typical character keys and modifier keys, keyboard layouts can also use dead keys. A *dead key* is a special kind of modifier key that, instead of being held while another key is struck (like the ⇧ Shift key), is pressed and released before the other key. The dead key does not generate any character by itself (hence, it appears to be "dead"); instead, it modifies the character generated by the key struck immediately after. Typically, dead keys are used for typesetting characters with diacritical marks: for example, on many keyboards designed for the Latin script, the ` key is a dead key that produces the grave accent (on such keyboards, for example, striking ` and then E produces è, that is, e with a grave accent).

The Russian Extended keyboard allows users who type Russian on the standard ЙЦУКЕН keyboard to access additional characters used for Church Slavonic via Level 3 and Level 4. The keys on Level 1 and Level 2 (presented in Figure 16) are mapped in the same way as in the ЙЦУКЕН layout with the exception of the ` key being mapped as the superscription dead key (its function is described below). This key on the Russian ЙЦУКЕН keyboard is mapped to the Russian letter "ё" (*yo*), but since "ё" is rarely used in typography and in most contexts can be inserted via an autocorrect feature, its omission from Level 1 is warranted. If the "ё" is needed, it may still be accessed on Level 3 by typing AltGr ` (and AltGr ⇧ Shift ` for the uppercase "Ё"). The key combination ⇧ Shift ` is presently unmapped; it is reserved for potential future use as another dead key.

On Level 3 (presented in Figure 17 together with Level 4), the characters used commonly in Church Slavonic, but not used in Russian, are mapped. The uppercase forms of these characters are accessible on Level 4, together with additional variants and various Typicon symbols. In general, the Church Slavonic characters are mapped on these levels either to the same key as their modern Russian analogs or other graphically similar characters on Level 1 (for example, since the letter O (о) is mapped to J, the letter Wide O (ѻ) is mapped to AltGr J); or in mnemonically intuitive ways (for example, the Narrow O (ᲂ) is mapped to AltGr 0); or in the same manner as in the Church Slavonic standard layout, discussed below, which allows users to interchange between the two layouts more easily. The keyboard layout also provides access to three types of spaces: the usual U+0020 SPACE at Space, U+00A0 NO-BREAK SPACE at ⇧ Shift Space and U+202F NARROW NO-BREAK SPACE at AltGr Space (see the discussion of Spacing in Section 3.6 for usage guidelines).

This keyboard layout provides a *superscription dead key*, mapped to the ` key, although here the term "dead key" is used somewhat more broadly. Instead of modifying the character generated by the next keystroke, this dead key allows the next keystroke to produce a different, but related, character. Striking the superscription dead key informs the input system that the next keystroke should produce a combining (superscript) character. Thus, striking ` and then C produces U+2DED COMBINING CYRILLIC LETTER ES (◌ⷭ) instead of the character Es (с). Any combining Cyrillic letter that has been encoded in Unicode may be accessed via this method, and on any level of the keyboard. For example, striking ` and then A produces U+A69E COMBINING CYRILLIC LETTER EF (◌ꚞ) while striking ` and then AltGr A produces U+2DF4 COMBINING CYRILLIC LETTER FITA (◌ⷴ). However, striking ` and then AltGr R will not produce anything, since a combining Ksi has not been encoded in Unicode. Note that in all instances, superscript (combining) characters entered with this dead key should be rendered *without* the Pokrytie, since this method is primarily intended for working with palæographic editions where the use of Tilto, Pokrytie, or other supralineation, is inconsistent.

On some rendering systems (for example, in IBus used on GNU/Linux), when the superscription dead key is struck, a dotted circle will appear, indicating to the user that the next character will be a superscript; on other rendering system, nothing will appear until after the next keystroke, and so the ` key will

Figure 16: Level 1 (lower row of characters) and Level 2 (upper row of characters) of the Russian Extended keyboard layout

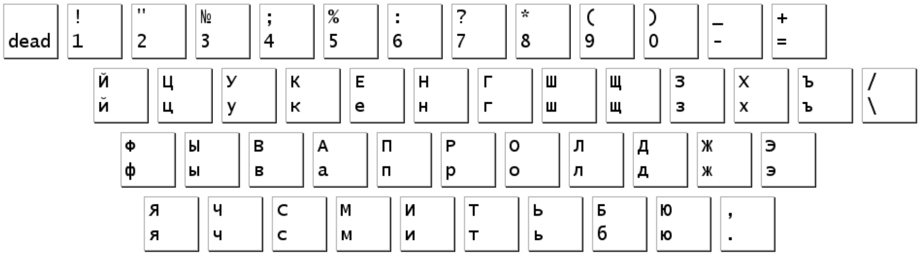
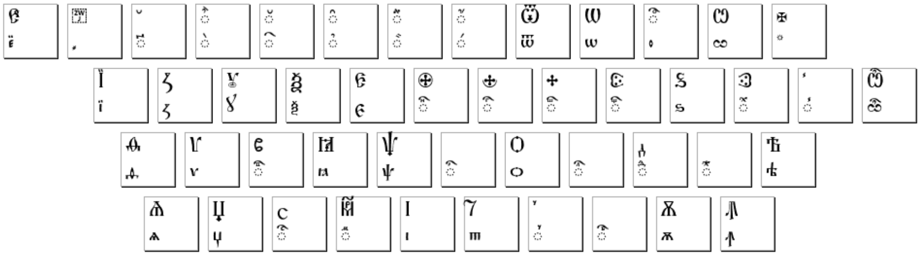


Figure 17: Level 3 (lower row of characters) and Level 4 (upper row of characters) of the Russian Extended keyboard layout



really appear to be "dead". Finally, striking ` and Space produces U+25CC DOTTED CIRCLE. This character (◌) is used to present combining characters in documentation of the Unicode Standard.

Generally, this keyboard layout allows for typesetting modern Russian as well as modern Church Slavonic (the Synodal recension), the Church Slavonic of early printed poluustav books (the early printed books of the 16th and 17th centuries) and ecclesiastical Romanian Cyrillic, but does not provide access to the characters needed for work with early manuscripts. The keyboard layout has not been optimized for Church Slavonic in any way; rather, the goal is to create a simple solution for someone who types mostly in Russian and occasionally needs access to a Church Slavonic character.

6.2 Optimized Church Slavonic Keyboard

For entering large volumes of Church Slavonic text and for working in more complex settings, an optimized keyboard layout is desirable. Such a keyboard layout is designed for a professional user willing to invest some time into learning a new layout. We operate on the assumption that a professional user will be using the touch typing technique, since a user who relies on 'hunt and peck' typing is by definition less concerned with keyboard layout optimization, though some intuitive layout of keys may still be desirable.

The design of an 'optimal' keyboard layout is not an entirely scientific process. Nonetheless, some general principles of optimizing a layout for the touch typing technique can be identified:

- limit the use of weak fingers (little finger and ring finger)
- limit the use of the bottom row of the keyboard
- increase the use of the home row
- limit the finger travel distance between two keystrokes
- balance the use of the left and right hand

Keeping in mind these principles, one can attempt to write down a mathematical model for keyboard entry. A number of projects attempt to do so and to choose a keyboard layout that optimizes some parameters of their mathematical model.[31] The keyboard optimization problem can be expressed as:

$$\min_{C(\mathbf{x})} E(\cdot|C(\mathbf{x}), M(\mathbf{x})) \tag{1}$$

where $\mathbf{x}$ is the vector of Church Slavonic characters; $C(\mathbf{x})$ is the matrix representing their configuration on the keyboard; $M(\mathbf{x})$ is the corpus of existing Church Slavonic text, which provides information about the rules of the Church Slavonic writing system; and the function $E(\cdot)$ is some measure of typing effort. This minimization problem can be solved in principle given proper specifications for $E(\cdot)$; but we need not do that here, since, given the relationship between Church Slavonic and Russian, we are not in need of a fully optimized solution. In fact, instead of solving Problem 1, we are actually looking for the solution to a somewhat more complex problem:

$$\min_{C(\mathbf{x})} E(\cdot|L(C(\mathbf{x}) - C(\mathbf{r})), C(\mathbf{x}), M(\mathbf{x})) \quad \text{given} \quad L(\cdot) \leq \bar{L} \tag{2}$$

where $L(\cdot)$ is the cost of learning a new keyboard layout, $C(\mathbf{r})$ is the matrix of the Russian keyboard layout, and $\bar{L}$ is some maximal learning cost that the user is willing to tolerate. Though in principle this constrained optimization problem may also be solved, given that we have no adequate model for the function $L(\cdot)$, and given the fact the $\bar{L}$ is not constant but depends on the user's preferences, we will not solve Problem 2, but will only rely on its statement for intuition. In any case, we cannot use Carpalx (or similar software) directly since our keyboard layout is too complex, involving four levels as well as dead keys.

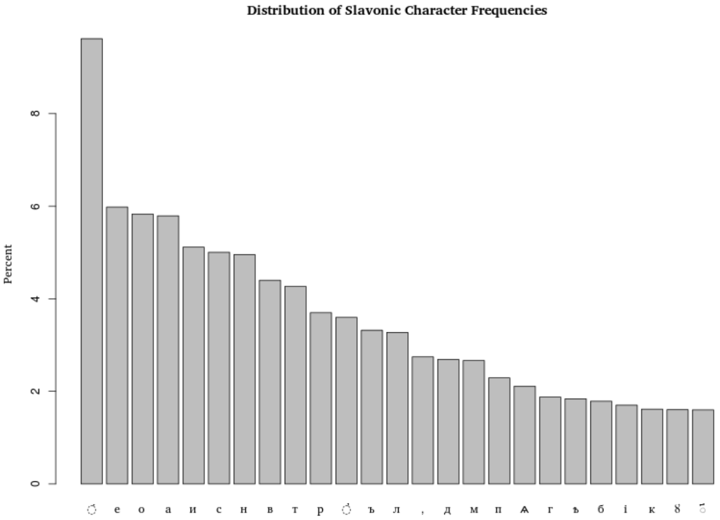
Given Problem 2, a number of observations are in order. First, generally speaking we have $\bar{L}_P > \bar{L}_O$, in other words the professional user is willing to endure a much higher cost of learning a new keyboard layout than an occasional user. For the occasional user, the Russian Extended keyboard is the useful solution, since $\bar{L}$ is minimal. Second, we see that the typing effort $E(\cdot)$ is dependent on the corpus of Church Slavonic text, more specifically, on the frequency with which the various characters occur in Church Slavonic. It would be a mistake to assume that the character frequency for Church Slavonic words is similar to modern Russian, and, as the chart in Figure 18 demonstrates, the most widely used Church Slavonic character is in fact the acute accent ◌́. Observe also that the hard sign (ъ) occurs quite frequently, although it is hardly ever encountered in Russian text, and hence on the Russian keyboard has been relegated to the] key. Note also that capital letters in Church Slavonic are quite infrequent, only used at the beginning of a sentence and in titling. The frequency of the character variants used in Church Slavonic is higher than the frequency of capital letters. Since $L(\cdot)$ is also a function of $(C(\mathbf{x}) - C(\mathbf{r}))$ – in other words, the cost of learning is dependent on the difference between the Russian and Church Slavonic layouts – our approach to placing the variants on the keyboard in an intuitive way is justified. For example, the keyboard positions for ѻ and ᲂ should somehow be related to the position of о and the positions for ѿ, ꙍ, and ѽ should probably have some relationship with ѡ.

Given all of this intuition, we are now ready to present an "optimized" Church Slavonic keyboard layout for use by the professional user. We will not attempt to solve the mathematical problem directly, but rather we operate on the basis of intuition. The Russian ЙЦУКЕН layout is again taken as a starting point, but it is significantly reconfigured in order to minimize the typing effort $E(\cdot)$ by making a set of changes $(C(\mathbf{x}) - C(\mathbf{r}))$.

The optimized Church Slavonic keyboard is a four-level layout with two dead keys. The layout of Levels 1 and 2 is presented in Figure 19. Level 1 follows the Russian ЙЦУКЕН layout, but with some exceptions: ꙋ is mapped to the E key instead of у and ѧ is placed on the Z key instead of я (the latter character

[31] One such example is the Carpalx keyboard optimization algorithm; see http://mkweb.bcgsc.ca/carpalx/ for more information.

Figure 18: Frequency of letters occurring in the modern Church Slavonic corpus



is not used in Church Slavonic). In addition, the hard sign (ъ) has been relocated to the 4 key instead of the] key for reasons outlined above. Generally, the number row provides access to the most widely used diacritical marks and the additional Church Slavonic letters ѣ, ı (the dotless variant), ѡ and ѵ.[32]

Level 2 (the Shift Level) is accessed by holding down either ⇧ Shift key, but, unlike the ЙЦУКЕН or QWERTY layouts, it provides access to additional characters rather than uppercase forms. Uppercase forms occur less frequently, and, hence, have been moved to Level 3, since Level 2 is easier to access (holding down either ⇧ Shift key is much easier than holding down the right Alt key, which is the standard AltGr implementation). The placement of the variant forms is intuitive; thus є҆ is available as ⇧ Shift T since the base form є is mapped to the T key. For some keys, the Level 2 mapping is the combining (superscript) version of the letter, and in all cases where this combining character occurs with the *pokrytie* in modern Church Slavonic, the keystroke produces a sequence of the combining letter followed by the *pokrytie* (for example, ⇧ Shift C produces U+2DED COMBINING CYRILLIC LETTER ES and U+0487 COMBINING CYRILLIC POKRYTIE). If in modern Church Slavonic the combining character occurs without the *pokrytie*, then the keystroke only produces the combining character (for example, ⇧ Shift L produces U+2DE3 COMBINING CYRILLIC LETTER DE). The keyboard layout also provides on Level 2 the digraph Onik (ᲂу) as a single keystroke (⇧ Shift E) mapped to the sequence U+1C82 CYRILLIC SMALL LETTER NARROW O U+0443 CYRILLIC SMALL LETTER U. The letter у (used in modern Church Slavonic only as a numeral) is also provided as a standalone character at ⇧ Shift 6. Digraphs are also provided for the commonly used accents *iso* (◌҆́) and *apostrof* (◌҆̀) and for the standalone forms of the decimal i (ї) and the Izhitsa (ѷ).

Level 3 and Level 4 of the keyboard layout are presented in Figure 20; Level 3 (the AltGr level) is accessed by holding down the AltGr key, as discussed above. The uppercase forms of the characters mapped on Level 1 are available on Level 3 and the uppercase forms of the characters mapped on Level 2 — on Level 4, which is accessed by holding down both the AltGr and ⇧ Shift keys. The keyboard layout also uses the ⇪ CapsLk in the traditional manner of providing access to the capital letters, but since the capital letters have been moved to Level 3, this is now defined as a Level 3 lock. In addition, Level

[32] The original design of this keyboard (for working with the HIP markup language) is due to Daniil Dremachyov. It was redesigned by the authors for working with Unicode.

Figure 19: Level 1 (lower row of characters) and Level 2 (upper row of characters) of the Church Slavonic keyboard layout

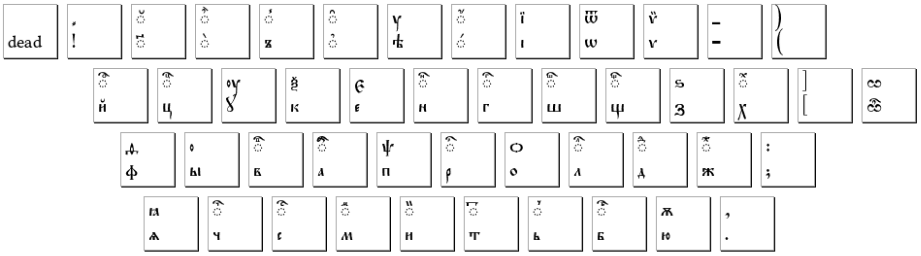
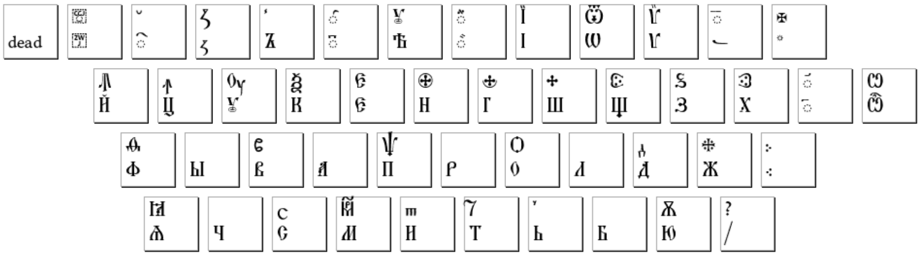


Figure 20: Level 3 (lower row of characters) and Level 4 (upper row of characters) of the Church Slavonic keyboard layout



4 provides access to a number of less frequently used variant forms; the Typicon symbols and various typographical embellishments commonly encountered in Church Slavonic books; the characters used for double titloi and Cyrillic supralineation (see the discussion in Section 3.4); and quick access to Unicode format control characters used to control the appearance of ligatures and diacritical marks: U+200D ZERO WIDTH JOINER and U+034F COMBINING GRAPHEME JOINER at AltGr 1 and AltGr ⇧ Shift 1, respectively.

The optimized keyboard layout provides two dead keys: the superscription dead key, described in detail in the previous section, and again mapped at `, and a *variation dead key*, mapped at ⇧ Shift `. The variation dead key may be used to access various character variants that only occur in early ustav-era manuscripts. The implementation is similar to the superscription dead key: pressing ⇧ Shift ` and then some key produces a character that is related to the one mapped to that key. For example, ⇧ Shift ` T produces U+0465 CYRILLIC SMALL LETTER IOTIFIED E (ѥ); pressing ⇧ Shift ` ⇧ Shift P produces U+A645 CYRILLIC SMALL LETTER REVERSED DZE. The full list of characters available via this dead key is given in the keyboard layout documentation provided by the authors.[33] Finally, the same three types of spaces are available in this keyboard layout as in the Russian Extended layout described above.

6.3 Installation and Editing of Keyboard Layouts

We refer the reader to the documentation shipped with our keyboard layout drivers for a complete discussion of features and installation instructions. Here, we briefly mention how keyboard layout drivers are implemented in operating systems and what software is available should the user choose to modify a keyboard layout.

[33] See http://www.ponomar.net/cu_support/keyboard.html for more information.

On Microsoft Windows, keyboard layouts are stored in keyboard layout DLL's, which are located in the standard System32 directory of Windows. On most Windows installations, the actual path to this directory is `C:\Windows\System32` or `C:\WinNT\System32`. The names of all layout files usually begin with `kbd` and have the `.dll` extension. In addition to installing the DLL in the relevant path, the new keyboard layout must be registered in the relevant section of the Windows Registry. All of this means that Windows keyboard layouts are actually quite complicated pieces of software. Luckily, Microsoft provides the free Microsoft Keyboard Layout Creator (MSKLC) tool for editing keyboard layouts and generating the DLL's and installation programs – Windows installer packages and the `setup.exe` program – which makes editing keyboard layouts or creating new layouts for Windows a relatively painless process.

One pitfall occurs because Microsoft does not recognize Church Slavonic as a valid language. Hence, the keyboard layout will be registered and displayed in the Windows Language Bar as a Russian layout. This may be problematic for users who wish to have both the Russian ЙЦУКЕН keyboard and the optimized Church Slavonic keyboard installed simultaneously. In addition, some software will automatically set the language of text entered with the layout to Russian, a "feature" that can be quite annoying. The MSKLC tool will also produce warnings during validation, indicating that some characters mapped on the keyboard layout that are not part of Microsoft's standard windows-1251 codepage for Russian. Unfortunately, no known workarounds exist, and users are encouraged to contact Microsoft with a request to add Church Slavonic to the list of valid languages.

On Apple systems, starting with version 10.2 (Jaguar), keyboard layout drivers for Apple OS X are basically simple XML files in a specific format with the `.keylayout` extension. For some added functionality, the XML file may be bundled with some additional information – metadata describing the keyboard layout and icon files for the flag displayed on the language bar – into an archive with a `.bundle` extension. Though, in principle, these files can be created by hand (using a text editor), the simplest way to generate keyboard layouts for OS X is by using the Ukelele app, which is provided by SIL International. Ukelele stands for Unicode Keyboard Layout Editor and is compatible with OS X versions 10.2 and later. OS X users desiring to edit keyboard layouts are encouraged to consult the extensive documentation provided by SIL in order to become familiar with the application and the keyboard layout design process.

Under GNU/Linux and other Unix-based operating systems, Church Slavonic keyboard entry is provided by the Intelligent Input Bus (`IBus`), a full-featured and user-friendly input method user interface. Note that it is not possible to provide keyboard entry for Church Slavonic using X keyboard extension (XKB) directly because of limitations in the XKB architecture. Keyboard layout data for `IBus` are provided via the m17n database, part of the `m17n` library, which realizes multilingualization for GNU/Linux. The authors maintain the package `m17n-cu`, which provides Church Slavonic keyboard layout data for `IBus` as well as collation data for `glibc` and other localization information. The keyboard layout files in the `m17n-cu` package are the files with the `.mim` extension. These files may be edited using any standard GNU/Linux text editing utility, such as the GNU Text Editor (`gedit`), `emacs`, or `vi`. The files describe the keyboard layout using a syntax that resembles the LISP programming language. Documentation for this syntax is available on the m17n project website.

7 Implementation of Cyrillic Numerals

The final section of our document addresses the issue of working with numerals used in Church Slavonic. Since numerals quite frequently occur in typography (for example, for page numbering and timestamps), they need to be correctly supported by word-processing and other software. Since the 18th century, the use of European digits has become standard in many publications; however, the older system of Cyrillic numerals may still be encountered in ecclesiastical publications.

Cyrillic numerals (also called Slavic numerals or Slavonic numerals) are a numbering system derived

Table 15: List of Digits used in Cyrillic Numerals

Disp.	Codept.	Value	Disp.	Codept.	Value	Disp.	Codept.	Value
а҃	U+0430	1	і҃	U+0456	10	р҃	U+0440	100
в҃	U+0432	2	к҃	U+043A	20	с҃	U+0441	200
г҃	U+0433	3	л҃	U+043B	30	т҃	U+0442	300
д҃	U+0434	4	м҃	U+043C	40	у҃	U+0443	400
є҃	U+0454	5	н҃	U+043D	50	ф҃	U+0444	500
ѕ҃	U+0455	6	ѯ҃	U+046F	60	х҃	U+0445	600
з҃	U+0437	7	о҃	U+043E	70	ѱ҃	U+0471	700
и҃	U+0438	8	п҃	U+043F	80	ѿ	U+047F	800
ѳ҃	U+0472	9	ч҃	U+0447	90	ц҃	U+0446	900

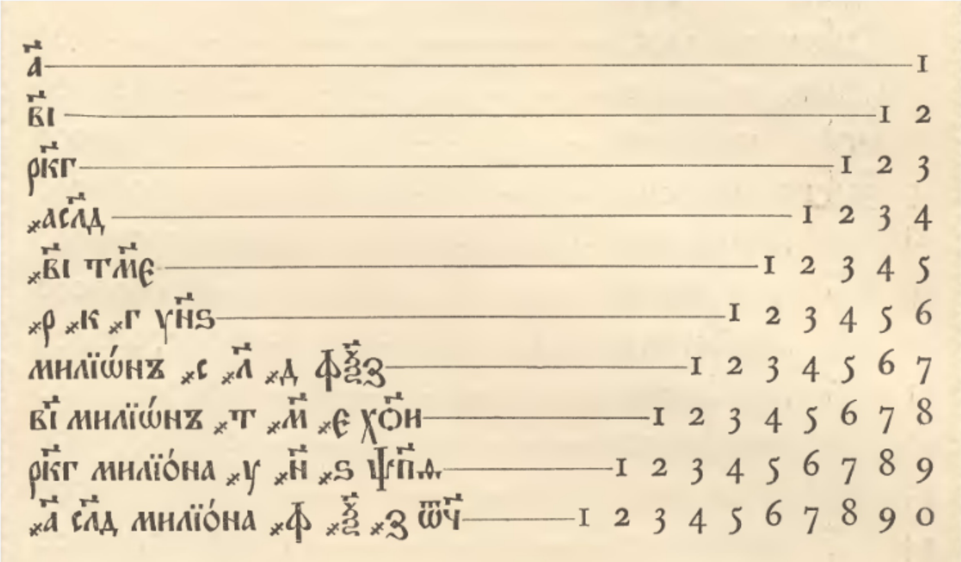
from Greek (Ionian) numerals. This system was originally developed in Bulgaria in the 10th century (probably together with the Cyrillic alphabet), and subsequently spread to East Slavic countries as well. Although the system is clearly derived from Greek, it has certain characteristic features which make it a unique numbering system: namely, it uses Cyrillic rather than Greek characters, and the order in which characters are written is based on the numeral's pronunciation in Church Slavonic.

In Table 15 we present the characters used in the Cyrillic numeral system, together with their Unicode codepoints and their numerical values. This Table reproduces the information provided in standard Church Slavonic textbooks, such as those of Archb. Alypy (Gamanovich) (p. 22), Vorob′yeva (2008, p. 44), and Pletnyeva et al. (1996, p. 43). The system is quasi-decimal, with a separate grapheme assigned to each unit, each multiple of ten, and each multiple of one hundred. The characters in the Table are displayed together with the *titlo* (encoded in Unicode as U+0483 COMBINING CYRILLIC TITLO), which is placed above the character to indicate that the text is numeric (analogous to the usage of the *dexia keraia* in Greek). In a single-character numeral, the *titlo* occurs directly over the character (examples: а҃ for one, ф҃ for 500). The numeral 800 is usually represented using the Cyrillic Letter Ot (ѿ), and the *titlo* is often omitted in this case. Although some textbooks use the representation with the Cyrillic Letter Omega and the *titlo* (ѡ҃; for example, see Vorob′yeva (2008, p. 44)), we have reviewed Church Slavonic printed books and have found that this usage is not well attested and the usage with the ѿ should be preferred.

In multiple-character numerals, the characters are written left-to-right in the order in which they are read in the Church Slavonic language. Generally, graphemes representing higher order digits are written first; thus, we have: а҃ (one), к҃а (21), л҃а (31), рк҃а (121), and so forth. However, numbers between 11 and 19 are written with the lower order numeral first, to reflect how they are read. Thus, we find: а҃і (11; read as є҆диннона́десѧть; lit., *one-after-ten*); в҃і (12; read as двана́десѧть; lit., *two-after-ten*); and so forth to ѳ҃і (19; read as девѧтьна́десѧть; lit., *nine-after-ten*). This order is preserved for higher numbers involving the teens, thus: ра҃і (111); рѳ҃і (119). To represent numerals of 1,000 (one thousand) and above, the symbol ҂ (U+0482 CYRILLIC THOUSANDS SIGN) is prepended to the numeral (for example, ҂а҃ represents 1,000; ҂вд҃і represents 2,014).

For longer numerals, the *titlo* usually occurs over the penultimate digit (examples: к҃а for 21, ҂арк҃а for 1,121). With the numeral 800, different usages may be encountered. In the editions printed by the Holy Governing Synod in Russia in the early 1800's, the *titlo* was printed over the ѿ (for example, ҂аѿ҃з for 1,807), but in subsequent editions (from the 1820's onward), the *titlo* was placed following the ѿ, even if that meant placing it in the final position (for example, ҂аѿѯ҃ for 1,860). We recommend that the later convention should be followed. We also recommend placing the *titlo* over the ѿ when indicating the numeral 800 by itself (ѿ҃), even though this is often not done, in order to distinguish the numeral from the Church Slavonic preposition “from” for purposes of algorithmic conversion.

Figure 21: Examples of higher-order Cyrillic numerals. Source: *Arifmetika*, Moscow, 1703, as reprinted by Baranov (1914, p. 59)



а҃	1
в҃і	1 2
р҃кг	1 2 3
҂ас҃лд	1 2 3 4
҂в҃і т҃мє	1 2 3 4 5
҂р ҂к ҂г у҃нѕ	1 2 3 4 5 6
мил҇іѡнъ ҂с ҂л҃ ҂д ф҃ѯз	1 2 3 4 5 6 7
в҃і мил҇іѡнъ ҂т ҂м҃ ҂є х҃ои	1 2 3 4 5 6 7 8
р҃кг мил҇іѡна ҂у ҂н҃ ҂ѕ ѱ҃пѳ	1 2 3 4 5 6 7 8 9
҂а҃ с҃лд мил҇іѡна ҂ф ҂ѯ҃ ҂з ѡ҃ч	1 2 3 4 5 6 7 8 9 0

There is no consistent system for representing higher multiples of one thousand since numerals higher than ten thousand are hardly ever encountered in Church Slavonic text. Even numerals between five thousand and ten thousand usually occur only to record years *anno mundi* according to the Byzantine reckoning. In some sources, such as Archb. Alypy (Gamanovich) (p. 22), higher multiples of one thousand may be written by adding further occurrences of the thousands symbol (҂); thus, the string ҂҂а҃ represents one million (one thousand thousands). In other sources, one may find alternative methods; for example, in the *Arifmetika* of Leonty Magnitsky published in 1703, the thousands may be written as a separate group of numerals, preceded by the thousands symbol, separated from the rest of the digits by a (non-breaking) space (see line 5 in Figure 21). The same approach is implied by Pletnyeva et al. (1996, p. 43). Alternatively, a copy of the thousands symbol may be placed before each digit (see line 6 in Figure 21). Note that without a clear set of rules, ambiguities may arise when representing higher order numerals. For example, the numerals 1,010 and 11,000 need to be distinguished (and not both written as ҂а҃і). We recommend that the non-breaking space (U+00A0 NO-BREAK SPACE or U+202F NARROW NO-BREAK SPACE) be used as a thousands separator for numerals higher than 10,000 whenever it is warranted (so, for example, 11,001 should be written as ҂а҃і а҃), following the usage in the *Arifmetika*. Note that a thousands separator is not used for numerals below 10,000. In instances where confusion still arises and cannot be resolved using spacing only, we recommend placing the Cyrillic Thousands Sign before each digit. For example, 11,000 is then recorded as ҂а҃҂і, distinguishing it from 1,010 (which is recorded as ҂а҃і).

In early manuscripts, special symbols were used for representing numerals of ten thousand and higher. These symbols (the *Cyrillic number signs*) are listed in Section 2.2. Since they are not used in Synodal Church Slavonic, they do not need to be automatically generated by software and are not discussed in this section. They should still be available in fonts since they may occur in academic publications and various didactic materials. It should also be noted that in ustav and poluustav manuscripts, the *titlo* may balance over several digits of the numeral or over the entire numeral. Working with such more complex supralineation is described in Section 3.4; in Synodal Church Slavonic, the *titlo* always balances only over one character.

In the Unicode Common Locale Data Repository (CLDR), numbering systems are classified as being of one of two different types: algorithmic and numeric. Numeric systems are simple, decimal based systems that do not require complex rules but instead can be implemented by specifying which Unicode codepoints are used to encode the digits. Algorithmic systems are more complex, and require rules indicating their proper formatting, as, for example, for Roman and Greek numerals (Unicode Consortium, 2015b, Part 3). The Cyrillic numeral system is likewise an algorithmic numbering system. In the CLDR, rules for defining algorithmic numbering systems are specified using the Rule-Based Number Formatting (RBNF) syntax, a set of rules that maps a number to a string representation. In Listing 8.1, we provide the RBNF-formatted rules for generating Cyrillic numerals.[34] Because of the complexity of the Cyrillic numeral system, several *rule sets* are required. Only the rule set labeled `%cyrillic-lower-titlo` (line 87 of Listing 8.1) is public and is called externally; the remaining rule sets are private rule sets used internally to consider various cases. For an overview of the RBNF syntax, consult the ICU documentation.[35] Note that though the CLDR requires rules for negative numbers, zero, and decimals, these cases are not specified in the Cyrillic numeral system and are not used in Church Slavonic publications.

The RBNF-formatted rules may be used to provide support for Cyrillic numerals in CLDR-aware applications and in any software that relies on ICU for internationalization support. In other settings, for example in TEX or in LibreOffice, developers may need to provide code for algorithms generating Cyrillic numerals as part of a localization package. While we have outlined the properties of the numbering system and provided the RBNF-formatted rules as one example, providing individual implementations is beyond the scope of this paper.

8 Limitations

This Technical Note has outlined the essential features required for support of Church Slavonic text in modern computing settings: encoding (within the framework of the Unicode Standard), font design (relying on modern “smart” font technologies such as OpenType), collation, input method design, and numerals. A number of features required for proper Church Slavonic typography are still not supported. These include algorithmic hyphenation and spell checking. The authors have been involved in the development of hyphenation patterns for Synodal Slavonic – a feature urgently needed not only for traditional typography, but also for support of Church Slavonic text on mobile devices with limited effective screen size – and hope to present results in the near future. Algorithmic spell checking of Church Slavonic – a feature that would not only assist users involved in typesetting modern liturgical texts but also could be used to check for errors in a Slavonic corpus – presents various challenges given the highly inflected nature of the language. To our knowledge, some research into spell checking using the Hunspell program has been undertaken, but is still at a nascent stage. Work with a corpus of Church Slavonic text in an information retrieval (IR) system requires also the availability of a stemmer – an algorithm for reducing inflected words to their root form. Such a stemmer is needed, for example, for the proper functionality of dictionaries and search engines. Though not directly part of typography, work in this area is nonetheless related to it and, to a large extent, depends heavily on many of the basic building blocks – encoding and collation – described in this Technical Note. There is, thus, still much work left to do for full support of Church Slavonic typography. This work will assist not only those involved in the typesetting of modern Church Slavonic texts for the day-to-day liturgical life of the Orthodox and Byzantine Catholic churches, but also anyone working to preserve, study, and appreciate the literary heritage of the Slavic peoples.

[34] The authors would like to thank Kent Karlsson for help in drafting these rules.

[35] See `http://www.icu-project.org/apiref/icu4j/com/ibm/icu/text/RuleBasedNumberFormat.html`.

Listing 8.1: Rule Based Number Formatting rules for Cyrillic numerals

```
%%cyrillic-lower-1-10:
  1: \u0430;
  2: \u0432;
  3: \u0433;
  4: \u0434;
  5: \u0454;
  6: \u0455;
  7: \u0437;
  8: \u0438;
  9: \u0473;
 10: \u0456;
 11: ERROR;

%%cyrillic-lower-final:
  0: \u0483;
  1: \u0483=%%cyrillic-lower-1-10=;
  11: \u0430\u0483\u0456;
  12: \u0432\u0483\u0456;
  13: \u0433\u0483\u0456;
  14: \u0434\u0483\u0456;
  15: \u0454\u0483\u0456;
  16: \u0455\u0483\u0456;
  17: \u0437\u0483\u0456;
  18: \u0438\u0483\u0456;
  19: \u0473\u0483\u0456;
  20: \u0483\u043A;
  21: \u043A>>;
  30: \u0483\u043B;
  31: \u043B>>;
  40: \u0483\u043C;
  41: \u043C>>;
  50: \u0483\u043D;
  51: \u043D>>;
  60: \u0483\u046F;
  61: \u046F>>;
  70: \u0483\u047B;
  71: \u047B>>;
  80: \u0483\u043F;
  81: \u043F>>;
  90: \u0483\u0447;
  91: \u0447>>;
  100: ERROR;

%%cyrillic-lower-post:
  0: \u0483;
  1: =%cyrillic-lower-titlo=;

%%cyrillic-lower-thousands:
  0: \u0483;
  1: \u0483\u0482\u0430;
  2: \u0483\u0482\u0432;
  3: \u0483\u0482\u0433;
  4: \u0483\u0482\u0434;
  5: \u0483\u0482\u0454;
  6: \u0483\u0482\u0455;
  7: \u0483\u0482\u0437;
  8: \u0483\u0482\u0438;
 9: \u0483\u0482\u0473;
10: \u0483\u0482\u0456;
11: \u0482\u0430\u0483\u0482\u0456;
12: \u0482\u0432\u0483\u0482\u0456;
13: \u0482\u0433\u0483\u0482\u0456;
14: \u0482\u0434\u0483\u0482\u0456;
15: \u0482\u0454\u0483\u0482\u0456;
16: \u0482\u0455\u0483\u0482\u0456;
17: \u0482\u0437\u0483\u0482\u0456;
18: \u0482\u0438\u0483\u0482\u0456;
19: \u0482\u0473\u0483\u0482\u0456;
20: \u0482\u043A>>;
30: \u0482\u043B>>;
40: \u0482\u043C>>;
50: \u0482\u043D>>;
60: \u0482\u046F>>;
70: \u0482\u047B>>;
80: \u0482\u043F>>;
90: \u0482\u0447>>;
100: \u0482\u0440>>;
200: \u0482\u0441>>;
300: \u0482\u0442>>;
400: \u0482\u0443>>;
500: \u0482\u0444>>;
600: \u0482\u0445>>;
700: \u0482\u0471>>;
800: \u0482\u047F>>;
900: \u0482\u0446>>;

%cyrillic-lower-titlo:
 -x: ->>;
 x.x: <<.>>>;
 0: 0\u0483;
 1: =%%cyrillic-lower-1-10=\u0483;
 11: \u0430\u0483\u0456;
 12: \u0432\u0483\u0456;
 13: \u0433\u0483\u0456;
 14: \u0434\u0483\u0456;
 15: \u0454\u0483\u0456;
 16: \u0455\u0483\u0456;
 17: \u0437\u0483\u0456;
 18: \u0438\u0483\u0456;
 19: \u0472\u0483\u0456;
 20: \u043A>%%cyrillic-lower-final>;
 30: \u043B>%%cyrillic-lower-final>;
 40: \u043C>%%cyrillic-lower-final>;
 50: \u043D>%%cyrillic-lower-final>;
 60: \u046F>%%cyrillic-lower-final>;
 70: \u047B>%%cyrillic-lower-final>;
 80: \u043F>%%cyrillic-lower-final>;
 90: \u0447>%%cyrillic-lower-final>;
 100: \u0440>%%cyrillic-lower-final>;
 200: \u0441>%%cyrillic-lower-final>;
 300: \u0442>%%cyrillic-lower-final>;
 400: \u0443>%%cyrillic-lower-final>;
 500: \u0444>%%cyrillic-lower-final>;
 600: \u0445>%%cyrillic-lower-final>;
 700: \u0471>%%cyrillic-lower-final>;
 800: \u047F\u0483;
```

```
801: \u047F>>;
900: \u0446>%%cyrillic-lower-final>;
1000: \u0482<%%cyrillic-lower-1-10<>%%cyrillic-lower-post>;
10000/1000: \u0482<<[ >>];
11000/1000: <%%cyrillic-lower-thousands<[ >>];
1000000: \u0482\u0482<<[ >>];
1000000000: \u0482\u0482\u0482<<[ >>];
1000000000000: \u0482\u0482\u0482\u0482<<[ >>];
1000000000000000: \u0482\u0482\u0482\u0482\u0482<<[ >>];
1000000000000000000: =#,##0=;
```

Appendix A Transliteration Tables

Transliteration, which is frequently needed in computer-based manipulation of data, is the process of converting characters from one writing system to another. In our case, we provide rules for converting between the Glagolitic and Cyrillic alphabets; for converting Church Slavonic Cyrillic to modern Russian characters ("Russification"); and for the romanization of Church Slavonic (conversion of Cyrillic to Latin characters) according to two different systems. The Unicode CLDR distinguishes between two types of transliteration: a reversible system (where the text in the original writing system can be recovered) and a non-reversible system, where the exact orthography of the original text is lost. CLDR also specifies other criteria for transliteration systems: where possible, transliteration systems should be *standard* (follow existing standards or conventions); *predictable* (lend themselves to algorithmic manipulation without knowledge of the language); and *pronounceable* (meaning that the "resulting characters have reasonable pronunciations in the target script") (Unicode Transliteration Guidelines). These guidelines are mutually exclusive, and no transliteration system can be designed to fulfill all of these requirements simultaneously.

We have attempted to present currently existing systems (or extend such systems to Church Slavonic, where necessary) in the chart below. The system of Glagolitic to Cyrillic transcription (due to Jagić (1879)) is standard (in the sense that it has become classical and is used by all scholars in this field) and both pronounceable and reversible (with the exception of two combining Cyrillic characters missing from Unicode). Russification of Church Slavonic is by definition non-reversible but pronounceable; it is standard in the sense that it reflects current Church Slavonic pronunciation. Note that this system is based on the pronunciation of modern Church Slavonic.

Two systems of romanization of Church Slavonic are also presented. In the first system, we extend the BGN/PCGN romanization system for Russian to Church Slavonic. This system is non-reversible but pronounceable and is fairly intuitive for anglophone speakers. Note that in the processes of Church Slavonic Russification and BGN/PCGN romanization, we do not consider the resolution of superscript characters and titloi. These must be either resolved algorithmically (that is, by expanding all abbreviations via the use of a dictionary) or can be recorded using lowercase letters in parentheses (*e.g.*: ◌ⷣ → (д), etc.).

The second romanization system is a reversible system created by the authors by extending the ISO 9:1995 standard for Cyrillic romanization. ISO 9:1995 (which agrees with System A of GOST 7.79 (2000)) is useful because it is language-independent and reversible (it transliterates each character by a one character equivalent). We have added to this system additional characters used in Church Slavonic and modified the transliteration of other characters to use a more intuitive approach (for example, the grapheme J is used in our system to indicate iotified vowels since the Cyrillic letter J is not used in Church Slavonic). All additions to or modification of ISO 9 are shaded in light gray. Because the information presented in this chart is preliminary, we publish these results in an Appendix in the hope that further discussion among industry members will lead to consensus that will allow for a formal extension to support Church Slavonic characters in ISO 9:1995 and related standards.

Glagolitic			Cyrillic					Latin				
			Church Slavonic			Russian		BGN/PCGN		ISO 9:1995 Ext.		
Ⰰ 2C00	ⰰ 2C30	◌𞀀 1E000	А 0410	а 0430	◌ⷶ 2DF6	А 0410	а 0430	A 0041	a 0061	A 0041	a 0061	a̭ a+032D
Ⰱ 2C01	ⰱ 2C31	◌𞀁 1E001	Б 0411	б 0431	◌ⷠ 2DE0	Б 0411	б 0431	B 0042	b 0062	B 0042	b 0062	b̭ b+032D
Ⰲ 2C02	ⰲ 2C32	◌𞀂 1E002	В 0412	в 0432	◌ⷡ 2DE1	В 0412	в 0432	V 0056	v 0076	V 0056	v 0076	v̭ v+032D
				ᲀ 1C80			в 0432		v 0076		v[36] 0076	
Ⰳ 2C03	ⰳ 2C33	◌𞀃 1E003	Г 0413	г 0433	◌ⷢ 2DE2	Г 0413	г 0433	G 0047	g 0067	G 0047	g 0067	g̭ g+032D
			Ґ 0490	ґ 0491		Г 0413	г 0433	G 0047	g 0067	G̀ G+0300	g̀ g+0300	
Ⰴ 2C04	ⰴ 2C34	◌𞀄 1E004	Д 0414	д 0434	◌ⷣ 2DE3	Д 0414	д 0434	D 0044	d 0064	D 0044	d 0064	ḓ d+032D
				ᲁ 1C81			д 0434		d 0064		d 0064	
			Ꙣ A662	ꙣ A663		Д’ 0414	д’ 0434	D’ 0044	d’ 0064	Ḋ D+0307	ḋ d+0307	
Ⰵ 2C05	ⰵ 2C35	◌𞀅 1E005	Е 0415	е 0435	◌ⷷ 2DF7	Е 0415	е 0435	Ye (E) 0045	ye (e) 0065	E 0045	e 0065	ḙ e+032D
			Є 0404	є 0454	◌ꙴ A674	Е 0415	е 0435	Ye (E) 0045	ye (e)[37] 0065	Ê E+0302	ê e+0302	ḙ̂ ê+032D
			Э 042D	э 044D		Е 0415	е 0435	E 0415	e 0435	Ê͔ Ê+0354	ê͔ ê+0354	
Ⰶ 2C06	ⰶ 2C36	◌𞀆 1E006	Ж 0416	ж 0436	◌ⷤ 2DE4	Ж 0416	ж 0436	Zh 005A 0068	zh 007A 0068	Ž Z+030C	ž z+030C	ž̭ ž+032D
Ⰷ 2C07	ⰷ 2C37		Ѕ 0405	ѕ 0455		З 0417	з 0437	Z 005A	z 007A	Ẑ Z+0302	ẑ z+0302	
			Ꙃ A642	ꙃ A643		З 0417	з 0437	Z 005A	z 007A	Ẑ̧ Ẑ+0327	ẑ̧ ẑ+0327	
			Ꙅ A644	ꙅ A645		З 0417	з 0437	Z 005A	z 007A	Ẑ͔ Ẑ+0354	ẑ͔ ẑ+0354	
Ⰸ 2C08	ⰸ 2C38	◌𞀈 1E008	З 0417	з 0437	◌ⷥ 2DE5	З 0417	з 0437	Z 005A	z 007A	Z 005A	z 007A	z̭ z+032D
			Ꙁ A640	ꙁ A641		З 0417	з 0437	Z 005A	z 007A	Z̧ Z+0327	z̧ z+0327	
Ⰻ 2C0B	ⰻ 2C3B	◌𞀋 1E00B	И 0418	и 0438	◌ꙵ A675	И 0418	и 0438	I 0049	i 0069	I 0049	i 0069	i̭ i+032D
Ⰹ 2C09	ⰹ 2C39	◌𞀉 1E009	І 0406	і 0456	◌ꙶ A676	И 0418	и 0438	I 0049	i 0069	Ị I+0323	ị i+0323	ị̭ ị+032D

Continued on next page

[36] Since the character ᲀ and other similar characters fold onto their base forms under casing (see Section 3.7), it is proposed that they should also fold under transliteration.

[37] The form Ye is used whenever э, е and є occur in initial position or after a vowel, semi-vowel, or yer; the form E is used otherwise

Continued from previous page

Glagolitic			Cyrillic					Latin				
			Church Slavonic			Russian		BGN/PCGN		ISO 9:1995 Ext.		
Ⰺ 2C0A	ⰺ 2C3A	◌𞀊 1E00A	Ꙇ A646	ꙇ A647	–[38]	И 0418	и 0438	I 0049	i 0069	Į I+0328	į i+0328	į̭ į+032D
Ⰼ 2C0C	ⰼ 2C3C	◌𞀌 1E00C	Ꙉ A648	ꙉ A649	◌ⷸ 2DF8	Дж 0414	дж 0434	Dzh[39] 0044 (0064) 007A 0048	dzh	Ĝ G+0302	ĝ g+0302	ĝ̭ ĝ+032D
Ⰽ 2C0D	ⰽ 2C3D	◌𞀍 1E00D	К 041A	к 043A	◌ⷦ 2DE6	К 041A	к 043A	K 004B	k 006B	K 004B	k 006B	k̭ k+032D
Ⰾ 2C0E	ⰾ 2C3E	◌𞀎 1E00E	Л 041B	л 043B	◌ⷧ 2DE7	Л 041B	л 043B	L 004C	l 006C	L 004C	l 006C	ḽ l+032D
			Ꙥ A664	ꙥ A665		Л’ 041B	л’ 043B	L’ 004C	l’ 006C	L̇ L+0307	l̇ l+0307	
Ⰿ 2C0F	ⰿ 2C3F	◌𞀏 1E00F	М 041C	м 043C	◌ⷨ 2DE8	М 041C	м 043C	M 004D	m 006D	M 004D	m 006D	m̭ m+032D
			Ꙧ A666	ꙧ A667		М’ 041C	м’ 043C	M’ 004D	m’ 006D	Ṁ M+0307	ṁ m+0307	
Ⱀ 2C10	ⱀ 2C40	◌𞀐 1E010	Н 041D	н 043D	◌ⷩ 2DE9	Н 041D	н 043D	N 004E	n 006E	N 004E	n 006E	ṋ n+032D
			Ҥ 04A4	ҥ 04A5		Н’ 041D	н’ 043D	N’ 004E	n’ 006E	Ṅ N+0307	ṅ n+0307	
Ⱁ 2C11	ⱁ 2C41	◌𞀑 1E011	О 041E	о 043E	◌ⷪ 2DEA	О 041E	о 043E	O 004F	o 006F	O 004F	o 006F	o̭ o+032D
			Ѻ 047A	ѻ 047B		О 041E	о 043E	O 004F	o 006F	Ǒ O+030C	ǒ o+030C	
				ᲂ 1C82			о 043E		o 006F		o 006F	
			Ꙩ A668	ꙩ A669		О 041E	о 043E	O 004F	o 006F	Ȯ O+0307	ȯ o+0307	
			Ꙫ A66A	ꙫ A66B		О 041E	о 043E	O 004F	o 006F	Ö O+0308	ö o+0308	
			Ꙭ A66C	ꙭ A66D		О 041E	о 043E	O 004F	o 006F	Ꝏ̈ Ꝏ+0308	ꝏ̈ ꝏ+0308	
				ꙮ A66E			о 043E		o 006F		o̊ o+030A	
			Ꚙ A698	ꚙ A699		О 041E	о 043E	O 004F	o 006F	Ꝏ A74E	ꝏ A74F	
			Ꚛ A69A	ꚛ A69B		О 041E	о 043E	O 004F	o 006F	O̽ O+033D	o̽ o+033D	
Ⱂ 2C12	ⱂ 2C42	◌𞀒 1E012	П 041F	п 043F	◌ⷫ 2DEB	П 041F	п 043F	P 0050	p 0070	P 0050	p 0070	p̭ p+032D
Ⱃ 2C13	ⱃ 2C43	◌𞀓 1E013	Р 0420	р 0440	◌ⷬ 2DEC	Р 0420	р 0440	R 0052	r 0072	R 0052	r 0072	r̭ r+032D

Continued on next page

[38] A combining Cyrillic Iota is not available in Unicode, making Glagolitic to Cyrillic transliteration not fully complete.

[39] The sequence дж (or ⰴⰶ) is transliterated as d·zh, to distinguish it from the letter Ꙉ (where · is U+2027 HYPHENATION POINT). The same method of disambiguation is used in other instances.

Continued from previous page

Glagolitic			Cyrillic					Latin				
			Church Slavonic			Russian		BGN/PCGN		ISO 9:1995 Ext.		
Ⱄ 2C14	ⱄ 2C44	◌𞀔 1E014	С 0421	с 0441	◌ⷭ 2DED	С 0421	с 0441	S 0053	s 0073	S 0053	s 0073	s̭ s+032D
				ᲃ 1C83			c 0063		s 0073		s 0073	
Ⱅ 2C15	ⱅ 2C45	◌𞀕 1E015	Т 0422	т 0442	◌ⷮ 2DEE	Т 0422	т 0442	T 0054	t 0074	T 0054	t 0074	ṱ t+032D
				ᲄ 1C84			т 0442		t 0074		t 0074	
				ᲅ 1C85			т 0442		t 0074		t 0074	
Ⱆ 2C16	ⱆ 2C46	◌𞀖 1E016	Ꙋ A64A	ꙋ A64B	◌ⷹ 2DF9	У 0423	у 0443	U 0055	u 0075	Ǔ U+030C	ǔ u+030C	ṷ̌ ǔ+032D
				ᲈ 1C88			у 0443		u 0075		ǔ u+030C	
			У 0423	у 0443	◌ꙷ A677	У 0423	у 0443	U 0055	u 0075	U 0055	u 0075	ṷ u+032D
Ⱇ 2C17	ⱇ 2C47	◌𞀗 1E017	Ф 0424	ф 0444	◌ꚞ A69E	Ф 0424	ф 0444	F 0046	f 0066	F 0046	f 0066	f̭ f+032D
Ⱈ 2C18	ⱈ 2C48	◌𞀘 1E018	Х 0425	х 0445	◌ⷯ 2DEF	Х 0425	х 0445	Kh 004B 0068	kh 006B 0068	H 0048	h 0068	h̭ h+032D
Ⱉ 2C19	ⱉ 2C49		Ѡ 0460	ѡ 0461	◌ꙻ A67B	О 041E	о 043E	O 004F	o 006F	O̱ O+0331	o̱ o+0331	o̱̭ o̱+032D
			Ꙍ A64C	ꙍ A64D		О 041E	о 043E	O 004F	o 006F	Ǒ̱ O+0331	ǒ̱ o+0331	
			Ѿ 047E	ѿ 047F		От 041E (043E) 0442	от	Ot 004F (006F) 0074	ot	O̱t O (o)+0331+t	o̱t	
Ⱊ 2C1A	ⱊ[40] 2C4A		П 041F	п 043F		П 041F	п 043F	P 0050	p 0070	P̂ P+0302	p̂ p+0302	
Ⱌ 2C1C	ⱌ 2C4C	◌𞀜 1E01C	Ц 0426	ц 0446	◌ⷰ 2DF0	Ц 0426	ц 0446	Ts 0054 0073	ts 0074 0073	C 0043	c 0063	c̭ c+032D
			Ꙡ A660	ꙡ A661		Ц 0426	ц 0446	Ts 0054	ts 0074	C͔ C+0354	c͔ c+0354	
Ⱍ 2C1D	ⱍ 2C4D	◌𞀝 1E01D	Ч 0427	ч 0447	◌ⷱ 2DF1	Ч 0427	ч 0447	Ch 0043 0068	ch 0063 0068	Č C+030C	č c+030C	č̭ č+032D
			Ҁ 0480	ҁ 0481		Ч 0427	ч 0447	Q 0051	q 0071	Q̌ Q+030C	q̌ q+030C	
Ⱎ 2C1E	ⱎ 2C4E	◌𞀞 1E01E	Ш 0428	ш 0448	◌ⷲ 2DF2	Ш 0428	ш 0448	Sh 0053 0068	sh 0073 0068	Š S+030C	š s+030C	š̭ š+032D
Ⱋ 2C1B	ⱋ 2C4B	◌𞀛 1E01B	Щ 0429	щ 0449	◌ⷳ 2DF3	Щ 0429	щ 0449	Shch 0053 0068 0063 0068	shch 0073 0068 0063 0068	Ŝ S+0302	ŝ s+0302	ŝ̭ ŝ+032D

Continued on next page

[40] The identity and shape of this Glagolitic character are dubious; see Kempgen (2008).

Continued from previous page

Glagolitic			Cyrillic					Latin				
			Church Slavonic			Russian		BGN/PCGN		ISO 9:1995 Ext.		
Ⱏ	ⱏ	◌𞀟	Ъ	ъ	◌ꙸ	Ъ	ъ	″	″	″̱	″	″̭
2C1F	2C4F	1E01F	042A	044A	A678	042A	044A	02BA	02BA	″+0331	02BA	″+032D
				ᲆ			ъ		″		″	
				1C86			044A		02BA		02BA	
ⰟⰉ	ⱏⰹ		Ꙑ	ꙑ		Ы	ы	Y (Y·)	y (y·)	Y̌	y̌	
2C1F	2C4F		A650	A651		042B	044B	0059	0079	Y+030C	y+030C	
2C09	2C39											
Ⱐ	ⱐ	◌𞀠	Ь	ь	◌ꙺ	Ь	ь	ʹ	ʹ	ʹ̱	ʹ	ʹ̭
2C20	2C50	1E020	042C	044C	A67A	042C	044C	02B9	02B9	ʹ+0331	02B9	ʹ+032D
ⰠⰉ	ⱐⰹ		Ы	ы	◌ꙹ	Ы	ы	Y (Y·)	y (y·)	Y	y	y̭
2C20	2C50		042B	044B	A679	042B	044B	0059	0079	0059	0079	y+032D
2C09	2C39											
			Ꙏ	ꙏ		Ъ	ъ	ʹ	ʹ	ˋ̱	ˋ	
			A64E	A64F		042A	044A	02B9	02B9	ˋ+0331	02CB	
Ⱑ	ⱑ	◌𞀡	Ѣ	ѣ	◌ⷺ	Е	е	Ye (E)	ye (e)	Ě	ě	ḙ̌
2C21	2C51	1E021	0462	0463	2DFA	0415	0435	0059 0065	0079 0065	E+030C	011B	ě+032D
				ᲇ			е		ye (e)		ě	
				1C87			0435		0079		011B	
			Ꙓ	ꙓ		Е	е	Ye	ye	JĚ	jě	
			A652	A653		0415	0435	0059 0065	0079 0065	004A+Ě	006A+ḙ̌	
Ⱒ	ⱒ[41]		Х	х		Х	х	Kh	kh	Ĥ	ĥ	
2C22	2C52		0425	0445		0425	0445	004B 0068	006B 0068	H+0302	h+0302	
Ⱓ	ⱓ	◌𞀣	Ю	ю	◌ⷻ	Ю	ю	Yu	yu	JU	ju	jṷ
2C23	2C53	1E023	042E	044E	2DFB	042E	044E	0059	0079	004A+U	006A+u	ju+032D
			Ꙕ	ꙕ		Ю	ю	Yu	yu	JŲ	jų	
			A654	A655		042E	044E	0059	0079	004A+Ų	006A+ų	
			Ꙗ	ꙗ	◌ⷼ	Я	я	Ya	ya	JA	ja	ja̭
			A656	A657	2DFC	042F	044F	0059	0079	004A+A	006A+a	ja+032D
			Ѥ	ѥ	◌ꚟ	Е	е	Ye	ye	JE	je	jḙ
			0464	0465	A69F	0415	0435	0059	0079	004A+E	006A+e	je+032D
Ⱔ	ⱔ	◌𞀤	Ѧ	ѧ	◌ⷽ	Я	я	Ya	ya	Ę	ę	ę̭
2C24	2C54	1E024	0466	0467	2DFD	042F	044F	0059 0061	0079 0061	E+0328	e+0328	ę+032D
			Ꙙ	ꙙ		Я	я	Ya	ya	Ę̌	ę̌	
			A658	A659		042F	044F	0059 0061	0079 0061	Ę+032D	ę+032D	
Ⱘ	ⱘ	◌𞀨	Ѫ	ѫ	◌ⷾ	Ю	ю	Yu	yu	Ǫ	ǫ	ǫ̭[42]
2C28	2C58	1E028	046A	046B	2DFE	042E	044E	0059 0075	0079 0075	O+0328	o+0328	ǫ+032D
			Ꙛ	ꙛ		Ю	ю	Yu	yu	Œ̨	œ̨	
			A65A	A65B		042E	044E	0059 0075	0079 0075	Œ+0328	œ+0328	
Ⱙ	ⱙ	◌𞀩	Ѩ	ѩ	-[43]	Я	я	Ya	ya	JĘ	ję	ję̭

Continued on next page

[41] This character occurs in a small number of Glagolitic manuscripts; it has no Cyrillic analog (Ivanova, 1997, p. 28).

[42] ISO 9:1995 transliterates the character ѫ as ă; we propose instead a more traditional transcription.

[43] A combining Cyrillic Iotated Little Yus has not been encoded in Unicode, possibly by oversight. The authors may propose such a character for encoding in the future.

Continued from previous page

Glagolitic			Cyrillic					Latin				
			Church Slavonic			Russian		BGN/PCGN		ISO 9:1995 Ext.		
2C27	2C57	1E027	0468	0469		042F	044F	0059 0061	0079 0061	004A+Ę	006A+ę	ję+032D
			Ꙝ	ꙝ		Я	я	Ya	ya	JĘ̌	ję̌	
			A65C	A65D		042F	044F	0059 0061	0079 0061	004A+Ę̌	006A+ę̌	
Ⱙ	ⱙ	𞀩	Ѭ	ѭ	ⷿ	Ю	ю	Yu	yu	JǪ	jǫ	jǫ̭
2C29	2C59	1E029	046C	046D	2DFF	042E	044E	0059 0075	0079 0075	004A+Ǫ	006A+ǫ	jǫ+032D
			Ѯ	ѯ		Кс	кс	Ks	ks	KS	ks	
			046E	046F		041A 0421	043A 0441	004B 0053	006B 0073	004B 0053	006B 0073	
			Ѱ	ѱ		Пс	пс	Ps	ps	PS	ps	
			0470	0471		041F 0421	043F 0441	0050 0053	0070 0073	0050 0053	0070 0073	
Ⱚ	ⱚ	𞀪	Ѳ	ѳ	ⷴ	Ф	ф	F	f	F̀	f̀	f̭̀
2C2A	2C5A	1E02A	0472	0473	2DF4	0424	0444	0046	0066	F+0300	f+0300	f̀+032D
Ⱛ	ⱛ		Ѵ	ѵ		И (В)[44]	и (в)	I (V)	i (v)	Ŷ	ŷ	
2C2B	2C5B		0474	0475		0418 (0412)	0438 (0432)	0049 (0056)	0069 (0076)	Y+0302	y+0302	
			Џ	џ		Дж	дж	Dzh	dzh	D̂	d̂	
			040F	045F		0414 0416	0434 0436	0044+zh	0064+zh	D+0302	d+0302	
			Ꙟ	ꙟ		Ын	ын	Yn	yn	ÎN	în	
			A65E	A65F		042B 041D	044B 043D	0059 006E	0079 006E	I+0302+N	i+0302+n	

Appendix B Church Slavonic Abbreviations

This Appendix lists abbreviations (*nomina sacra* and sigla) encountered in Church Slavonic, both of the Synodal recension and of earlier poluustav printed texts. The listing provides character sequences that need to be resolved by a stemmer, parser, or other tool performing string comparison. The resolution of abbreviations itself is beyond the scope of this paper; consult the documentation for the `Lingua::CU` Perl module created by the authors for more details. Second, this listing provides a guide for font developers, indicating which superscript letters occur typically with a pokrytie. Font developers may choose to provide precomposed glyphs for these instances or to support the proper display of such glyphs via mark-to-mark positioning. See Section 4 for more information.

Abbreviations used in Synodal Typography

Abbreviation	Full Form	Meaning
Combining Be (U+2DE0, 1 form, always with pokrytie):		
сꙋⷠ҇	сꙋббѡ́та	Sabbath, Saturday
Combining Ve (U+2DE1, 1 form, always with pokrytie):		
глⷡ҇а, глаⷡ҇	глава̀	chapter
Combining Ge (U+2DE2, 1 form, with or without pokrytie):[45]		

Continued on next page

[44] The character и (in Latin, i) is used whenever the letter ѵ occurs after a consonant or with a *kendema*; the character в (in Latin, v) is used otherwise.

[45] In books published by Holy Trinity Monastery in Jordanville, NY, this superscript letter occurs without a pokrytie; in all other books, it occurs with a pokrytie. It is not clear if the Jordanville books reflect a different typographic tradition or a defect.

Continued from previous page

Abbreviation	Full Form	Meaning
є҆ѵⷢ҇іе, є҆ѵⷢіе	є҆ѵа́нгелїе	Gospel
Combining De (U+2DE3, 15 forms, always without pokrytie):		
бцⷣа, бг҃оро́дица	богоро́дица	Godbearer (Theotokos)
бчⷣенъ	богоро́диченъ	Theotokion
блгⷣть, бл҃года́ть	благода́ть	grace
влⷣка	Влады́ка	Master
влⷣчца	Влады́чица	Mistress (female Master)
мрⷣъ, мⷣръ[46]	мꙋ́дръ	wise (adjective)
млⷣнцъ	младе́нецъ	infant (noun)
нлⷣѧ	недѣ́лѧ	Sunday
првⷣкъ, првⷣенъ, првⷣн–	пра́ведникъ, пра́веденъ	righteous (adjective and noun)
пнⷣе, пнⷣльникъ	понедѣ́льникъ	Monday
прⷣтеча	предте́ча	Forerunner
прпⷣбенъ, прпⷣбн–, препⷣбн–	преподо́бенъ, преподо́бн–	Venerable
поⷣ	подо́бенъ	prosomœon (type of hymn)
срⷣе	среда̀	Wednesday, middle
срⷣце	се́рдце	heart
Combining Zhe (U+2DE4, 2 forms, always without pokrytie):		
дваⷤ	два́жды	twice
триⷤ	три́жды	thrice
Combining Ze (U+2DE5, 2 forms, always without pokrytie):		
праⷥ	пра́здникъ	feast
роⷥ	розво́дъ	expansion[47]
Combining Ka (U+2DE6, 3 forms, always with pokrytie):		
проⷦ҇	прокі́менъ	prokimenon
пѧⷦ҇	пѧто́къ	Friday
чеⷦ҇	четверто́къ	Thursday
Combining El (U+2DE7, 3 forms, always with pokrytie):		
ндⷧ҇ѧ	недѣ́лѧ	Sunday
ѱаⷧ҇	ѱало́мъ	psalm
Combining Em (U+2DE8, 1 form, always without pokrytie):		
риⷨ	ри́млѧнѡмъ	(to the) Romans
Combining En (U+2DE9, 1 form, always with pokrytie):		
солꙋⷩ҇	солꙋ́нѧнѡмъ	(to the) Thessalonians
Combining O (U+2DEA, 2 forms, always with pokrytie):		
прⷪ҇ркъ, прⷪ҇ркъ	проро́къ	Prophet
трⷪ҇ца	тро́ица	Trinity
Combining Er (U+2DEC, 2 forms, always with pokrytie):		
и҆́мⷬ҇къ, и҆́м҃рекъ	и҆́мѧ ре́къ	say name here
втоⷬ҇	вто́рникъ	Tuesday
Combining Es (U+2DED, 33 forms, always with pokrytie):		

Continued on next page

[46] The second form is especially common in compound words, such as премⷣръ.

[47] This term appears in chant books and indicates the expansion of a Znamenny fita.

Continued from previous page

Abbreviation	Full Form	Meaning
а҆пⷭ҇лъ, а҆пⷭ҇толъ	а҆по́столъ	apostle
бжⷭ҇тво̀	божество̀	the Divinity
блгⷭ҇ве́нъ	благослове́нъ	blessed
воскрⷭ҇и́ти	воскреси́ти	to resurrect
воскрⷭ҇нїе	воскресе́нїе	resurrection
гдⷭ҇ь	Госпо́дь	Lord
гпⷭ҇жа̀	Госпожа̀	Lady
глаⷭ҇	гла́съ	tone
двⷭ҇тво	дѣ́вство	virginity
є҆пⷭ҇кпъ, є҆пⷭ҇копъ	є҆пи́скопъ	bishop
і҆ерⷭ҇ли́мъ	і҆ерꙋсали́мъ	Jerusalem
крⷭ҇ти́тель	крести́тель	(St. John) the Baptist
крⷭ҇тъ	кре́стъ	Cross
млⷭ҇рдїе	милосе́рдїе	loving-kindness
млⷭ҇ть	ми́лость	mercy
мнⷭ҇ты́рь	монасты́рь	monastery
мцⷭ҇ъ, мⷭ҇цъ	мѣ́сѧцъ	month
мчⷭ҇ный	мѣ́сѧчный	monthly
нбⷭ҇ный	небе́сный	heavenly
пречⷭ҇тый, пречтⷭ҇ый	пречи́стый	most pure
прⷭ҇нѡ	при́снѡ	always
прⷭ҇то́лъ	престо́лъ	throne, altar
ржⷭ҇тво̀	рождество̀	nativity, offspring
стрⷭ҇тоте́рпецъ	страстоте́рпецъ	passion-bearer
стрⷭ҇ть	стра́сть	passion
трⷭ҇то́е	трисвѧто́е	Trisagion (type of hymn)
хрⷭ҇тїа́нинъ	хрїстїа́нинъ	Christian
хрⷭ҇то́съ	Хрїсто́съ	Christ
црⷭ҇кїй	ца́рскїй	royal
црⷭ҇тво	ца́рство	kingdom
чтⷭ҇ны́й	честны́й	honorable
чтⷭ҇ый, чⷭ҇тый	чи́стый	pure
Combining Ha (U+2DEF, 2 forms, always without pokrytie):		
варꙋⷯ	варꙋ́хъ	Baruch
сⷯ, Сⷯ	сті́хъ	verse
Combining Che (U+2DF1, 1 form, always with pokrytie):		
заⷱ҇	зача́ло	pericope
Combining Fita (U+2DF4, 1 form, always with pokrytie):		
корінⷴ҇	корі́нѳѧнѡмъ	(to the) Corinthians
Words written with a common titlo (U+0483):		
а҆́гг҃лъ	а҆́нгелъ	Angel
бг҃обл҃года́тный	богоблагода́тный	[possessing] God-given grace
бг҃ъ, бж҃-, бз҃-	Бо́гъ, бо́ж-, бо́з-	God (and oblique cases)

Continued on next page

Continued from previous page

Abbreviation	Full Form	Meaning
бл҃гі́й	благі́й	good (long adjective)
бл҃гъ, бл҃ж-, бл҃з-	бла́гъ, бла́ж-, бла́з-	good (short adjective)
бл҃же́нъ	блаже́нъ	blessed (short adjective)
воскр҃си́ти	воскреси́ти	to resurrect
воскр҃ше́нїе	воскреше́нїе	resurrection (gerund)
воцр҃и́ти	воцари́ти	to reign
всн҃овле́нїе	всыновле́нїе	adoption
гл҃авый	глаго́лавый	spoke (past active participle)
гл҃ати	глаго́лати	to speak
гл҃го́лъ	глаго́лъ	word, saying
гл҃ъ	гла́съ	voice, Tone (musical paradigm)
дв҃а	дѣ́ва	Virgin (usually referring to Mary the Theotokos)
дв҃дъ	Даві́дъ	David
дв҃ца	дѣ́вица	young girl, maiden
дн҃сь	дне́сь	today
дн҃ь	де́нь	day
дх҃ъ, дс҃-, дш҃-	дꙋ́хъ, дꙋ́с-, дꙋ́ш-	[Holy] Spirit (and oblique cases)
дш҃а̀	дꙋ́ша̀	soul
і҆н҃съ	І҆нсꙋ́съ	Jesus
і҆н҃ль	І҆сра́нль	Israel
кн҃ги́нѧ	кнѧги́нѧ	princess
кн҃зь	кнѧ́зь	prince
кр҃сти́ти	крести́ти	to baptize (perfective)
кр҃ща́ти	креща́ти	to baptize (imperfective)
мл҃тва	моли́тва	prayer
мр҃їа́мъ, мр҃їа́мь	Марїа́мъ	Maryam
мр҃і́ѧ, мр҃і́а	Марі́а	Mary
мт҃и, мт҃р-, мт҃ер-	ма́ти, ма́тер-	mother (including oblique cases)
мⷱ҇нкъ	мꙋ́ченикъ	martyr
нб҃о, нб҃са̀	не́бо, небеса̀	heaven(s)
нн҃ѣ	ны́нѣ	now
ѻ҆ц҃ъ, ѻ҆́т҃че	Ѻ҆те́цъ, Ѻ҆́тче	Father (usually the person of the Trinity)
ѻ҆ⷱ҇ь	ѻ҆те́чь	of the Father (adjective)
ѻ҆ⷱ҇еск-	ѻ҆те́ческ-	fatherly (adjective)
пл҃ть	пло́ть	flesh
сл҃нечный	со́лнечный	solar
сл҃нце	со́лнце	sun
см҃рть	сме́рть	death
сн҃ъ	Сы́нъ	Son (usually the person of the Trinity)
сп҃са́ти	спаса́ти	to save
сп҃си́тель	спаси́тель	Savior (long form)
сп҃съ	спа́съ	Savior (short form)
ст҃и́ти	свѧти́ти	to make holy, to sanctify (perfective)

Continued on next page

Continued from previous page

Abbreviation	Full Form	Meaning
ст҃ль	свѧти́тель	hierarch
ст҃ъ	свѧ́тъ	holy (adjective)
сщ҃а́ти	свѧща́ти	to make holy (imperfective)
сщ҃е́нникъ	свѧще́нникъ	priest
оу҆ч҃нїе	оу҆че́нїе	teaching, doctrine
оу҆чн҃къ	оу҆чени́къ	disciple
оу҆чт҃ль	оу҆чи́тель	teacher
цр҃ковь	це́рковь	church
цр҃ца	цари́ца	queen
цр҃ь	ца́рь	king
чл҃вѣ́къ, чл҃овѣ́къ	человѣ́къ	male, man
чл҃вѣ́чь, чл҃овѣ́чь	человѣ́чь	of man (adjective)
чл҃ко–	человѣко–	man (in compound words)

Abbreviations used in Poluustav Print Typography

The abbreviations encountered in poluustav (pre-Nikonian) typography are more varied. Here we only list some of the commonly occurring forms; the listing is extensive, but not exhaustive, and is intended primarily as a guideline for font developers. Note that in this section, the orthography reflects the pre-Nikonian usage, which may differ from Synodal-era orthography. The usage of pokrytie is generally standardized, but non-standard usage may also be encountered.

Abbreviation	Full Form	Meaning
Combining A (U+2DF6, without pokrytie):		
на́шⷶ	на́ша	our (fem. nom. sing.)
моⷶ	моѧ̀	my (fem. nom. sing.)
Combining Be (U+2DE0, with pokrytie):		
на ѻⷠ҇	на ѻ҆боро́тѣ	see reverse, on the reverse page
Combining Ve (U+2DE1, with pokrytie):		
глаⷡ҇	глава̀	chapter
слаⷡ҇	сла́ва	glory
оу҆стаⷡ҇	оу҆ста́въ	rule, order, Typicon
Combining Ge (U+2DE2, with pokrytie):		
боⷢ҇	богоро́дниченъ	Theotokion (type of hymn)
ва́шеⷢ҇	ва́шего	your (gen. sing.)
є҆ѵⷢ҇а́лїе, є҆ѵⷢ҇лїе	є҆ѵа́нгелїе	Gospel
є҆ѵⷢ҇ли́стъ	є҆ѵангели́стъ	Evangelist
ст҃аⷢ҇	ста́гѡ	saint (gen. sing.)
чертоⷢ҇	черто́гъ	bridal chamber
Combining De (U+2DE3, without pokrytie):		
бг҃ороⷣ	богоро́дниченъ	Theotokion (type of hymn)
блⷣгть	благода́ть	grace
бⷣца	богоро́дица	Theotokos
влⷣко	влады́ко	Master

Continued on next page

Continued from previous page

Abbreviation	Full Form	Meaning
влⷣчца, влⷣца	влады́чица	Lady
влⷣчество	влады́чество	dominion
в̾ пнеⷣ	въ понедѣ́лникъ	on Monday
коⷣ, коⷣн	конда́къ	Kontakion (type of hymn)
люⷣскі́и	людскі́и	[relating to] the people
млⷣнцъ, млⷣнецъ	младе́нецъ	infant
мⷣрость	мꙋдрость	wisdom
мⷣръ	мꙋдръ	wise
в̾ нлⷣю	въ недѣ́лю	on Sunday
поⷣ	по́дъ or подо́бенъ	under *or* prosomœon (type of hymn)
прпⷣбенъ	преподо́бенъ	venerable
првⷣнъ	пра́веденъ	righteous
првⷣныи	пра́ведныи	righteous
преⷣте́ча, прⷣтча	предоте́ча or предте́че	Forerunner
премⷣрость	премꙋ́дрость	wisdom
премⷣръ	премꙋ́дръ	wise
срⷣца̀	сердца̀	hearts
срⷣце	се́рдце	heart
сѣⷣ	сѣда́ленъ	Sessional Hymn (type of hymn)
ѹ҆щеⷣри	ѹ҆ще́дри	be merciful
Combining De-I (U+2DE3 U+A675, without pokrytie):		
раⷣꙵ	ра́ди	for the sake of
Combining E (U+2DF7, without pokrytie):		
ѻ҆́бщⷷ	ѻ҆́бще	common, general
с҃рдцⷷ	се́рдце	heart
Combining Zhe (U+2DE4, without pokrytie):		
аⷤ	є҆ди́ножды	once
вⷤ	два́жды	twice
гⷤ	три́жды	thrice
молюⷤ	молю́ же	I pray also
мыⷤ	мы́ же	we also
ниⷤ	ни́же	below
таⷤ	та́же	also
тоⷤ	то́же	the same (nom. sing.)
тогоⷤ	того́ же	the same (gen. sing.)
тоиⷤ	то́йже	the same one, likewise
Combining Ze (U+2DE5, without pokrytie):		
беⷥако́нїѧ	без̾ꙁако́нїѧ	iniquities
воⷥва́хъ	возва́хъ	I cried
воⷥгласъ	во́згласъ	ecphonesis
воⷥсїѧ̑	возсїѧ̑	shined forth
праⷥдникъ	пра́здникъ	feast

Continued on next page

Continued from previous page

Abbreviation	Full Form	Meaning
ро꙼	розво́дъ	expansion[48]
Combining I (U+A675, without pokrytie):		
вели́кꙵ	вели́кїи	great
творѧꙵ	творѧ́й	he creates, creating
ѵ҆пакоꙵ, и҆пакоꙵ	ѵ҆пако́й	Hypacoē (type of hymn)
Combining Ka (U+2DE6, with pokrytie):		
велиⷦ҇	вели́кїи	great
кондаⷦ҇	конда́къ	Kontakion (type of hymn)
поⷦ҇	покло́нъ	bow, prostration
проⷦ҇	проки́менъ	Prokimenon (type of hymn)
вꙿ пѧⷦ҇	въ пѧто́къ	on Friday
вꙿ черⷦ҇	въ четверто́къ	on Thursday
Combining El (U+2DE7, with pokrytie):		
помиⷧ҇	поми́лꙋй	have mercy
Combining Em (U+2DE8, without pokrytie):		
вѣкѡⷨ	вѣкѡ́мъ	[of] ages
на́шыⷨ	на́шымъ	[to] our
посеⷨ	посе́мъ	after this, then
твоиⷨ	твои́мъ	[to] thy
Combining En (U+2DE9, with pokrytie):		
є҆ктеⷩ҇	є҆ктенїѧ̀	litany, ectenē
каноⷩ҇	кано́нъ	Canon (type of hymnography)
іⷩ҇	и҆ ны́нѣ	both now
ныⷩ҇	ны́нѣ	now
пи́саⷩ҇	пи́санъ, пи́саныи	written
поклоⷩ҇	покло́нъ	bow, prostration
Combining O (U+2DEA, with pokrytie):		
прⷪ҇ркъ, прⷪ҇ро́къ	проро́къ	Prophet
сотвⷪ҇рѝ	сотворѝ	create
трⷪ҇ца	тро́ица	Trinity
трⷪ҇ченъ	тро́йченъ	Triadicon (type of hymn)
Combining Pe (U+2DEB, with pokrytie):		
крѣⷫ҇стъ	крѣ́постъ	strength
троⷫ҇	тропа́рь	Troparion (type of hymn)
Combining Er (U+2DEC, with pokrytie):		
втоⷬ҇	вто́рникъ	Tuesday
и҆мⷬ҇къ, і҆мⷬ҇къ	и҆мѧ ре́къ	say name here
на о҆боⷬ҇	на о҆боро́тѣ	see on reverse
чюдотвоⷬ҇	чюдотво́рецъ	wonderworker
Combining Es (U+2DED, with pokrytie):		
а҆пⷭ҇лъ	а҆по́столъ	Apostle

Continued on next page

[48] This term appears in chant books and indicates the expansion of a Znamenny fita.

Continued from previous page

Abbreviation	Full Form	Meaning
бжⷭ҇твенын	боже́ственын	Divine
бжⷭ҇тво̀	божество̀	Divinity
блгⷭ҇ве́нъ	благослове́нъ	blessed
воскрⷭ҇нїе	воскресе́нїе	resurrection
воскрⷭ҇нѝ	воскреснѝ	arise
гдⷭ҇и	го́споди	Lord (masc. voc. sing.)
гдⷭ҇ь	госпо́дь	Lord (masc. nom. sing.)
гдⷭ҇и́нъ	господи́нъ	lord (secular ruler)
гдⷭ҇рь	госꙋда́рь	ruler
гдⷭ҇ьство	госпо́дство	dominion
гⷭ҇жа̀, гⷭ҇пжа̀	госпожа̀	lady
глаⷭ҇	гла́съ	tone
двⷭ҇тво	дѣ́вство	virginity
є҆пⷭ҇кпъ	є҆пи́скопъ	bishop
є҆стⷭ҇тво̀	є҆стество̀	essence
и҆рмоⷭ҇, ї҆рмоⷭ҇	и҆рмо́съ, ї҆рмо́съ	Hirmos (type of hymn)
ї҆єрⷭ҇ли́мскїи	ї҆єросали́мскїи	[of] Jerusalem
ї҆єрⷭ҇ли́мъ	ї҆єросали́мъ	Jerusalem
крⷭ҇тль	крести́тель	Baptist
крⷭ҇тъ	кре́стъ	Cross
млⷭ҇рдїе	милосе́рдїе	loving-kindness
млⷭ҇тивъ	ми́лостивъ	merciful, compassionate
млⷭ҇тынѧ	ми́лостынѧ	charity, alms-giving
млⷭ҇ть	ми́лость	mercy
мцⷭ҇ъ	мѣ́сѧцъ	month
мⷭ҇чный	мѣ́сѧчный	monthly
нбⷭ҇ныи	небе́сныи	heavenly
пречⷭ҇таѧ	пречи́стаѧ	most pure
пречⷭ҇тны́и	пречестны́и	most honorable
прⷭ҇нѡ	при́снѡ	always, ever
прⷭ҇то́лъ	престо́лъ	throne
прⷭ҇тъ	пресвѧ́тъ	most-holy
пѣⷭ҇	пѣ́снь	Ode
ржⷭ҇твенъ	рожде́ственъ, роже́ственъ	[of the] Nativity
ржⷭ҇тво̀	рождество̀, рожество̀	Nativity
ста́рⷭ҇ть	ста́рость	old age
стрⷭ҇ть	стра́сть	passion
трⷭ҇тъ	трис҃тъ, трисвѧ́тъ	thrice-holy
трⷭ҇то́е	трис҃то́е, трисвѧто́е	Trisagion hymn
хрⷭ҇тїѧ́нинъ	христїѧ́нинъ	Christian
хрⷭ҇тїѧ́нъ	христїѧ́нъ	[of] Christians
хрⷭ҇то́въ	христо́въ	[of] Christ
хрⷭ҇то́съ	христо́съ	Christ
црⷭ҇кїи	ца́рьскїи	[of the] king

Continued on next page

Continued from previous page

Abbreviation	Full Form	Meaning
црⷭ҇тво	ца́рьство	kingdom
чⷭ҇тенъ	че́стенъ	honorable (short adj.)
чⷭ҇тны́и	честны́и	honorable (long adj.)
чⷭ҇тыи	чи́стыи	pure (long adj.)
чⷭ҇тъ	чи́стъ	pure (short adj.)
чⷭ҇ть	че́сть	honor
Combining Es-Te (U+2DED U+2DEE,[49] always without pokrytie):		
лиⷭⷮ	ли́стъ	folio
поⷭⷮ	по́стъ	fast
ѿпоⷭⷮу	ѿпꙋ́стъ	dismissal
Combining Te (U+2DEE, always without pokrytie):		
живоⷮ	живо́тъ	life
свѣⷮ, с҃вѣⷮ	свѣти́ленъ	Photagogicon, Exapostilarion (type of hymns)
сꙋбѡⷮ	сꙋбѡ́та	Saturday, Sabbath
оу҆слы́шиⷮ	оу҆слы́шитъ	[he will] hear
Combining Uk (U+2DF9, always without pokrytie):		
є҆ди́нⷹ	є҆ди́нꙋ	one
Combining Uk (U+2DF9, always without pokrytie):		
иⷯ	и҆́хъ	them
н҃бсѣⷯ	небесѣ́хъ	[in the] heavens
своиⷯ	свои́хъ	[his] own
стиⷯ, сⷯ, Сⷯ	сти́хъ	verse
сⷯтры	стихѣ́ры	stichera
с҃тыⷯ	свѧты́хъ	[of the] saints
Combining Tse (U+2DF0, always with pokrytie):		
конеⷰ҇	коне́цъ	the end
мѣ́сѧⷰ҇	мѣ́сѧцъ	month
Combining Che (U+2DF1, always with pokrytie):		
веⷱ҇	ве́черъ	evening
заⷱ҇	зача́ло	pericope
о҆троⷱ҇	о҆троча̀	child
прѡⷱ҇	прѡ́чаѧ	others, the rest
Combining Sha (U+2DF2, always with pokrytie):		
болⷲ҇	болшо́й	greater
Combining Shcha (U+2DF3, always with pokrytie):		
аⷳ҇	а҆́ще	if
бл҃говѣⷳ҇	благовѣ́щенїе	annunciation
о҆бⷳ҇	о҆́бщыи	general, common
Combining Yat (U+2DFA, always without pokrytie):		
на є҆динⷺ	на є҆динѣ̀	alone
Combining Yu (U+2DFB, always without pokrytie):		

Continued on next page

[49] This double titlo was originally encoded at the codepoint U+2DF5, however as of version 8.0 of the Unicode Standard, it is properly handled as a digraph.

Continued from previous page

Abbreviation	Full Form	Meaning
на́шⷻ	на́шю	our
Combining Iotified A (U+2DFC, always without pokrytie):		
моⷼ	моѧ̀	my
Combining Little Yus (U+2DFD, always without pokrytie):		
ноⷽ҇	ноѧ́бръ	November
Combining Fita (U+2DF4, always without pokrytie):		
каⷴ	каѳи́сма	cathisma (section of Psalter)
Words written with a common titlo (U+0483):		
а҆́нг҃лъ or а҆́гг҃лъ	а҆́нгелъ	Angel
а҆́нг҃льскїи	а҆́нгельскїи	[of the] Archangel
а҆рха́нг҃лъ	а҆рха́нгелъ	Archangel
б҃гъ	Бо́гъ	God (nom.)
б҃же	Бо́же	God (voc.)
бл҃гослове́нъ	благослове́нъ	blessed
бл҃гъ	бла́гъ	good
бл҃же́нъ	блаже́нъ	blessed
г҃и or гⷭ҇ди	го́споди	Lord (voc.)
г҃ли	глаго́ли	say (imper.)
д҃хъ	дꙋ́хъ	Spirit
дв҃дъ	давы́дъ	David
д҃во	дѣ́во	Virgin
д҃вца or д҃ви́ца	дѣви́ца	maiden
д҃нь	де́нь	day
дн҃сь	дне́сь	today
д҃ша	дꙋша̀	soul
і҆н҃ль	і҆зра́иль	Israel
і҆с҃ъ or і҆с҃	І҆сꙋ́съ	Jesus
кр҃ще́нїе	креще́нїе	baptism
мл҃тва	моли́тва	prayer
м҃рі́ѧ	марі́ѧ	Mary
м҃ти	ма́ти	mother (voc.)
мт҃ръ	ма́теръ	mother (nom.)
мч҃нкъ or мч҃нникъ	мꙋ́ченникъ	martyr (masc.)
мч҃ца or мч҃нница	мꙋ́ченница	martyr (fem.)
н҃бо	не́бо	heaven (nom. sing.)
нб҃съ	небе́съ	heaven (gen. pl.)
н҃нѣ	ны́нѣ	now
н҃шъ	на́шъ	our (nom. sing.)
о҆ц҃а	о҆тца̀	Father (gen. sing.)
о҆ц҃ъ	о҆те́цъ	Father (nom. sing.)
о҆ч҃ь	о҆те́чь	[of the] Father
сл҃ва	сла́ва	glory
с҃лнце	со́лнце	sun
см҃рть	сме́рть	death

Continued on next page

Continued from previous page

Abbreviation	Full Form	Meaning
с҃нъ	сы́нъ	Son (nom. sing.)
сп҃съ	спа́съ	Savior
ст҃ль	свѧти́тель	hierarch
с҃ты́и	свѧты́и	holy, saint
с҃щє́нникъ	свѧщє́нникъ	priest
трис҃то́е or трⷭ҇то́е	трисвѧто́е	Trisagion (type of hymn)
трис҃тъ or трⷭ҇тъ	трисвѧ́тъ	thrice-holy
ѹ҆чн҃къ	ѹ҆чени́къ	disciple
ѹ҆чт҃ль	ѹ҆чи́тель	teacher
х҃съ or х҃с	Христо́съ	Christ
ц҃рковь	це́рковь	church
цр҃ца	цари́ца	queen
ц҃рь	ца́рь	king (nom. sing.)
ц҃рю	царю̀	king (voc. sing.)
чл҃къ	человѣ́къ	man, human (nom. sing.)
чл҃чь	человѣ́чь	[of the] man
ѱл҃о́мщикъ	ѱало́мщикъ	psalmist, chanter
ѱл҃о́мъ	ѱало́мъ	Psalm (nom. sing.)
ѱламо́въ	ѱаламо́въ	Psalms (gen. pl.)
ѱл҃мы̀	ѱалмы̀	Psalms (nom. pl.)

References

Adobe Systems (2007). *Glyph (Adobe Developer Connection)*. Adobe Systems. http://www.adobe.com/devnet/opentype/archives/glyph.html.

Andreev, A., H. Miklas, and Y. Shardt (2014). Proposal to Encode Additional Glagolitic Characters in Unicode. ISO/IEC JTC1/SC2/WG2 N4608; http://www.ponomar.net/files/glagolitic.pdf.

Andreev, A., Y. Shardt, and N. Simmons (2012). Proposal to Encode Mediæval East-Slavic Music Notation in Unicode. ISO/IEC JTC1/SC2/WG2 N4206, http://www.ponomar.net/files/kievan.pdf.

Andreev, A., Y. Shardt, and N. Simmons (2013). Proposal to Encode Combining Half Marks Used for Cyrillic Supralineation in Unicode. ISO/IEC JTC1/SC2/WG2 N4475; http://www.ponomar.net/files/halfmarks.pdf.

Andreev, A., Y. Shardt, and N. Simmons (2014). Proposal to Encode Additional Cyrillic Characters Used in Early Church Slavonic Printed Books. ISO/IEC JTC1/SC2/WG2 N4607; http://www.ponomar.net/files/variants_final2.pdf.

Andreev, A., Y. Shardt, and N. Simmons (2015a). Private Use Area (PUA) Allocation Policy. Draft Technical Manual, http://www.ponomar.net/files/pua_policy.pdf.

Andreev, A., Y. Shardt, and N. Simmons (2015b). Proposal to Encode some Additional Symbols used in Church Slavonic Text. L2/15-173 (Revision 2), http://www.unicode.org/L2/L2015/15173r2-add-typicon.pdf.

Andreev, A. and N. Simmons (2015). Proposal to Standardize the Encoding of Palæoslavic Musical Notations in Unicode. Working paper available at http://www.ponomar.net/cu_support/music.html.

Baranov, P. A. (1914). *Арифметика Магницкого: точное воспроизведение подлинника.* Moscow.

Bulatova, R. V. (1973). Надстрочные знаки в южнославянских рукописях XI-XIV вв. In *Методическое пособие по описанию славянорусских рукописей для Сводного каталога рукописей, хранящихся в СССР*. Moscow.

Cleminson, R. and M. Everson (2009). Proposal to Encode two Cyrillic characters in the BMP of the UCS. Working Group Document: ISO/IEC JTC1/SC2/WG2 N3563R.

Davydenkova, M., N. Kaluzhnina, O. Striyevskaya, and N. Mazurina (2013). Словарь речений из богослужебных книг прот. А. И. Невоструева. *Vestnik PSTGU III. Filologiya* (2 (32)), 114–138.

D′yachenko, G. (1899). *Полный церковно-славянский словарь.* Moscow.

Gamanovich, A. and J. Shaw (2001). *Grammar of the Church Slavonic Language.* Jordanville, NY: Holy Trinity Monastery Press. (Archb. Alypy (Gamanovich); trans. into English by J. (now Bishop Jerome) Shaw).

Gilterbrandt, P. (1898). *Справочный и объяснительный словарь к Псалтири.* St. Petersburg: Sinodal′naya Tipografiya.

Ginkulov, Y. (1840). *Начертание правил валахо-молдавской грамматики.* St. Petersburg.

Golyshenko, V. S. (1987). *Мягкость согласных в языке восточных славян XI-XII вв.* Moscow: Nauka.

GOST 7.79 (2000). *Правила транслитерации кирилловского письма латинским алфавитом.* Межгосударственный совет по стандартизации, метрологии и сертификации.

Hosken, M., B. Hallissy, W. Cleveland, S. Correll, and A. Ward (2011). *Graphite Description Language version 2.004.* SIL International.

Irish NB and German NB (2011). Revised Proposal to Enable the use of Combining Triple Diacritics in Plain Text. ISO/IEC JTC1/SC2/WG2 N4078.

Ivanova, T. A. (1997). *Старославянский язык.* Moscow: Vysshaya Shkola.

Izotov, A. I. (2007). *Старославянский и церковнославянский языки.* Moscow: Filomatis.

Jagić, V. (1879). *Quattuor Evangeliorum codex glagoliticus olim Zographensis nunc Petropolitanus.* Berlin: Apud Weidmannos.

Karsky, E. F. (1979). *Славянская Кирилловская Палеография.* Moscow: Nauka Press.

Kempgen, S. (2008). Unicode 2C1A – Glagolitic "Pe": Fact or Fiction? *Scripta & e-Scripta 6*, pp. 65–82.

Kostić, Z. (2009). Exaplanation of the Proposal for Standardization of the Old Slavonic Cyrillic Script and its Registration in Unicode. In G. Jovanović, J. Grković-Major, Z. Kostić, and V. Savić (Eds.), *Standardization of the Old Church Slavonic Cyrillic Script and Its Registration in Unicode.* Serbian Academy of Sciences and Arts.

Kostić, Z., V. Savić, et al. (2009). Standard of the Old Slavonic Cyrillic Script. In G. Jovanović, J. Grković-Major, Z. Kostić, and V. Savić (Eds.), *Standardization of the Old Church Slavonic Cyrillic Script and Its Registration in Unicode.* Serbian Academy of Sciences and Arts.

Kozlovsky, M. (1885). Исследование о языке Остромирова евангелия. In *Исследования по русскому языку*, Volume 1. St. Petersburg.

Koz′min, K. (1903). *Грамматика Церковно-славянского языка нового периода* (13 ed.). Moscow: Sinodal′naya Tipografiya.

Kravetsky, A. G. and A. A. Pletneva (2001). *История церковнославянского языка в России (конец XIX-XX в.).* Moscow: Yazyki Russkoy Kul′tury.

Krylov, G. (2009). *Книжная справа XVII века: богослужебные Минеи.* Moscow: Indrik.

Microsoft Corporation (2002). *Developing OpenType Fonts for Standard Scripts.* Microsoft Corporation. http://www.microsoft.com/typography/OpenTypeDev/standard/intro.htm.

Microsoft Corporation (2010). *Recommendations for OpenType Fonts.* Microsoft Corporation. http://www.microsoft.com/typography/otspec/recom.htm.

Mironova, T. (2008). *Церковнославянский язык.* Moscow.

Nemirovsky, E. L. (2003). *История славянского кирилловского книгопечатания XV- начала XVII века.* Moscow: Nauka.

Nicholas, N. (2006). *Unicode Technical Note 20: Byzantine Musical Notation.* Unicode Consortium.

Pletnyeva, A. A., A. G. Kravetsky, and V. M. Zhivov (1996). *Церковно-славянский язык: для общеобразовательных учебных заведений.* Moscow: Prosveshcheniye.

Russian Imperial Academy of Sciences (1847). *Словарь Церковно-Славянского и Русского Языка.* St. Petersburg: Tipografiya Imperatorskoy Akademii Nauk.

Sedakova, A. O. (2005). *Церковнославянско-русские паронимы: материалы к словарю.* Moscow: Greko-latinskiy kabinet Yu. A. Shichalina.

Shchepkin, V. N. (1999). *Русская Палеография.* Moscow: Apekt Press.

Sove, B. I. (1970). Проблема исправления богослужебных книг в России в XIX–XX веках. *Bogoslovskiye Trudy 5*, pp. 30–31.

Syrnikov, N. (1910). *Ключ к церковному уставу.* Moscow: Synodal Press.

Unicode Consortium (2015a). *The Unicode Standard Version 8.0 – Core Specification.* Mountain View, CA: Unicode Consortium.

Unicode Consortium (2015b). Unicode Technical Standard #35: Unicode Locale Data Markup Language (LDML). version 27; http://www.unicode.org/reports/tr35/.

Uspensky, B. A. (1987). *История русского литературного языка (XI-XVII вв.).* München: Verlag Otto Sagner.

Vorob′yeva, A. G. (2008). *Учебник церковнославянского языка.* Moscow: St. Tikhon’s Orthodox Humanitarian University.

Vostokov, A. K. (1863). *Грамматика церковно-словенскаго языка, изложенная по древнѣйшим онаго письменным памятникам.* St. Petersburg: Imperial Academy of Sciences.

Yelkina, N. M. (1960). *Старославянский язык.* Moscow, Russia: Ministry of Education of the RSFSR.

Zagrebin, V. N. (2006). *Исследование памятников южнославянской и древнерусской письменности.* Moscow: Al′yans-Arkheo.