



ELSEVIER

Computer Standards & Interfaces 18 (1996) 201-252

COMPUTER STANDARDS
& INTERFACES

Standardizing characters, glyphs, and SGML entities for encoding early Cyrillic writing¹

David J. Birnbaum*

*University of Pittsburgh, Dept. of Slavic Languages and Literatures,
1417 Cathedral of Learning, Pittsburgh, PA 15260, USA*

Abstract

The present study discusses the differences among CHARACTERS, GLYPHS, and SGML ENTITIES (§2), evaluates how these distinctions might be applied to electronic text projects involving early Cyrillic materials (§3), and proposes basic inventories of the characters, glyphs, and entities needed for computer processing of early Cyrillic written materials (§4). None of the issues examined in this study is unique to early Cyrillic writing, and the principles elucidated here can be generalized to problems affecting the standardized encoding of other complex writing systems.

Keywords: Characters; Glyphs; SGML entities; Encoding; Cyrillic writing

1. Introduction and overview

1.1. Problems with the technical adequacy of existing systems

Slavic philologists work extensively with manuscript artifacts. Like many other scholars, they are exploring the use of computers as an aid in scholarly analysis, but this use has been hampered, and even harmed, by the relatively poor representation

of the objects of study that can be achieved in digital electronic form, where the only reasonably effective representation of Slavic manuscripts to date has been as digital images. While these images can represent much of the physical richness of original manuscripts, the ability to process the textual content of these electronic copies has been limited. Conversion of the images to a symbolic electronic form, or coded character representation, would allow for more extensive and sophisticated processing,

* Email: djbpitt+@pitt.edu <http://clover.slavic.pitt.edu/~djbpitt>.

¹ Because upper-case and lower-case letters usually do not differ in shape in early Cyrillic, my discussion of the properties of early Cyrillic letters normally applies equivalently to both upper and lower case, even if illustrated with only one or the other. Exceptions are disclaimed explicitly.

Most references to the names of Cyrillic letters in the text of this article follow normal Slavistic scholarly practice, as does the transliteration of Slavic words. The transliteration that has been used in ISO and Unicode character names and AFII glyph

descriptions is less systematic and less consistent, and is retained here only where these names themselves are a particular object of study (such as in the inventory listings at the end of the article). Because standards specialists may be unfamiliar with traditional Slavistic practice and Slavists may be unfamiliar with traditional ISO naming strategies, I have attempted to minimize confusion by identifying characters and glyphs in the text of this article wherever possible by both a UCS byte value (or, for glyphs, AFII reference and glyph numbers) and a representative rendering.

but most available representations are poor in features, and tend to neutralize textual differences in the original document that may be needed for scholarly analysis. The fundamental problem is that some units in an original source (e.g., the letterforms "ѡ" and "Ѣ") should be considered the same for one type of processing and different for another, and the most useful encoding system should be capable of representing both the identity and the difference simultaneously.

Recent developments in computer standards have provided new possibilities for representing the textual content of digitized manuscript materials. These developments have occurred at three levels:

- (1) The international standards community has adopted a new global, multilingual character encoding standard that subsumes and expands many existing international, national, popular, and industry standards. This has been accompanied by a compartmentalization of issues related to the elements of written text and the symbolic representation of those elements. This development offers new opportunities for those concerned with the encoding and rendering of textual data, particularly in areas which were not served well by earlier standards, such as Slavic philology. The new standard is called UNICODE, and has been registered with the International Organization for Standardization (ISO) as a subset of ISO/IEC 10646-1:1993.
- (2) The international standards community has adopted a standard for the representation of documents that offers new techniques and rigor for encoding both the structure and content of documents. This standard is called STANDARD GENERALIZED MARKUP LANGUAGE (SGML), and it has been registered with the International Organization for Standardization as ISO 8879: 1986.
- (3) The international community has been working for many years to develop a superordinate standard, based on SGML, that will allow for the encoding of scholarly documents in the humanities. This enterprise is called the Text Encoding Initiative (TEI), and the standard in question is described in the *Guidelines for Electronic Text Encoding and Interchange*, also known as P3.

The present article reviews these developments (§2), shows how they contribute to a solution for the symbolic encoding of early Cyrillic manuscripts (§3), and proposes for consideration and discussion an architecture and set of inventories for encoding early Cyrillic manuscript materials (§4). Every effort has been expended in developing this approach to make appropriate use of the standards listed above, to make modifications to standards only when they are required, to account for the needs of the various research communities involved, and to keep the solution simple and implementable.

1.2. Problems with the standardization of existing systems

STANDARDIZATION IN HUMANITIES COMPUTING, as elsewhere, is intended to support the interchange of information among processes and among users, and thus to facilitate cooperative research. Unfortunately, the de facto lack of standardized ways of encoding and representing Cyrillic data on computers has long been an embarrassing example of academic and computational chaos and parochialism. The coexistence and widespread use of mutually incompatible international character set standards (e.g., ISO 8859-5: 1988), national standards (e.g., GOST 19768-74 [= "old KOI-8"], GOST 19768-87 [= "new KOI-8"]), popular standards (e.g., Brjabrin's *basic* [*osnovnoj variant, OV*] and *alternative* [*alternativnyj variant, AV*] encodings), and industry standards (e.g., *Apple Cyrillic* and *Apple Extended Cyrillic*; CP 866 [used in Russified MS-DOS] and CP 1251 [used in Microsoft Cyrillic Windows 3.1]; CECP 037 and 500 [both variants of DKOI-8, or Cyrillicized EBCDIC]) has created a situation where Slavists who wish to render (display or print) electronic files developed by others often must exchange fonts before they are able to do so (the TEXT RENDERING PROBLEM). No less importantly, routines written to process one type of encoded Cyrillic data cannot be applied to other data without prior filtering (the TEXT PROCESSING PROBLEM)².

² This division is not absolute. Text rendering and text processing are necessarily interconnected, since one normally wishes to render at least the final results of processing operations, and processing normally is intended to lead eventually to something that will be rendered.

An analogous situation in the English-speaking world would long ago have been proclaimed intolerable, but there is still no single Cyrillic character set that can be compared to ASCII as virtually a universal hardware-independent, operating-system-independent, application-independent, and font-independent resource for information encoding, processing, and interchange³.

Scholars who need to encode *early* Cyrillic texts face even a worse problem: there are *no* official standards for encoding premodern materials⁴. As a result, many Slavic philologists who work with computers design their own idiosyncratic encoding schemes, as if in isolation from any scholarly community, and it is rare to find two systems produced by different parties that are compatible. Macintosh and MS-DOS and UNIX users can all create and share English-language ASCII text files without even having to think about character set or font compatibility, but Slavists who work with mediæval manuscript materials are almost certain to be unable to do so. One goal of this paper is to provide part of the solution to this problem.

³ See Clews 64-77 for a general survey of Cyrillic character set standards through approximately 1987; for more complete and up-to-date information, see also Kornai.

Incompatibilities remain in English character encoding as well, particularly due to differences between ASCII and EBCDIC. But at least there are only two widely-used and well-established English character encoding standards, and industry has managed to develop strategies to help translate between them in ways that do not require the active intervention of possibly naive end users. This is not the case with Cyrillic encoding, which has a much greater number of competing encoding schemes and almost no well-established facilities that might resolve the differences among them for a computationally unsophisticated end user.

⁴ I use the term *EARLY CYRILLIC* to identify Cyrillic writing that requires letters that were once part of certain Slavic or Romanian Cyrillic writing systems, but are not currently in use in standard Russian, Ukrainian, Belarusian, Bulgarian, Macedonian, or Serbocroatian typography. Some of these items are in current use in modern Church Slavonic of various recensions.

ISO DIS 6861.2 contained an early Cyrillic inventory, but all Cyrillic was deleted from the draft before final balloting, and the approved version of that standard addresses only Glagolitic. As a result, the Cyrillic information in this DIS has no official status.

2. Developments in and theory of symbolic electronic representation

2.1. Principles and procedures

2.1.1. Unicode

The early Cyrillic character inventory presented here is based on Unicode design principles (*Unicode* 7-25). References in this article to UCS (Universal Character Set) byte values (indicated by U+xxxx, where xxxx represents a four-digit hexadecimal value) conform to both the Unicode standard (as documented in *Unicode*) and ISO/IEC 10646-1: 1993⁵.

Reasonable persons have long disagreed over many of the issues discussed below (plain versus fancy text, variant spellings, character/glyph distinction), but the publication of *Unicode* and the ratification of ISO/IEC 10646-1: 1993 have provided — albeit with compromises — a robust standard framework for encoding multilingual data. Since a principal purpose of the present report is to enable Slavic mediævalists to join the community of those who observe standards, the character inventory proposed below is based on an acceptance of all Unicode design principles, and, where necessary, an adaptation of these principles to the peculiarities of early Cyrillic writing.

2.1.2. PLAIN and FANCY TEXT

Unicode distinguishes *PLAIN TEXT*, which contains nothing but character codes, from *FANCY (or RICH) TEXT*, which incorporates additional information (usually structural or presentational). Unicode's productive definition of characterhood assumes that graphic, orthographic, or palæographic distinctions that do not distinguish sound or meaning should not be representable in the character set, i.e., should

⁵ When the DIS (Draft International Standard) for ISO 10646 was voted down by the relevant ISO member bodies, SC2 and the Unicode Consortium undertook to merge the Unicode standard and ISO 10646. This merger was the basis for a second DIS, which eventually became ISO/IEC 10646-1: 1993. One aspect of this merger involved the replacement of original Unicode names with ISO 10646 names in the small number of instances where the two sets had differed. *Unicode* reflects original Unicode names, the present article identifies UCS characters by their post-merger names.

not be representable in plain text. This means that UCS is deliberately not intended to be able to represent many features of mediæval Slavic manuscripts that scholars might nonetheless wish to use in the computer-assisted analysis of these manuscripts. Such features must instead be represented in fancy text.

2.1.3. Levels of representation

The system described below is based on a complex model that decomposes text into different levels of representation. These levels distinguish CHARACTERS from GLYPHS, and also identify two types of glyphic variation, involving changes of GLYPH IDENTIFIERS and changes of FONT IDENTIFIERS. SGML ENTITIES are employed to represent correspondences of characters and glyphs, since neither of the latter inventories is fully predictable from the other.

Parts of this general approach may be familiar to the standards community, but it remains virtually unknown among Slavist end users, who tend to regard and treat electronic text as a single, unlayered encoding. This unlayered strategy has the apparent advantage of design simplicity, in that a single inventory of units is employed to record everything scholars need to process or render. Furthermore, the use of a single layer means that all electronic text can be regarded as plain text, and can be processed by applications that recognize only plain text⁶.

In many cases a one-to-one correspondence between the components of two electronic text layers is not inappropriate, which may lead a Slavist to conclude that a more complex system offers no advantages that would outweigh the convenience of the simple, single-layer model described above. Unfortunately, this type of multistratal coincidence does not always obtain, and the present article attempts to identify situations where the failure to distinguish structural levels in text encoding may complicate, or even prohibit, certain types of

processing. A user who simply wants to type and print plain text in modern Russian may be served adequately by the simpler model, but a humanities scholar who needs to perform a wider array of operations on early Cyrillic electronic texts will find this simpler model inadequate.

2.1.4. Principles and Consistency

A set of characters or glyphs ideally should be developed according to consistent principles: the architect should define what distinguishes characters from glyphs, and what distinguishes one character from another or one glyph from another, and the sets should contain every element that conforms to the appropriate definition and no element that does not. These definitions should be descriptive, so that they can be applied to new data that may later need to be incorporated into the inventories.

This rigor and consistency has not been the case with the standard character and glyph inventories discussed below for the following reasons:

- (1) principles based on inappropriate criteria;
- (2) conflicting principles;
- (3) insufficiently precise definitions, so that conflicting solutions may simultaneously conform to stated principles;
- (4) data that involve subjective evaluation, where more than one analysis may be equally valid;
- (5) peculiarities of new data, which may involve issues not present in the data used originally to develop the principles;
- (6) a prohibition against removing items from a set, even when these are later shown to be erroneous or to violate stated principles.

Some of the preceding sources of inconsistency are inevitable, and their enumeration does not necessarily imply ignorance or negligence on the part of architects, standards organizations, registration authorities, or other responsible parties. In many cases, the decisions leading to the problems noted above have their own justification, and it may have been impossible to resolve one set of problems without creating another. Nevertheless, one consequence of these limitations is that although I have attempted to define and apply consistent

⁶ Files encoded in an eight-bit plain-text representation may be amenable to processing with plain-text applications only on the platform on which they were originally encoded, since different platforms reserve different non-ASCII characters for internal control functions.

procedures in developing the inventories proposed in this report, neither the character nor the glyph inventory is, ultimately, fully consistent. I have tried to document these inconsistencies, and to suggest guidelines that may help regulate further development. Where inconsistent or subjective decisions are inevitable, a system should acknowledge and document them explicitly.

2.2. CHARACTERS, GLYPHS and ENTITIES

2.2.1. What is a TEXT ELEMENT?

Unicode employs the term TEXT ELEMENT to identify what users of a language or writing system perceive as the "fundamental units" of written text in that language (7)⁷. Text elements are the units subjected to TEXT PROCESSES, which are operations such as sorting and rendering. Because different text processes are actually performed on different sets of significant units, the inventory of distinctive text elements may vary not only with respect to language or writing system, but also with respect to text process within a single language or system. For example, upper- and lower-case variants of the same letter normally function as the same text element for alphabetizing, but as different text elements for rendering. CHARACTERS, GLYPHS, and SGML ENTITIES, discussed immediately below, provide different ways of encoding these different inventories of text elements.

2.2.2. What is a CHARACTER?

ISO/IEC 10646-1: 1993 defines a character as "a member of a set of elements used for the organization, control, and representation of data". This definition does not restrict the type of data (which N 915 glosses as "units of information") that can be represented by characters, even though what is purely presentational for some processes may be informational for others. A more precise definition of a character is required if character sets are to be developed according to consistent and objective principles.

2.2.2.1. The productive definition of character

N 915 attempts to satisfy this requirement by specifying that "a character conveys distinctions in meaning or sounds"⁸. If we interpret "meaning" as lexical or grammatical meaning, this definition implies that two different text elements that can be interchanged without changing the sound or meaning of a word in which they occur should be considered representations of the same character, and should be encoded identically on the character level. This definition does not require the existence of minimal pairs; if replacing one text element with another would not convert one existing word form into a different existing word form, but would nonetheless introduce a change in sound or meaning (perhaps by neutralizing an optional sound distinction or by creating an impossible or unnatural sequence), the two text elements should not be considered representative of the same character. If it is nonetheless important to encode the difference between such noncharacters for processes not involved in discriminating sound or meaning, that difference must be encoded other than at the character level.

2.2.2.2. The round trip rule

The preceding constitutes the PRODUCTIVE definition of character, but the UCS inventory also includes many items that do not satisfy this definition because they do not convey distinctions in meaning or sounds within a script. These exceptions to the productive definition of character are admitted because Unicode sought to preserve a reversible mapping between Unicode texts and texts encoded in any of the earlier character sets that served as input to the design of Unicode. "Many characters have been included solely because they are part of an existing standard in widespread use, despite the fact that they violate the general principles of the Unicode standard in some instances." (*Unicode 3*) "Where variant forms are given separate codes within one included standard, they are also kept separate within the Unicode standard. This guarantees that there

⁷ The Unicode use of the term TEXT ELEMENT has nothing in common with the SGML use of the term ELEMENT.

⁸ This is purely a structural definition, in the sense that characters convey not specific meaning or sound, but distinctions in meaning or sound.

will always be a mapping between the Unicode standard and included national standards." (*Unicode 11*)⁹ N 915 refers to this principle as the ROUND TRIP RULE, which it defines as follows¹⁰:

If a form is included as a character in any of the character sets from which ISO/IEC 10646 is derived, then that form shall be included as a character in ISO/IEC 10646 such that distinctions among characters in the source character sets are maintained as distinctions in ISO/IEC 10646.

Two clear examples of this policy are the separate encoding of CJK Compatibility Ideographs (U+FA2D through U+FA2D) and Fullwidth Forms (U+FF00 through U+FF5E). In the early Cyrillic portion of the UCS inventory, this policy is responsible for the inclusion of several items that do not satisfy the productive Unicode definition of character. Specifically, U+047D (ѿ), U+047F (ѿ), and U+0477 (ѿ) are not unitary characters from a structural perspective, and would not have been encoded as such in Unicode were it not for their presence in ISO DIS 6861.2¹¹.

2.2.2.3. Scripts and writing systems and the limitations of script-based character sets

Certain early Cyrillic text elements may distinguish sound or meaning in some early Slavic writing systems but not in others. For example, Old Church Slavonic U+0467 (Ѧ) and U+044F (Ѧ) distinguish both sound and meaning, while what appear to be the same text elements in early East Slavic writing represent a single sound and do not distinguish meaning. That Old Church Slavonic and early East Slavic nonetheless must be encoded with the same characters is a consequence of the Unicode

decision to encode scripts (e.g., Latin, Greek, Cyrillic, etc.), rather than languages or orthographies. This decision affords many advantages, but it also introduces complications both for formulating a meaningful definition of character and for encoding early Cyrillic materials.

It is nearly always impossible to determine whether two text elements distinguish sound or meaning on the level of script because different languages or orthographic systems that use the same script may assign different sounds or meanings to what appear to be the same graphic items. This is comparable to the linguistic analysis of phones (physical language sounds) into phonemes (distinctive sounds), where distinctiveness can be determined only in the context of a particular linguistic system. Just as it is not meaningful to ask on a language-independent level whether [t] and [t^h] are independent phonemes or allophones of a single phoneme, it is not meaningful to ask on an orthography-independent level whether U+0467 (Ѧ) and U+044F (Ѧ) are independent characters or variants of a single character. Under the productive Unicode definition of character, these items would be two characters in Old Church Slavonic and one in early East Slavic.

There are at least five approaches one can take to dealing with the existence of a script-based character set in an environment where the basic definition of characterhood supports an orthography-based character set:

- (1) Adopt the existing UCS inventory without modification;
- (2) Modify the existing UCS inventory so that only items that always distinguish sound or meaning are encoded as independent characters;
- (3) Modify the existing UCS inventory so that items that distinguish sound or meaning in any language or orthography written in the script are treated as independent characters within the script;
- (4) Optimize the existing UCS inventory to support lexical, textual, and grammatical analysis; or
- (5) Modify the existing UCS inventory by adding as new characters those text elements whose distinctive graphic features already distinguish other independent characters in the inventory.

⁹ What *Unicode* apparently means is that there should always be a *one-to-one* mapping between members of the two standards. It would have been possible to establish mappings between the two even in the absence of this principle, although in that case a single character in an earlier character set might have been mapped to a sequence of two characters in Unicode, and potential ambiguities might have had to be resolved arbitrarily, thus rendering the mapping irreversible.

¹⁰ It is also sometimes called the "source set rule".

¹¹ Ironically, as is noted in § 1.2 (fn), above, the entire Cyrillic inventory was stricken from this DIS before approval, which removes the original justification for including these ersatz characters in Unicode.

The present paper adopts a combination of the fourth and fifth solutions, for reasons discussed below.

Option 1 (adopting the current UCS character set without modification) is rejected because it dismisses without consideration the opportunity to improve the existing inventory. The current inventory does not follow entirely coherent principles, and it can be improved through reconsideration from both philological and informatic perspectives.

Option 2 (modifying the existing UCS character set so that only items that always distinguish sound or meaning are encoded as independent characters) is rejected as unattainable and undesirable for two reasons. First, UCS already includes many characters that do not always distinguish sound or meaning, and the need to support legacy data makes it impossible to remove items from an existing ISO inventory¹². Second, and more significant, items that may not *always* distinguish sound or meaning may nonetheless reflect such distinctions *very frequently*. If the purpose of the character level is to support what philologists will consider useful processing (cf. § 3.1), Option 2 would maximize conflation even where it is appropriate for only a minority of manuscripts, thus creating a minimal character set that would require greater dependence on glyph distinctions as bearers of what is normally considered character-level information. In other words, Option 2 would result in a character set that would serve no definable purpose.

Option 3 (modifying the existing UCS character set so that two items are independent characters in a script if they are independent characters in any language or orthography written in that script) is rejected as undesirable. Certain distinctions in text elements are capable of distinguishing sound and meaning in a very small number of early Cyrillic documents, but such distinctions may be relatively uncommon, or even numerically negligible. Furthermore, many such distinctions (such as different *e*-type or *o*-type letters or front nasal vowels,

discussed in §§ 3.4.4.1-3, below) reflect differences in sound that often do not engender different meanings, and that frequently are not reflected consistently even within individual manuscripts. Implementing this option would lay the heaviest processing burden on the majority of manuscripts, where these oppositions do not discriminate sound or meaning, while yielding little practical benefit for most processing. What Options 2 and 3 have in common is that they lead to a character level that is not optimized for any type of processing that Slavic philologists would be likely to perform.

Options 4 and 5 are most appropriately considered together. Option 4 (modifying the existing UCS character set so that it is optimized for lexical, textual, or grammatical analysis) is appealing, since this type of analysis is based on differences in meaning, which means that the purpose of the character set and the definition of characterhood would coincide under this strategy. And the existing UCS character set is already fairly well optimized for most lexical, textual, or grammatical analysis, and it coincides largely with the inventories traditionally used in normalized transcriptions or reconstructions of early Cyrillic manuscripts, as well as in handbooks, dictionaries, and other linguistic references.

Option 5 can be considered a strategy for increasing the internal consistency of Option 4. Certain linguistic features, such as jotation and palatal pronunciation, may be reflected through specific graphic features, such as a jotation bar (contrast U+0430 "а" and U+044F "ѧ"; cf. §§ 3.4.3.1-2) or a palatalization hook (contrast U+043B "ѧ" and U+0459 "ѧ̆"; cf. § 4.1.1), respectively. These examples represent a combination of consistent graphic form with consistent semantics. Because some text elements that include these graphic features are already part of UCS, it seems most consistent to incorporate as characters all text elements that include these graphic features¹³.

Perhaps the most significant burden of a script-level character set is not the extra processing for an individual document, but the general corruption

¹² The presence of unremovable non-characters in the inventory is a problem under any solution, but it is particularly objectionable where the application of the productive definition of characterhood across the entire script is taken as the main organizing principle underlying the inventory.

¹³ Some of these new characters are extremely rare, but where they do occur, Slavists would most often want to treat them identically to other characters that reflect the same graphic features.

of the integrity of the character level throughout the system. Early Cyrillic writing is so varied that almost any early Cyrillic document encoded according to a script-level definition of character very likely will reflect some distinctions in text elements that are not correlated in that document with differences in sound or meaning. In other words, not only do individual pseudocharacter distinctions on the level of an individual document complicate processing for that document, but early Cyrillic writing is so permeated with such distinctions as to compromise on a system-wide scale many of the advantages that a well-defined character level might otherwise provide.

The inventories proposed below are intended to be adequate for encoding almost any early Cyrillic manuscript, are optimized for encoding and processing the majority of early Cyrillic manuscripts, and can be extended should they later prove technically inadequate for specific unusual problems. The character inventory proposed below includes those items needed for standard lexical or textological work with early Slavic manuscripts, plus a small number of marginal additions, the justification for which is given in § 3.4, below¹⁴.

2.2.3. What is a GLYPH?

ISO/IEC 9541-1: 1991 defines a glyph as "a recognizable abstract graphic symbol which is independent of any specific design". As N 915 points out, "the degree of abstraction is not defined; nor are criteria defined that would allow determining whether two potential images (forms) are instances of one

abstract glyph, or are to be considered two distinct glyphs, each having an independent image"¹⁵.

ISO/IEC 10036: 1993 standardizes the registration of glyph identifiers, so that glyph information can be exchanged among applications, and the Association for Font Information Interchange (AFII) serves as the registration authority for this information. "Font-specific design information may vary from one font resource to another", (N 915), and analysis of *AFII-REGISTRY/0002* and *AFII-REGISTRY/0004* confirms that independent glyph identifiers are not assigned to images that represent what is normally considered merely a difference in typeface. For example, the image "a" in the first of these inventories bears the glyph name "Lowercase Latin letter a", with no further specification of its shape. This practice suggests a criterion, although perhaps largely a subjective one, for constraining the degree of abstraction in glyph registration.

2.2.3.1. Glyph identifiers and font identifiers

Just as character-based data is used to represent the content of electronic text, glyph-based data is used to represent the presentational form of electronic text. As N 915 explains:

The result of the composition and layout process is a "final form document" which contains font identifiers, glyph identifiers, coordinate positions, along with either references to font resources, or the actual font resources themselves. Such a document form contains all the necessary information required to present the formatted document onto some presentation medium. An examples of such a final form document is an SPDL (ISO/IEC 10180) document instance.

Glyph images in physical renderings of documents are generated with reference to a combination of font identifier and glyph identifier. Following ISO/IEC 9541-1: 1991, a font can be understood as "a collection of glyph images having the same basic design, e.g., Courier Bold Oblique". To continue

¹⁴ Although the facts of early Cyrillic writing are poorly suited to a script-level definition of character, this definition is what UCS provides. Since UCS will probably underlie most of the operating systems of the immediate future, it will be necessary for Slavic mediaevalists to work within the UCS standard. Under these circumstances, it makes obvious sense for Slavic philologists to mold that standard as closely as possible to their particular needs. The disadvantages of a script-level character set do not pose insurmountable obstacles to processing, although they do impose a greater processing burden on some texts than on others. Slavic mediaevalists should recognize the limitations of applying the Unicode definition of character to early Cyrillic materials, and should nonetheless develop a character inventory within this framework that minimizes these limitations, even if it is unable to eliminate them. The inventory presented below is offered as a suggested solution to these problems.

¹⁵ One might reasonably refer to abstract glyphs as "glyphemes" their identifiable variant representations as "allo-glyphs", and their physical instantiations as "glyphs". Cf. the traditional linguistic identification of phonemes, allophones, and phones, and also the discussion of graphemes and allo-graphs in § 3.1.2 (fn), below.

the example from the preceding section, a glyph identifier might describe "Lowercase Latin letter a" irrespective of font, while a font identifier could specify the font from which that particular glyph is to be drawn when rendering a particular final-form document.

Glenn Adams (personal communication) has drawn my attention to an alternative glyph model. The model described above is based on a final-form document structure whose references have two components: , but a model that incorporates an additional level of indirection would allow a single font resource to store alternative concrete instances of the same abstract glyph. Such a model would be based on a tripartite reference structure , where the glyph identifier would point to a mapping table within the font that would point to the variant concrete representations of a single abstract glyph, and the glyph selection criteria would tell the rendering engine how to select among the variants. Both TrueType GX (Apple) and TrueType Open (Microsoft) allow for multiple instances of the same abstract glyph in a single font.

2.2.3.2. How to distinguish different abstract glyphs from variants of a single abstract glyph

Unicode regards different early Cyrillic and modern Cyrillic letterforms as a font change (*Unicode 44*), which, it seems, means a combination of a change in the inventory of AFII identifiers (e.g., U+0447 corresponds to both modern AFII 047/151-10089 [ѧ] and early AFII 052/345-10981 [ѧ]) and typeface-level changes that do not entail a change in AFII glyph identifiers (since many early Cyrillic letterforms are not assigned separate glyph identifiers from their modern counterparts in the AFII registry).

Study of the AFII glyph registry suggests that the ambiguities and uncertainties about how to distinguish different abstract glyphs from different instances of the same abstract glyph result from inconsistent implementation of the glyph cataloguing strategy. The introduction to *Register* specifies that "each entry within this printed glyph identifier register is comprised [sic] of an ISO/IEC 10036 glyph identifier, an AFII reference number, a sample glyph

shape, and a glyph description including at least a name or title for the glyph and *any significant information about the meaning or intended usage* (emphasis added)". A more useful strategy would have been to require that every glyph description identify specifically the distinctive properties of the glyph in question.

In particular, there seems to be no firm or consistent criterion for determining when an early Cyrillic letterform justifies the registration of a distinct glyph identifier and when the difference in form should be considered merely a difference in typeface. For example, U+0430 corresponds to both modern AFII 047/121-10065 (a), described as "Cyrillic small letter A" (with a note, not relevant in the present instance, that this glyph may or may not have the same shape as 000/141-97 "Latin small letter A"), and early AFII 052/361-10093 (a), described entirely as "Cyrillic small letter A, alternate (early), variant form 3 of (047/121)". Aside from the fact that the registry appears to contain a "variant form 3" but no "variant form 2" of 047/121, the definitions of these two abstract glyphs do not provide the information a user would require to identify whether a particular glyph instance represents one or the other of these abstract glyphs, or is a candidate for registration as yet a third abstract glyph. If a glyph instance matches neither of the register examples perfectly, the user has no guide for determining which graphic differences between the two examples are normative, and therefore relevant to the identification of new glyph instances, and which are accidental. Furthermore, although the distinct glyph identifiers in this example clearly differ in basic shape, an ill-defined difference in basic shape is not the definition of independent glyphhood; for example, Helvetica "g" and Times Roman "g", whose formal difference is subjectively comparable to that between the Cyrillic glyphs in question, both correspond to the same glyph identifier, AFII 000/147-103.

It is recommended that the AFII glyph registry be revised to include this type of informative description for all registered glyphs. The discussion below provides information that can be used to construct such guidelines for the Cyrillic portion of the registry, and also discusses the relative roles of font and glyph identifiers in representing early Cyrillic texts.

2.2.4. Similarities between characters and glyphs

Although encoding inventories (CHARACTER SETS, sets of CHARACTERS) and rendering inventories (FONTS, sets of GLYPHS) are crucially different, what they do have in common is that both represent ordered sets of text elements, so that both normally consist of an INVENTORY of distinct units and a specific ARRANGEMENT of these units.

2.2.4.1. Inventory

Many previous attempts to develop both modern and early Cyrillic character sets and fonts have suffered from inadequacies in inventory¹⁶. In some cases this reflects ignorance or errors of judgment on the part of developers, while in other cases it may result from legitimate differing opinions among Slavists about just which units are needed to encode early Cyrillic manuscripts materials. Inadequate inventory in official standards is one of the principal reasons why Slavists have been forced to design their own encoding and rendering schemes¹⁷.

2.2.4.2. Arrangement

No particular arrangement of characters is necessarily superior for all uses, which means that although a character inventory might be assigned a specific arrangement for the convenience of human

editors, a different arrangement would not render the set technically inadequate in the way that omissions would. Standardizing the order of a character set is nevertheless crucial, not for reasons of adequacy, but because computers identify characters by numerical values corresponding to their positions in a set, and interchange requires that users agree on a common correspondence of informational units and bit combinations. UCS is an obvious candidate for standardizing both the inventory and the arrangement of characters.

The order of units in some character sets, such as ASCII, supports the implementation of certain text operations as simple machine operations, such as sorting by bitwise comparison or case folding by bit masking. Problems arise with applying such strategies to multilingual data, and especially to data as orthographically varied as early Cyrillic writing. I do not believe that there is any advantage to attempting to follow some sort of alphabetic or other order in an early Cyrillic character set; once the orthographic data has reached this level of complexity, these text operations must rely on lookup tables.

The arrangement of glyphs in a font may or may not require standardization. As N 915 notes, glyphs may be specified as sequences of glyph identifiers (names), in which case an individual glyph representation can be retrieved from a font by searching for the identifier. But glyphs may alternatively be identified as sequences of indices into a font, in which case glyph order within the font may become significant.

2.2.5. Differences between characters and glyphs

Historically, different ISO subcommittees have been responsible for developing and maintaining character and glyph standards, and they have tended to operate independently of each other, with the result that they have developed somewhat conflicting terminology¹⁸. The following coordinated definitions are taken from N 915, with some simplification of the

¹⁶ For example, ISO 8859-5: 1988 omits the Ukrainian hard g (U+0491 "r"), although this character occurs in Ukrainian publications, including a small number of linguistic publications issued in the Soviet Ukraine. The same standard omits guillemets (U+00AB "«" and U+00BB "»"), although these are extremely common in Russian text. GOST 19768-74 ("old KOI-8"), a Soviet national standard that generally includes upper- and lower-case versions of all Russian letters, omits upper case hard sign (U+042A "Ъ"). This last decision may have been motivated by the fact that this letter is never found in word-initial (and, therefore, also sentence-initial) position, but it is nonetheless needed for text that may be printed entirely in upper case.

¹⁷ See Birnbaum 1989 for a critique of the design principles underlying ISO 8859-5: 1988. Scholarly requirements change with time, and the inventories proposed in the present article will surely also require extension eventually.

Units missing from one character set may sometimes be supplied from another through a switching protocol (e.g., ISO 2022: 1986), but in many cases these missing units may simply be unavailable in any standardized inventory. UCS is intended to make switching protocols unnecessary; all characters needed to encode all scripts are candidates for inclusion in this standard.

¹⁸ ISO/IEC JTC1/SC2 is responsible for such character sets as ISO/IEC 10646-1: 1993, while ISO/IEC JTC1/SC18 is responsible for such font standards as ISO/IEC 9541-1: 1991 and ISO/IEC 10036: 1993. N 915 is intended to stimulate the cooperative development of a common character/glyph operational model by these two committees.

discussion where this does not influence the evaluation of early Cyrillic encoding problems:

- (1) A character conveys distinctions in meaning or sounds. A character has no intrinsic appearance.
- (2) A glyph conveys distinctions in form. A glyph has no intrinsic meaning.

The relationship between character codes and glyph identifiers may be one-to-one, one-to-many, many-to-one, or many-to-many.

As an example of the character/glyph distinction, consider the alternatives for encoding and representing what we think of "á". This can be encoded as one or two characters: it is either a single character "a_with_acute_accent" or a sequence of character "a" plus character "acute_accent", with additional logic to specify whether "acute_accent" pertains to a preceding or a following character in the backing store. Similarly, "â" is either a single glyph "a_with_acute_accent_on_top" or a combination of glyph "a" plus glyph "acute_accent_on_top", with additional logic to specify that these two glyphs are centered in the same vertical space. The number of characters used to encode the data and the number of glyphs used to render it are mutually independent; one may map two characters to one glyph or a single character to two glyphs. As is illustrated by the preceding alternatives, the establishment of character and glyph inventories is to some extent subjective, and even arbitrary; for example, whether a character set includes "á" as one or two characters (or both) is decided by the designer, and either solution may be adequate for information interchange purposes¹⁹.

¹⁹ The failure among lay persons (and some professionals) to distinguish characters from glyphs may reflect a legacy of data processing operations where a one-to-one relationship between the two inventories does obtain. That is, if one's only character set is ASCII, and one uses a single font that assigns a single glyph to a single ASCII character, the abstract distinction between characters (informational units) and glyphs (presentational units), or between character sets (sets of characters) and fonts (sets of glyphs), has little practical significance for most end users. The simplest course of action for a Slavist without training in information science who wants to be able to print a set of early Cyrillic glyphs might seem to be to design a "character set" that corresponds on a one-to-one basis with rendering units.

2.2.6. Relationships between characters and glyphs

If we adopt the preceding definitions as a model for encoding and rendering electronic texts, operations that depend on character-level information, such as sorting, are performed on the plain-text character stream, while rendering is implemented by mapping characters to glyphs. This mapping should be able to take place without reference to fancy-text font tags, since font selection should never affect meaning. "Plain text must contain enough information to permit the text to be rendered legibly, nothing more." (*Unicode 10*) As is noted in § 2.2.5, above, mapping between characters and glyphs need not be one-to-one, but in most cases these mappings are algorithmic, and in many writing systems the glyphic representation of a character can usually be determined unambiguously, although often only with reference to the environment in which it occurs²⁰.

A commonly cited example of character/glyph distinctions in English involves such typographic ligatures as the single "ff" glyph that corresponds to a sequence of two "f" characters in certain fonts. For most text processing purposes (e.g., sorting), "ff" behaves like a sequence of two characters, yet for rendering purposes, it may be displayed most easily and attractively as one glyph. Many text processing systems, such as T_EX, recognize this character/glyph asymmetry by reading input with two "f" characters and generating output with a single "ff" glyph.

This situation is complicated slightly by an English typesetting convention that discourages ligation across parts of a compound word. For example, the three characters "ffl" are usually rendered as a single ligated "ffl" glyph, but the compound seven-character word "offload" is usually rendered in fine typography as six glyphs, with an "ff" ligature glyph followed by a separate "l" glyph at the

²⁰ Legible rendering does not necessarily mean culturally correct rendering. "The final appearance of rendered text is dependent on context (neighboring characters in the backing store), variations in typographic design of the fonts used, and formatting information (point size, superscript, subscript, and so on). The results on screen or paper can differ considerably from the expected or prototypical shape of a letter or character." (*Unicode 13*).

morpheme boundary. If the rendering system is not equipped to perform the linguistic analysis needed to identify such exceptions, they must be handled either manually or with a lookup table.

The preceding is a rare exception to the generally constant relationship between characters and glyphs in English writing, but such variable mappings are extremely common in early Cyrillic text. And not only can the same character be rendered with differing glyphs, but the same character in the same word in the same context in the same hand in the same manuscript can be rendered with differing glyphs. This precludes any possibility of mapping from character to glyph algorithmically, even with the help of a lookup table of exceptions (not only on the level of Cyrillic script in general, but even on the level of a single early Cyrillic electronic text fragment). And this, in turn, means that if glyphic information is needed in an early Cyrillic electronic text, this information must be encoded explicitly during text entry. A related problem is the reverse: not only may a character correspond in a nonsystematic way to variant glyphs, but glyphs of identical appearance may represent different characters in different texts, and even in the same text. These issues are discussed below.

2.2.7. SGML ENTITIES

Standard Generalized Markup Language (SGML, ISO 8879: 1986) provides a fancy-text mechanism that is capable of encoding character, glyph, and other information together. SGML entity sets are intended to provide a way of encoding information at a level higher than a character representation (Goldfarb 502), which means that early Cyrillic orthographic variants that do not fulfill the productive requirements for independent characterhood may nonetheless be encoded in an SGML document as distinctive SGML entities. An application may then map these entities onto character values (thereby conflating the variants for character-level processing) or onto glyph values (thereby rendering the variants differently). This use of SGML entities satisfies the general Unicode requirements that the character set not be overloaded with items that do not fulfill the definition of characterhood, while simultaneously providing an alternative mechanism that permits Slavic philologists who wish to perform

orthographic analysis on electronic texts to do so in a standardized way.

As was noted in § 2.2.2.1, above, UCS is not designed to represent distinctions outside the character level, which means that Slavic philologists who require such distinctions must employ fancy-text encodings, such as that provided by SGML. It is important for Slavists to understand that standardized encoding methods are capable of representing distinctions on different levels, and that if Slavists are to enjoy the interchange benefits of operating within the standards community, they will need to abandon the notion that all processing should be performed on plain-text encodings. This, in turn, may often mean that simple plain-text tools, such as GREP, will not be available to Slavists. The most productive approach to this reality is not to defy standards and create nonce character sets, as has been done in the past, but instead to use more powerful tools, such as SGML browsing software. Older plain-text tools were designed to solve problems quite different from those encountered in modern humanities computing.

2.2.7.1. DISPLAY ENTITY SETS

Goldfarb suggests that SGML entities that do not correspond to specific characters be mapped to descriptive definitions in a basic ENTITY SET (503) and to system-specific rendering instructions in corresponding DISPLAY ENTITY SETS (504). Goldfarb's model assumes that the basic entity set serves as documentation, while the display entity set will be invoked whenever rendering is required. It will be argued below that some philological analysis may also need to be performed on glyph-level data of the sort that is generally considered rendering information, but there is no reason that an SGML system could not use glyph information not only for rendering, but also for other specific types of text processing.

2.2.7.2. The WRITING SYSTEM DECLARATION (WSD)

Alternatively, *P3*, the guidelines developed by the Text Encoding Initiative (TEI) for encoding electronic texts with SGML for research in the humanities, combines the information that would be present in these two entity set declarations into a single WRITING SYSTEM DECLARATION (WSD), which is an

auxiliary SGML document that is able to assign to a single SGML entity simultaneously a character string, a UCS byte value, and a rendering value (such as an AFII glyph identifier) (679-99; see also TEI TR1 W4). For example, if an SGML document encodes an early Cyrillic U+0434 (А) by setting the active character set to Cyrillic and then inputting a 'd', the model illustrated in P3 could describe this in a WSD as follows²¹.

```
<character class=lexical>
  <form string='d'
    entityStd='dos'
    UCS-4='0434'
    afiicode='10069'>
    <desc>Old Slavonic small letter dobro</desc>
  </form>
</character>
```

This example assumes that 'dos' is the name for this entity in a registered entity set; the alternative attribute entityLoc may be used for entity names that are valid only locally. The string attribute may be omitted if the entity can be entered only as an SGML entity, rather than as a regular character in a transliteration system.

While currently there are no generally-available commercial applications capable of processing a WSD, P3 intends that standard character and glyph identifiers, such as the UCS and AFII numbers, be used to increase the portability of electronic texts by replacing prose descriptions and system-specific rendering instructions with references to public standards²². These references mean that a platform designed to understand UCS and AFII codes would be able to process any arbitrary WSD based on these inventories.

As the name implies, the Text Encoding Initiative has concentrated on developing standardized strategies for *encoding* texts, and there has been no comparable effort to develop text *processing* systems

that can use some of the more powerful features of these encoded texts. As a result, the WSD architecture has not yet been tested on material as varied, complicated, and at times contradictory as early Cyrillic writing. For example, the WSD provides a mechanism for recording the fact that certain text elements may be encoded in different ways at the character level (such as the representation of "á" as either one character or two). How a processing system should deal with the variability of these equations in different early Cyrillic orthographic microsystems is unclear, since the relationships among textual units, characters, and glyphs is not consistent across all early Cyrillic writing. Similarly it is unclear how a processing system should generate a character stream from an entity stream in a consistent way if a single entity can be mapped to alternative character representations.

2.2.7.3. The GLYPH Stream Alternative

It is also possible to encode text as a pure glyph stream, rather than a character or entity (or mixed) stream. One advantage of this strategy is that it allows orthographic analysis of the distribution of variant glyphs to be performed without reference to a WSD, thereby bypassing what is potentially a complex processing layer. A further advantage of this strategy is that the mapping from Cyrillic glyphs to characters is usually unambiguous, at least within a single document (although the reverse is rarely the case), which means that the character value is normally inferable from the glyph value. This approach then economizes by not requiring the user to encode redundant character-level information explicitly.

An application can normally convert from greater to lesser granularity (in this case, from glyph to character) by neutralizing distinctions. The glyph stream strategy is thus well suited to orthographic analysis, where the object of study is glyphic variation, and it does not prevent lexical or most textological analysis. It does, however, require that this neutralization of orthographic variants be performed by the processing application first, which means that the glyph stream strategy simplifies orthographic analysis at the expense of complicating character level (e.g., lexical and textological) analysis.

Unfortunately, one problem with the glyph stream approach is that character information is not always

²¹ The specific element and attribute names used in the WSD differ between P3 and TEI TR1 W4. The former represents the current TEI position.

²² System-specific renderings would still be required, but they would be developed for the relevant AFII inventory, rather than for each individual entity set.

unambiguously inferable from glyph information, since the same physical shape may correspond to different characters. For example, some Cyrillic written documents render both U+0432 (Р) and U+0434 (А) as a square "u" glyph, which means that if the different character values are to be recoverable, the glyphic principle would have to be compromised by encoding two different "u" units, one with the sound and meaning of U+0432 (Р) and the other with the sound and meaning of U+0434 (А)²³. The SGML entity strategy addresses this problem directly by not attempting to encode a glyph stream; an SGML entity may represent character and glyph information (distinctions in meaning and in form, respectively) simultaneously²⁴.

2.2.7.4. The CHARACTER stream alternative

One objection to the SGML entity approach described above is that it complicates defining the content of an SGML document instance in terms of characters and glyphs. Although SGML permits entities to represent non-characters, textual data in SGML documents is otherwise traditionally encoded as a character stream, and SGML entities that may represent other than character information can thus lead to what may be considered a confusing or undesirable (even if legal) mixture of levels²⁵. In the SGML

system envisioned in Goldfarb's description, entities that represent text normally represent character-level data, although they may be mapped to system-specific rendering units for display purposes. Furthermore, this reliance on characters, rather than glyphs, as the content of the text stream conforms to the N 915 definition of characters as information-bearing units. Because generalized tools may anticipate character data in this position, the use of glyph or other non-character data in its place may complicate the use of these tools, and thereby compromise one of the principal benefits of standardized encoding methods.

It would be possible to address this concern by encoding the text in the SGML document instance as character-level, rather than glyph-level data, and storing glyph-level information entirely in markup. This strategy ensures that the text output of a parsed SGML document (once markup has been removed) will be a pure character stream with no glyph-level data, which means that lexical and textological analysis, which is normally performed on the character level, can take advantage of standard SGML parsing tools. If one envisions an SGML system in which the basic character set used for data is UCS, this strategy ensures that a standard SGML parser will produce plain text without appeal to special processing. Rendering, on the other hand, will require that an application extract the necessary glyph information from markup.

As attractive as this strategy may be architecturally, it involves a significant cost in input control and integrity: because SGML is not concerned with data content, there is no way for an SGML parser to enforce the correct use of glyphic tags. For example, one might encode the Cyrillic character U+0434 (А), which I will represent here as "A", as "<RENDER AFICODE='10069'>A</RENDER>", where <RENDER> holds the glyph identifier as the value of the attribute AFICODE, and 10069 is the AFII identifier for "A"²⁶. The problem with this model is that an SGML parser cannot provide the declaration for <RENDER> with access to the content of the data stream, which means that there is no way for a parser

²³ A similar problem is posed by the use of letters to represent numbers in early Cyrillic writing, so that, for example, U+0432 (Р) sometimes represents the number "2" and sometimes does not. A solution for this is proposed in § 3.4.1.4, (fn), below.

²⁴ As is noted above, one could maintain this distinction in a glyph stream encoding by creating two identical "u" glyphs with different glyph identifiers, one named something like "Cyrillic small letter quadratic ve (early)" and the other something like "Cyrillic small letter quadratic de (early)". The problem with this strategy is that the graphic identity of these glyph images may be an important feature of this type of writing, and this identity is most accurately and efficiently maintained by using a single glyph name and number. A further alternative is described by Robinson and Solopova, whose encoding decisions are sensitive to both character (graphemic) and glyph (graphic) features, but without incorporating a separation of the two levels into the encoding architecture (26-29).

²⁵ Sets of SGML entities that represent text elements are called "character entity sets" in the standard, although the reference there to "graphic characters" (Goldfarb 255) indicates that the term "character" does not have the same meaning in this standard as it does in UCS.

²⁶ This example assumes a UCS data character set of which U+0434 (А) is a member. Appropriate adjustments would have to be made in an eight-bit environment.

to prevent the user from wrapping an inappropriate AFII identifier around a character. This system may be well suited to an implementation that must allow free association of characters and <RENDER> values, but this is not the case with early Cyrillic writing. And while a routine external to the parser could enforce the appropriate correspondences, such an approach would seem to sacrifice one of the principal advantages of SGML: it is capable of controlling and verifying the structural accuracy of much of what is input. The use of a <RENDER> element invites error by imposing a burden on the user that would be managed by the parser under either the WSD strategy or the use of a basic glyph stream encoding system.

An alternative strategy involves using different elements, rather than different attribute values of a common <RENDER> element, as a method of recording glyph identifiers. For example, the SGML document type definition (DTD) could define a separate element for each identifier that is legal in the writing system, and then encode the example from the preceding paragraph as <AFII10069>A</AFII10069>. But this strategy is subject to the same limitation noted above: there is no way for an SGML parser to prevent a user from wrapping the <AFII10069> element around the wrong character. Not only is SGML unable to police the relationships between attributes and data, but it is unable to police the relationship between elements and data, because it has no access whatsoever to specific data content.

2.2.8. Conclusion

The nature of early Cyrillic writing is such that glyphic information frequently is not predictable from character information and character information may not always be predictable from glyphic information, which means both types of information must be encoded explicitly in electronic texts. Of the three alternatives for encoding this composite information, a true glyph stream model is inadequate and a character stream model with glyph information encoded in markup invites user error that an SGML parser is unable to monitor. Only the SGML entity model enforces the correspondence of characters and glyphs and provides a mechanism for generating either a pure character stream or a pure glyph stream from the hybrid SGML data stream.

3. Issues, problems, and opportunities for Slavic philologists

3.1. LEXICAL, TEXTUAL, ORTHOGRAPHIC, and PALÆOGRAPHIC analysis

The availability of two basic levels of symbolic representation, characters and glyphs, makes it possible to optimize each inventory for a particular set of tasks. In principle, the character inventory could be designed to facilitate lexical and textual analysis, while the glyph level could be designed to facilitate orthographic analysis and typographically exact rendering. In fact, though, the inherent contradiction underlying the definition of character (which is based on scripts, although it is only meaningful for orthographies) compromises the utility of the resulting system severely.

3.1.1. Lexical and textual analysis and the character inventory

Lexical and textual analysis is normally sensitive to distinctions in sound or meaning (characters), rather than distinctions in letterform (glyphs). But because the current UCS early Cyrillic character inventory includes some items that do not distinguish sound or meaning in certain texts, lexical and textual analysis will often require the further neutralization of certain character distinctions during processing. This type of character neutralization is already required for most languages for case-insensitive searching or sorting; what distinguishes the early Cyrillic situation from a general operation like case folding is that specific neutralizations must be adjusted to the orthography of individual documents. Furthermore, this reduction in the character inventory through neutralization may require a separate expansion of the inventory; because certain text elements that are most commonly glyphic variants of a single character may nonetheless function as independent characters in some documents, users must be able to incorporate these distinctions into lexical and textual operations that normally apply only to characters.

The preceding means that the character inventory proposed here is not appropriate to all early Cyrillic texts, and that no such universal character set is possible. In fact, the character set proposed

here may not be fully appropriate (without adjustment) to any early Cyrillic texts. To take only one example of a problem in the use of the character inventory for lexical or textual analysis, U+0443 (ѣ), U+0479 (ѣ), and U+0475 (ѣ) are all already present in UCS. In some early Cyrillic manuscripts, U+0443 (ѣ) and U+0479 (ѣ) represent the same sound and meaning, which means that distinctions between these two characters must be neutralized for these manuscripts. In other early Cyrillic manuscripts, U+0443 (ѣ) and U+0475 (ѣ) represent the same sound and meaning, which means that distinctions between these two characters must be neutralized. The three-way distinction implied by the three characters in UCS represents the union of these two systems, and does not correspond to the use of these text elements in any individual source.

A more significant problem arises when one needs to compare manuscripts that have different logical character inventories (e.g., U+0443 "ѣ" = U+0479 "ѣ" in one manuscript and U+0443 "ѣ" = U+0475 "ѣ" in another, and the situation is even more complicated with nasal vowel letters, about which see § 3.4.4.1, below). Systems for preparing dictionaries, critical editions, or other materials that may be based on a corpus of manuscript sources will need not only to be able to conflate two characters for purposes of comparison, but also to be able to do so differently in different sources that are then to be compared with one another.

Although an ideal universal early Cyrillic character set is unattainable, UCS will be the global multilingual character set for most systems and applications in the future, SGML is a growing standard for encoding texts in the humanities, and Slavic mediævalists must decide how best to adapt these tools to their specific needs. Most lexical or textual analysis of most early Cyrillic documents can be conducted by using the character inventory proposed here (or, more precisely, a subset thereof), without reference to other distinctions in text elements. While this is not a universal or fully uniform solution, no such solution is possible, and the compromise proposed here will enable Slavic mediævalists to use modern tools and operating environments to conduct most lexical and textual research.

3.1.2. Orthographic and palaeographic analysis and the glyph inventory

In early Cyrillic writing, a single letter of the alphabet may have more than one letterform. Within Slavic philological tradition, this variation has two basic subtypes, which I identify as ORTHOGRAPHIC ("which of the available letterforms is used in this instance to represent letter *X*?") and PALÆOGRAPHIC ("what is the physical shape of letterform *Y* in this instance?")²⁷. This distinction is subjective and gradual, and the system must be flexible enough to allow for individual scholarly differences in classification.

3.1.2.1. Orthographic variation

Orthographic variation is a common object of study by Slavic philologists because it may provide evidence in at least three areas. First, correspondences or differences among manuscripts in the distribution of letterforms that do not distinguish meaning (allographs) may indicate specific scribal schools or traditions. Second, widespread agreement in the nonsystematic distribution of letterforms that do not distinguish meaning, beyond that which might be expected to arise by chance, may indicate that one manuscript is a copy of another²⁸. Third, the distribution of letterforms that do not distinguish sound or meaning in the orthography of a manuscript may nevertheless reflect linguistic distinctions retained from some ancestor of that manuscript. This third

²⁷ Graphemic analysis traditionally recognizes a GRAPHEME as a distinctive unit of writing (cf. phoneme). Its nondistinctive positional or free variants are called ALLOGRAPHS (cf. allophones), with each physical instance of a letterform called a GRAPH (cf. phone). I use ORTHOGRAPHY to refer to both the distribution of graphemes and the relationships between graphemes and their allographs. I use PALÆOGRAPHY to refer to the description of graphs. Cf. the slightly different terminology in Miklas 1988.

From a different perspective, palæography can be considered to address the question "is there a relationship between the shape of the letterform *Y* and the age or place of origin of the manuscript?", while orthography can be considered to address the question "is there a rule that governs the use of letter(form) *X*?"

²⁸ Only nonsystematic agreement is decisive in this respect, since systematic agreement may show a common scribal tradition, rather than a physical relationship between two specific manuscripts.

feature arises because the sounds associated with individual letterforms may differ geographically and chronologically, and conservative orthography may reflect distributional rules based on former linguistic features.

This type of orthographic analysis has been a central component of Slavic philology for over a century, and because of its statistical nature, it is particularly amenable to computer processing. The basic principle is to determine the individual orthographic norms employed by a particular scribe in a particular manuscript, and then to analyze the significance of deviations from these norms (cf. Mathiesen). Not all Slavic philologists may want or need to encode this type of nonlinguistic orthographic variation, but an encoding system that is to serve the general needs of the Slavic philological community must nonetheless be capable of representing this information for those who do need it. Because the distinctions in question do not reflect differences in sound or meaning, they must be encoded at a level other than that of the character set. For reasons discussed in § 2.2, above, I propose the use of SGML entities for this purpose.

There are at least three reasons why early Cyrillic orthography requires that this type of variation be encoded explicitly:

First, such variation is most commonly governed by orthographic rules, or norms. If the variation were always free, there would be no purpose to encoding it (other than an aesthetically-motivated attempt at more exact physical reproduction of the original, in which case a photographic facsimile would be more appropriate), since the individual differences would not provide a classificatory or analytic fingerprint of the manuscript. In other words, what Slavists consider orthographic variation is an important traditional object of scholarly inquiry.

Second, orthographic norms differ from manuscript to manuscript. If a single set of orthographic norms obtained in all early Cyrillic writing, there would also be little classificatory or analytic purpose in encoding orthographic variation. Because orthographic variation is not constant across different manuscripts, it must be encoded separately in each electronic document.

Third, orthographic norms are never observed consistently, and it is deviation from the individual

norms of an individual manuscript that provides the best evidence for ancestor texts, vernacular linguistic interference, and other valuable philological and linguistic details. If orthographic norms were observed consistently within a single manuscript, there would be no need to encode the variants, since their distribution could be restored algorithmically.

3.1.2.2. *Palæographic variation*

Other variation in the shapes of early Cyrillic letters is traditionally the object not of orthographic analysis, as described above, but of palæographic description. I have not attempted to provide for the encoding of palæographic variants for two reasons. First, it does not seem possible to design a system capable of distinguishing the full range of palæographic variation found in early Cyrillic writing, and I see no way to identify a useful level of granularity for palæographic variation, which is essentially analogue in nature. Second, while electronic texts are well suited to orthographic analysis, I see no advantage to performing palæographic analysis on electronic texts²⁹.

As is mentioned above, the boundary between orthographic and palæographic variation is often subjective, and I have made individual decisions based on my experience with primary and secondary sources in early Slavic manuscript studies, and in consultation with other Slavic philologists. The system I outline here is flexible; those who require less granularity can constrain themselves to a subset of the SGML entities I propose, while those who require greater granularity can expand the present inventory, either as a local modification or by submitting additions to the appropriate authorities for formal registration.

3.1.3. *Font identifiers*

The glyph descriptions found in the AFII inventory suggest that a letterform whose distinctive quality is governed only by the aesthetic design principles underlying the typeface to which it belongs

²⁹ I refer here to texts encoded as character or glyph streams. Digital images of manuscript pages are very well suited to some type of palæographic analysis, and particularly to quantitative studies.

should not be assigned a unique glyph identifier. From this perspective, a difference in basic letterforms within a single manuscript source, when it is not perceived by Slavic philologists as analogous to a change from one modern typeface to another, should not require a user to change fonts. This means that such variants should be incorporated into the same font, and should be assigned separate glyph identifiers³⁰.

Consider the letter represented by U+0438, which may have a crossbar that slants upward from left to right (и), that is completely horizontal (И), or that is drawn with some intermediary orientation. The angle of this crossbar is a common feature of palaeographic description, but not of orthographic analysis, and I know of no early Cyrillic manuscript in which this distinction has any non-palaeographic significance. A comparable situation is illustrated by the variants of U+0442 that have a single stem (АФII 047/144-10084 "т") or three stems (и), where the distinction would not normally be the object of orthographic study (although it is an object of palaeographic description). (Intermediate shapes are also possible, such as those with unequal descending serifs on either end of the crossbar, and a "two-and-a-half-legged" version, where the left serif reaches the baseline while the right does not). One might propose that only a single glyph identifier be registered for each of these sets, with the different letterforms relegated to different fonts, so that font designers might produce a font with one version to correspond to one manuscript hand and an almost identical font, but with different "и" or "т" letters, to correspond to a different manuscript hand. In this way, the glyph identifier level would be designed to exclude purely palaeographic variation, thereby optimizing for orthographic analysis.

One significant complication for this approach is the subjectivity of the division between orthography and palaeography, a distinction even less clearly defined than the distinction between character and glyph. This complication is addressed in the present report by using as inclusive a glyph inventory as possible, assigning independent glyph

identifiers to any uncertain cases, including the variants noted above. But because graphic variation is infinite, it is not possible to register all conceivable variants, and it would also not be desirable to encode variation that would never be put to any use in research or editing. The glyph inventory proposed here represents a compromise, which, I feel, is rich enough to encode almost all orthographic distinctions that will be needed for ustav (roughly analogous to uncial) and poluustav (roughly analogous to semiuncial) early Cyrillic writing.

This type of compromise, like the compromise character set discussed above, may not be ideal for any particular encoding task, but it should at least be adequate for most such tasks. One requirement that ensures the vitality of such compromise proposals is that the system remains open, so that additional variants can be added as needed, either by individual researchers (at the expense of deviating from a standard) or by submitting proposed additions to the appropriate standards bodies for official inclusion.

3.1.4. Conclusions

3.1.4.1. Characters

The most practical solution to the problem of unifying what are multiple Cyrillic orthographies into a single Cyrillic script is to optimize the character set for lexical, textual, and grammatical analysis, while treating particular graphic features with constant semantics in a consistent manner. Applications must be designed to allow users to neutralize distinctions between individual characters, thereby converting the general, script-based character set into a writing-system-based character (or text element) set, customized to the orthography of an individual text.

3.1.4.2. Glyphs

The glyph inventory serves two main functions: it is primarily intended to support rendering, but it may also be used in orthographic analysis, which focuses on the distribution of variant letterforms that represent the same character.

The appropriate level of detail for orthographic analysis is a subjective matter, but I do not believe that the inventory presented here omits anything that is likely to be needed widely. The present

³⁰ This discussion is based on the model of final form documents discussed in § 2.2.3.1, above.

inventory makes no attempt at a consistent representation of palaeographic variants that are not traditional objects of orthographic analysis.

A small number of essentially palaeographic (non-orthographic) variants is included in the glyph inventory for rendering purposes, even though these units may not be needed for any sort of analysis. I have included those variants that are most distinct graphically (a subjective decision) and that are widely used in *ustav* and *poluustav* writing, criteria selected because they are likely to be important to font designers. Because subjective evaluations may differ, and because other Slavists may have a need for or interest in additional variant letterforms, the glyph inventory should remain open to expansion, with the requirement that any new proposal be accompanied by an explicit statement of its distinctive features, including enough information to differentiate between it and all other registered glyphs with which it might be confused.

3.1.4.3. *The character/glyph boundary*

The most significant problem with the character/glyph model for encoding early Cyrillic writing is that the logical character inventory varies from manuscript to manuscript, yet UCS mandates a unified script-level character inventory. The present proposal addresses this problem by regarding the character level as adjustable in two ways: 1) a system should neutralize character distinctions where two UCS characters do not discriminate meaning in a specific source, and 2) a system should promote a glyph-level distinction to a local character-level one where two text elements that are not separate UCS characters nonetheless function as independent characters in a specific source.

3.1.4.4. *Font*

Font identifier changes should not be required except where they correspond to what modern typography would consider changes in typeface or in weight, size, and similar features. Decorative initials or superscript letters, which may differ not only in size and position, but also in basic shape, from their regular counterparts, may reasonably belong in different fonts. Fonts may be based on specific manuscript hands, on modern typefaces, or on any other principle that conforms to the ISO definition of font

cited in § 2.2.3.1, above: "a collection of glyph images having the same basic design, e.g., Courier Bold Oblique"³¹.

3.2. *Normalization*

Early Cyrillic electronic texts may be used for a wide variety of investigations, including linguistic, orthographic, palaeographic, lexical, and textological study, and these disciplines often require different levels of granularity in distinguishing text elements. That the character or glyph or entity set may be capable of encoding a very large range of items does not compel scholars to use all of them, and at least three types of normalization are readily available.

First, scholars who do not require distinctions at the fancy-text level need not use any fancy-text encoding features. Neither minimally nor maximally normalized texts are inherently superior for all types of analysis, and the degree of normalization for a particular project should be determined by the intended analytical purpose of the electronic text³².

³¹ As is noted in § 2.2.3.1, above, this description reflects a model that is incapable of storing multiple concrete representations of the same abstract glyph in a single font. A font technology that made use of glyph selection criteria could assign decorative and regular glyphs the same glyph identifiers in the same fonts, and select among them by applying glyph selection criteria.

³² One reason for preparing electronic texts according to generally accepted standards is that they can be exchanged among users, and it might be argued that encoding a text according to one user's minimal requirements may make the resulting files inappropriate for another user's maximal requirements. The technical answer to this objection is that one is free to enrich an electronic text by undoing normalization to suit one's needs; this may not be an easy task, but it is easier than having to encode the entire text from scratch. The academic answer to this objection is that it is not reasonable to expect the editor of an electronic text to undertake, with no benefit for his own work, the time and effort required to introduce additional encoding distinctions, as well as the time and effort then required to remove or neutralize these distinctions in order to perform the type of analysis for which the text was originally prepared. The present proposal is intended to broaden possibilities and opportunities, and not to dictate requirements.

One example of this type of cost is reported by Robinson and Solopova (25), who noted that even where recording finer distinctions during transcription imposed no time penalty, they observed a significant drop in accuracy, due to the transcribers' distraction while concentrating on maintaining these additional distinctions.

Second, manuscript material is analogue, and can contain an infinite variety of letterforms, while digital texts are encoded using finite character, glyph, and entity sets. This means that some degree of normalization is inevitable during data entry, as transcribers conflate instances whose differences they consider insignificant for their purposes. This type of normalization can occur in either plain or fancy text.

Third, normalization may alternatively (or additionally) be performed on the processing level, so that distinctions recorded in the plain or fancy encoded text may be leveled out during analysis. This normalization can be performed either by filtering the text before inputting it into whatever analytical processing is to be performed, or by allowing the process itself to perform the normalization. This last type of normalization is particularly important because it allows a single electronic text to be used for multiple purposes. For example, a system may record very fine orthographic detail in order to render the appearance of the original source as closely as possible, while normalizing many of these details when it processes the text in order to generate sorted word lists. This is the essence of the use of alternative entity sets or a WSD, described in §§ 2.2.7.1-2, above³³.

Robinson and Solopova (22) distinguish four basic types of transcription, listed here in order from most to least punctillious in preserving graphic distinctions: GRAPHIC, GRAPHETIC, GRAPHEMIC, and REGULARIZED. A GRAPHIC transcription records the exact physical features of the writing as closely as possible; this is well suited to palæographic analysis, but perhaps not as well suited as the use of graphic images. A GRAPHETIC transcription records distinctive differences in letterforms (e.g., jery with a front jer first component [“*ѣ*”] vs jery with a back jer first component [“*ѣ*”]), and is well suited to orthographic (and, in some cases, linguistic) analysis. A GRAPHEMIC transcription conflates variant

letterforms, but does not normalize spelling on the word level; Robinson and Solopova favored graphemic over graphetic transcription for their project because it simplified a number of complex problems. A REGULARIZED transcription normalizes all spelling, which is convenient for textual analysis.

Some of the normalizations inherent in the last three types of transcriptions could alternatively be performed at the processing, rather than the encoding, level. This means that an encoding that observes a high degree of granularity is not normally problematic, since a processing application can neutralize unneeded distinctions³⁴. Furthermore, within

³⁴ Conversely, a processing application can also sometimes restore or reintroduce distinctions that have been normalized during transcription, but only when those distinctions are dependent on factors retained in the encoded text. For example, Arabic encoding may not need to distinguish initial, medial, final, and isolated positional variants where these distinctions can be restored during processing by examining the environment in which a character occurs. The situation is quite different with early Cyrillic, as is discussed in § 3.1.2.1, above.

Robinson and Solopova (25-27) argue against the practicality of graphetic analysis on three grounds:

First, as is noted above, Robinson and Solopova found graphetic transcription more prone to error than graphemic transcription. Interestingly, they found that both less normalization (in graphetic transcription) and greater normalization (in regularized transcription) led to more error and inconsistency than an intermediate graphemic transcription.

Second, Robinson and Solopova were uncomfortable with situations where graphetes (letterforms) may overlap graphemes (abstract letters), such as instances in some of their manuscripts where long “s” is physically indistinguishable from “f”. Leveling these into a single encoding unit sacrifices the ability later to generate graphemic transcription from graphetic transcription automatically, while failing to level them subverts the graphic basis of the transcription system. This is the essence of the reservations to glyph stream encoding that I describe in § 2.2.7.3, above, and I suggest that the best solution is to avoid the problem by using SGML entities. This issue is a conundrum for Robinson and Solopova because they are trying to decide whether to encode form *or* function, while the system described here is designed to encode both simultaneously.

Third, Robinson and Solopova were unable to identify a principled way of determining how many graphetic variants to recognize for a particular grapheme. In part, this problem arises from a conflation of two types of information that are usually distinguished by Slavists into a single “graphetic” level: 1) which basic variant form of grapheme (letter) *X* did the scribe use here, and 2) what physical shape did this form take? The dividing line between these questions is often subjective, but, as is discussed in § 3.1.2, above, Slavists have traditionally

³³ This type of normalization is part of most standard alphabetizing routines, where distinctions between upper-case and lower-case letters that are maintained for rendering are neutralized for sorting. A system of this type designed for early Cyrillic materials (based on nonstandard character and glyph inventories) is described in Birnbaum 1988.

Robinson and Solopova's framework, distinctions that are graphemic in one type of early Cyrillic writing may be graphetic in another. For example, in an East Slavic manuscript where U+046B (ѡ) represents the sound [u], one could argue that this letterform functions not as a separate grapheme (which it would be in an Old Church Slavonic manuscript, where it has a different sound), but as one more variant (graphete) of the grapheme associated with the sound [u], alongside U+0443 (ѣ) (in some manuscripts), U+0479 (ѣѣ), and the ligated form that is not part of Unicode (AFII 052/326-10966 "ѣѣ"). Many types of analysis (textological, lexical, grammatical) of East Slavic manuscripts can be simplified by encoding all of these [u] letterforms, including U+046B (ѡ), identically (as is done, for example, in *Pamjatniki*).

As was noted above, Slavic philologists often need to compare manuscripts based on different (sometimes very different) orthographic systems. If two text elements function as independent characters in manuscript A, but convey the same sound and meaning in manuscript B, it may be important for a study of manuscript B to neutralize the distinction, either during input or during processing. But if the two manuscripts are then to be compared, it may then also be necessary to neutralize the distinction in manuscript A for comparative, rather than internal, reasons. As Slavists work with larger corpora — a likely situation as scholars share texts that may have been encoded originally in different ways and for different purposes — juggling neutralizations may impose a serious intellectual and processing overhead. At a minimum, the need for different types of normalization of the same text, depending on the specific analysis being performed, demonstrates the need to locate at least some of this neutralization at the processing, rather than input, stage.

It is expected that Slavists will follow a degree of normalization that is appropriate for their work.

distinguished orthographic features (such as the two forms of jery noted above) from palaeographic ones (such as whether jery has a connecting bar joining its two parts, and whether this connection is at the baseline or midline). I would consider this type of orthographic feature to be part of a graphetic transcription, while palaeographic features would more properly be part of a graphic transcription.

The inventories proposed here are designed to support what Robinson and Solopova would consider graphetic transcription, or transcription on what Slavists consider an orthographic level, as well as any less granular transcription (graphemic or regularized, in Robinson and Solopova's terminology). This basic system can be extended to support more discriminating palaeographic (graphic) transcription by expanding the glyph and entity inventories (the character set will not require modification).

3.3. Ambiguities

As Robinson and Solopova noted in their work with English manuscripts, two different letters of the alphabet (graphemes) may be represented in a manuscript by the same physical letterform (graphete). This situation also arises with early Cyrillic materials, and is particularly problematic with non-alphabetic superscript marks³⁵.

U+0486 (҃), a smooth breathing mark, traditionally looks like an apostrophe above a base character. U+0311 (̂), a kamora, traditionally looks like an inverted breve (a curved circumflex) above a base character³⁶. But a scribe who is writing quickly or carelessly may draw a smooth breathing mark that is rotated enough to be graphically identical to a kamora, or vice versa. Because breathing marks and kamora normally fulfilled different functions, it is usually possible to determine which was intended, much as modern readers decipher modern sloppy handwriting by examining the environment. But how should such forms be encoded?

³⁵ Nonalphabetic superscript marks in Cyrillic and many other writing systems are physically smaller and topologically less intricate than alphabetic marks. This means that a small amount of distortion in a superscript mark may be more likely to lead to illegibility or ambiguity than the same amount of distortion, measured in absolute physical area, in an alphabetic mark.

³⁶ Rough and smooth breathing marks in early Cyrillic orthography do not represent any phonetic content, unlike in Classical Greek. When applied to early Cyrillic writing, "breathing" should be understood as a palaeographic, rather than phonetic, term. Cyrillic breathing marks most often (although not always) indicate word or syllable anlaut, but this marking is usually redundant, since syllable structure can usually be inferred from alphabetic characters.

Three basic strategies are available:

- (1) Encode text elements according to their form. If a text element is indistinguishable from the basic shape of a kamora, it should be encoded as a kamora, even if its function seems to be that of a smooth breathing (or if the function is unclear). This will simplify rendering (map every kamora character to a kamora glyph), but complicate linguistic analysis (one cannot automatically distinguish true kamora from kamora-shaped smooth breathing).
- (2) Encode text elements according to their function, as determined by expert analysis. If it functions as smooth breathing, it is a variant glyph representing the smooth breathing character. This will simplify analysis (one can easily generate a list of all examples of true kamora), but complicate rendering (there will be no way to recover which smooth breathing looked like a kamora and which looked like regular smooth breathing). This solution is not available when the function cannot be deduced unambiguously from context.
- (3) Encode ambiguous text elements specially, in order to represent both their form and their function.

The third solution is the only one that prevents the loss of information, and it can be implemented in the model discussed here only in fancy text, where an SGML entity can associate a character value corresponding to meaning with the glyph value corresponding to shape. The inventory of such entities cannot be predicted, since it depends on individual hands (and on style of writing), and no attempt has been made to account for it in the current inventories. It is assumed that users who require additional entities will add them to the basic list provided here.

There may arise situations where the editor is unable to determine which of two possible readings is intended. This is a genuine ambiguity in the original, where the form is known and the function is unclear, and it cannot easily be transferred to an electronic text in which characters are defined by their ability to discriminate function. Such situations normally require individual attention, and the character inventory is not intended to be able to

resolve them without discussion³⁷. Robinson and Solopova resolve true ambiguities by an "arbitrary decision" (30); in such cases, it is helpful if the same arbitrary decision is made whenever the same ambiguity arises, and this policy should be documented in the edition. But even with consistency and careful documentation, the failure to encode an ambiguous reading differently from an unambiguous one means a loss of information. This can compromise both individual examples (a scholar would not choose to illustrate a phenomenon with an ambiguous example if an unambiguous one were available) and statistics (one cannot provide exact or meaningful counts of different types of data if some examples were classified one way or another arbitrarily).

3.4. *Special problems*

Any standard for interchange represents a best effort to meet all of the critical needs of those who will use the standard. As has been shown above, the adoption (with slight modifications) of UCS as a character standard, the AFII inventory as a glyph standard, and SGML as a document architecture standard provides the following advantages:

- (1) A character level that is optimized for lexical and textological analysis, such as compiling dictionaries or preparing critical editions. Unneeded distinctions in the character inventory can be neutralized during processing. Additional distinctions needed for this type of analysis can be obtained by "promoting" the appropriate glyph-level distinctions to character level where needed.
- (2) A glyph level that is optimized for orthographic analysis. This level is available not only for performing this type of analysis, but also for rendering the results of character-level processes. For example, it should be possible to prepare a critical edition by comparing variants at the character level but printing or displaying their glyphic values.

Beyond these general issues, the following specific problems must be addressed: composite characters (including non-spacing marks, jotation, jery, *i*-type

³⁷ One solution would be to create a special character whose meaning incorporates the ambiguity in question.

letters, \ddot{u} and \ddot{y} , ligatures); superscription; case distinctions; alphabetic order; the relationship among я , ia , and Ѧ ; u -type letters; jers; nasal vowel letters; o -type letters; e -type letters; Glagolitic; and Greek. This section provides the reasoning behind the encoding decisions made in each of these cases, with the goal of providing the best automation of processing, the easiest method of encoding, and the most appropriate method of rendering.

3.4.1. Composite characters

The definition of nonspacing marks as characters that combine relatively freely with a large number of base characters (see § 3.4.1.1, below) raises several questions. Some characters, such as accent marks, which can occur over any vowel letter, or abbreviation marks, which can occur over any letter or set of letters, are unambiguously promiscuous in the combinations in which they participate, and are therefore obvious candidates for encoding as independent nonspacing characters. Other candidates, some spacing and some not, are combined with a much more restricted set of characters, which makes it necessary to determine on an individual basis whether they are most efficiently encoded independently (as separable marks), or whether the few combinations in which they occur should instead be encoded as unitary characters. These individual cases are discussed below.

3.4.1.1. Nonspacing marks

One complication of early Cyrillic writing is that any of a large number of superscript items, often called floating diacritics, may appear over almost any base item (including accent marks that have been displaced to positions over consonants in abbreviated words, multiple marks on single base characters, etc.)³⁸. To reserve individual space in a character set

³⁸ Terms such as FLOATING DIACRITIC and ACCENT MARK are often used indiscriminately to identify any non-letter that occurs over alphabetic characters. These concepts may sometimes need to be distinguished, since not all such marks are functionally diacritics and not all such marks record accentual features, but the distinction is not crucial to the current discussion. What is meant is NONSPACING MARKS, on which see *Unicode 18*.

Miklas (1988) divides early Cyrillic marks into LINEAR (marks that are never supralinear, which essentially means punctuation),

for all such combinations of base plus superscript item as single "combined" characters would require the establishment of an impractically large inventory. Furthermore, the relatively free combination of base letters and superscript marks attested in early Cyrillic writing demonstrates that this process of combination was essentially productive and dynamic. This productivity, in turn, implies that the discovery of previously unattested combinations would not surprise Slavic philologists, which, finally, means that an optimal character set must allow for the representation of such unanticipated combinations, a flexibility that would not be possible if all combinations had to be defined in advance and encoded individually³⁹.

Accordingly, the character inventory proposed below encodes as an independent character any item that combines relatively freely with a large number of other items. In practice, this means primarily that superscript accentual and other marks are encoded as discrete characters, which can be superimposed during rendering on any base (alone or in combination with other superscript units), unless specific reasons justify treating a particular combination differently. In keeping with Unicode practice, nonspacing marks should be entered into the backing store following the base over which they are to be rendered⁴⁰.

SUPRALINEAR (marks that are only supralinear, such as non-spacing accent marks and others), and NEUTRAL (marks that may be either linear or supralinear, i.e., alphabetic symbols). While descriptively accurate, this terminology does not make clear that alphabetic symbols are not positionally neutral; they are inherently (unmarkedly) linear, although they may be written supralinearly in a restricted set of environments.

³⁹ Amending an ISO registered character set can be a complex and time-consuming matter, and a scholar who needs to represent text, if forced to choose between deviating from a standard and waiting until ISO protocols for amendment can be observed, will be likely to choose the former. A system that can encode unanticipated combinations, should a need suddenly and unexpectedly arise, thus has a clear practical advantage over one that cannot, and encoding "floating diacritics" separately provides precisely this needed flexibility. Just as a character set does not legislate which sequences of alphabetic character are legal, an early Cyrillic character set should not legislate acceptable combinations of base plus superscript.

⁴⁰ The accent marks neither follow nor precede their base logically, since the linguistic features they represent (stress, tone) are articulated simultaneously with the segmental features

"Combines relatively freely" and "large number of other items" are subjective terms that may be understood variously, but my application of these criteria is informed by a consideration of mediæval Slavic orthographic practice, which may indicate that certain apparent combinations functioned as units, while others did not. Additionally, the inventory below follows the general practice in Unicode, where precomposed characters are admitted only if they are present in widely used existing character sets. Thus, the only precomposed characters in my early Cyrillic inventory are those already present in Unicode, or those that do not reflect relatively free combination with a large number of items. These last cases are discussed individually below.

From an information interchange perspective, there is no requirement that modern encoding correspond to our reconstruction of mediæval orthographic thought. But there is a strong information processing argument in favor of respecting our understanding of mediæval orthographic practice: philologists most often need to operate with units that are orthographically meaningful, and programming tasks are likely to be simplest to write and most efficient to execute if elemental encoding units (characters) correspond to the units on which processing operations are to be performed. In other words, from an engineering perspective, early Cyrillic coded characters are most effectively and efficiently regarded as representing the set of items that seem to function as atomic informational units in the practice of mediæval Slavic scribes⁴¹.

represented by the base character. Superscript letters in early Cyrillic writing usually follow their bases logically, although this is not always the case. Marks of abbreviation may occur over the beginning, middle, or end of an abbreviated sequence, and do not therefore logically precede or follow their bases consistently. Breathing marks have no phonetic content in Cyrillic, but they frequently mark word or syllable anlaut, which would suggest that they logically precede their bases. I nonetheless suggest following the graphically-based Unicode decision to encode all nonspacing marks after their bases in the backing store. See *Unicode* 19-21 for details.

⁴¹ Such determinations are not always cut and dried, and some operations may more easily be performed on one type of encoding while other operations are more easily performed on another.

Because glyphs embody shape without meaning, whether to represent composite characters as unitary composite glyphs or as ad hoc compositions of independent glyphs is essentially an engineering, rather than an informatic, problem. But because it is likely to prove impractical to anticipate all possible combinations of base character plus nonspacing marks, the glyph inventory proposed below also assumes that composite glyphs that reflect productive combinations will be constructed on the fly during rendering, rather than stored as independent members of a font.

3.4.1.2. Jotation

Early Cyrillic writing contains several jotated (also called prejotated) letters; these are letters that appear to be composed of a jotation bar connected with a horizontal stroke to a following letter, and that are pronounced (at least in some positions in some early Slavic systems) as a jot (or English [y] sound) followed by the sound represented by the basic (unjotated) letter. For example, "ѣ" represents a [y] sound preceding the [a] sound represented by the letter "а". Because the jotation bar has constant semantics and is combined with several other letters, one might reasonably argue that it should be encoded as a separate character, but this approach is rejected for reasons discussed below⁴².

From an information interchange perspective, there is no difference between encoding jotated text elements as one character or as two. As was noted in § 3.4.1.1, above, there may be an information-processing advantage to encoding a text element that functions as an atomic unit as a single character. In the present case, it was decided that jotation should not be treated as an independent combining character for several reasons:

- (1) Jotation combines only with U+0454 (ѣ), U+0463 (ѣ), U+0430 (а), front and back nasals (U+0467

⁴² The jotation bar most likely originates palæographically from a separate *i*-letter, so that jotated letters are continuations of true ligatures or independent letters. The horizontal connecting stroke is occasionally omitted (Karskij 168); these alternate glyphs are not included in the current inventory because they are rare and essentially of palæographic interest, although they can be added should a need for them be felt.

“ӑ” and U+046B “ӓ”, respectively), and a variant front nasal not present in UCS (ӑ), which means that the number of potential combinations is considerably smaller than the number of combinations of base letters plus superscript marks. Additionally, UCS already contains four of these six combinations (U+0465 “ӕ”, U+044F “ӗ”, U+0469 “ӧ”, U+046D “ӧ̇”), which provides a precedent for encoding the remainder as unitary characters (so as to offer a uniform strategy for all jotation). Finally, because of the very small number of characters involved, extending this current strategy would not increase the size of the character set significantly.

- (2) The jotated letter U+044E (ӕ) represents a combination of jot plus the sound [u], although it is graphically more similar to a combination of jot plus the letter that traditionally represents the sound [o]. While decomposing other jotated vowels and encoding them as combinations of two characters might facilitate processing that depended on features of the sound system, encoding U+044E (ӕ) in this way would provide no such advantage. In other words, encoding jotation as a separate item does not provide a general way of encoding the phonetic composition (or any other property) of the letters in question.
- (3) Mediæval Slavic grammarians viewed jotated letters as independent letters of the alphabet, rather than as digraphs (see Worth for summaries and analyses of the major mediæval grammatical treatises, with bibliography for primary sources). Because much of the processing philologists will want to perform is based on letters of the alphabet as conceived by the scribes who produced our manuscripts, units used as independent letters are likely to be subject to the same set of processing operations, and encoding them all as characters, or atomic units, simplifies writing processing routines.
- (4) Rotated and mirror-image glyphs occur in early Cyrillic, but the order of Cyrillic letters is normally significant, so that a sequence two letters XY is not replaceable by YX. The existence of a reversed version of U+044E (ӕ) in some manuscripts suggests that this represents a mirror-image of a single grapheme,

rather than a metathesis of two independent graphemes⁴³.

- (5) Parallelism between early and modern Cyrillic also argues in favor of treating jotated letters as single characters, since modern U+044F (ӗ) and U+044E (ӕ), the only modern descendants of jotated letters, are already present in UCS as unitary characters.

One argument against encoding jotated letters individually is that this strategy precludes the easy addition of other jotated letters, should these be discovered. For example, there are several other variant nasal vowel letters that do not have attested jotated counterparts, as far as I have been able to determine, but the possibility that such jotated forms may eventually be discovered cannot be ruled out. I consider the likelihood of such a discovery remote enough that the advantages of encoding jotated letters as single characters outweighs the risk⁴⁴. Should a previously unattested jotated letter that fulfills the requirements for characterhood be discovered, the character standard would then require amendment⁴⁵.

The preceding discussion assumes that the character set should encode jotation in one way or another, and asks only whether it should be encoded as a single combining (prefix) character or as a set of composite jotated vowel characters that are parallel to nonjotated ones. In fact, the ability of jotation to discriminate sound or meaning depends both on the orthography of the individual manuscript and on the particular vowel with which the jotation is associated, and one might reasonably argue that text-processing operations traditionally performed

⁴³ Andrej Bojadžiev (personal communication) has drawn my attention to a theory that Cyrillic “ӕ” is ultimately derived from the Greek *oi* diphthong (although alternative explanations have also been proposed). This does not necessarily imply a metathesis of two letters from a Cyrillic perspective.

⁴⁴ Furthermore, jotated versions of other nasal vowel letters would probably not fulfill the definition of characterhood (cf. § 3.4.4.1, below).

⁴⁵ Alternatively, a separate jotation bar combining character could be added to serve to encode newly-discovered jotated forms, with its use deprecated where combined jotated forms already exist. The disadvantage of this strategy is that it increases the opportunity for multiple encodings of the same text element, thereby increasing the processing burden.

on the character level (comparison, sorting) would benefit from treating at least some jotated and nonjotated vowel letters, especially those involving front vowel phonemes, as allographs of one another.

This solution was not available on a general level because the UCS inventory already contains U+0465 (Ѡ), U+044F (Ѡ), U+0469 (Ѡ), and U+046D (Ѡ). As a result of this legacy, it seems most consistent to encode all jotation at the character level, even though it will usually prove advantageous to neutralize at least some jotated/nonjotated pairs during most content-based processing. Accordingly, the table at the end of the present article adds AFII 052/333-10923 (Ѡ) as a character. The jotated third jus (Ѡ) is treated as a glyphic variant of jotated U+0469 (Ѡ), for reasons discussed in § 3.4.4.1, below.

3.4.1.3. Jery

Jery (U+044B, “Ѡ”) is the only modern Slavic Cyrillic letter that is a structural digraph (consisting of two discrete pieces, neither of which is a superscript mark)⁴⁶. It is viewed as a single letter by both modern Slavic Cyrillic alphabets that include it (Russian, Belarusian), and according to the grammatical treatises mentioned above, also by mediæval Slavic grammarians. One final argument in favor of viewing modern U+044B (Ѡ) as an independent letter is that while its first part (Ѡ) does occur as a separate letter (U+044C), its second part does not (cf. the analysis of U+044E “Ѡ” in § 3.4.1.2, above)⁴⁷. But for reasons discussed below, early Cyrillic jery may sometimes be encoded as one character and sometimes as a sequence of two.

The early Cyrillic situation is complicated by several factors:

⁴⁶ One might reasonably consider modern Cyrillic U+045C (Ѡ), U+0453 (Ѡ), U+045E (Ѡ), U+0439 (Ѡ), U+0451 (Ѡ), U+0456 (Ѡ), and U+0457 (Ѡ) structural digraphs, but these all have a superscripted component. The argument is weaker for U+0458 (Ѡ), since the base part does not occur either in isolation (cf. Ѡ, Ѡ, Ѡ, Ѡ, Ѡ, Ѡ) or with different diacritics (cf. Ѡ, Ѡ).

⁴⁷ While modern U+044B (Ѡ) resembles a sequence of U+044C (Ѡ) and U+0456 (Ѡ), the latter always has a dot in modern Slavic Cyrillic, while the second component of U+044B (Ѡ) never has a dot. The sequence Ѡ in modern Belarusian differs from Ѡ; the former is two graphemes (soft sign plus vowel [i]), while the latter is one (vowel [y]).

- (1) While modern Cyrillic always constructs this character out of U+044C (Ѡ, a soft sign or front jery) with a vertical stroke, early Cyrillic uses both U+044C (Ѡ) and U+044A (Ѡ), front and back jerys, as the first part of jery, and the choice between the two jerys is often important for orthographic analysis. But because there is no difference in sound or meaning between the two ways of spelling jery, Unicode principles suggest that both spellings should be encoded identically on the character level. Furthermore, since UCS already includes a single-character encoding of this text element in modern Cyrillic, and since Unicode regards the relationship between early and modern Cyrillic as a font change, it seems most consistent to employ this same single character for the early Cyrillic counterpart.
- (2) The evidence of mediæval grammatical treatises again suggests that those who used early Cyrillic viewed jery, whatever its glyphic representation, as a single letter.
- (3) Early Slavic may also use any of several [i] type letters for the second component of jery. This variation is much more common in Glagolitic writing (where there are three basic [i] type letters) than in Cyrillic (where there are two), but even in the latter, a sequence like “ѠѠ” may represent either two sounds (“Ѡ” and “Ѡ”) or one (“Ѡ”).
- (4) Because the individual components of the different spellings of jery are also present in UCS as separate characters, there is no absolute impediment preventing scholars from decomposing jery into separate character sequences for data entry. It would be useful for analytical purposes to exploit the difference noted above by encoding the shape “Ѡ” as one character when it represents jery and as two characters when it represents a sequence of two vowels, but in some cases both readings may be possible.
- (5) In addition to having different first and second components, jery instances may vary according to whether the two parts are unconnected, connected by a horizontal stroke at the baseline, or connected by a horizontal stroke at the midpoint. The use of crossbars to connect the parts is of largely palæographic interest, and these variants are therefore not included in the

present inventory (cf. Karskij 203). As is noted above, this type of decision is subjective, and the glyph inventory can be expanded should a need for these additional variants arise.

The inventory below retains the single UCS jery character U+044B and the two existing jery glyph identifiers, AFII 052/332-10970 (Ѣ) and 047/155-10093 (Ѣ). Users who prefer to record early jery as a single character can do so; those who need to discriminate other glyphic variants can create additional SGML entities that map this character either to sequences of existing glyph identifiers (e.g., Ѣ + и) or to new composite glyphs. The graphically ambiguous text elements mentioned in item 4, above, require editorial decisions during text entry. Where there is true ambiguity, which cannot be resolved by context, editors should make a consistent decision and document it clearly (cf. § 3.3, above).

Users who prefer to record early jery as a sequence of a jer character plus an [i] character can do so using the existing mappings (supplemented as needed to represent the crossbars discussed in item 5, above). Unless the entity inventory is expanded, jery spelled as “Ѣи” will have to be encoded as two characters if the identity of the second component is to be retained, since my inventory currently contains no SGML entity that maps this shape to the U+044B jery character.

Ignoring the U+044B jery character in favor of encoding all instances of jery as sequences of two characters may be the most effective strategy, especially if data entry is to be performed by research assistants, who may lack the experience to deal effectively with ambiguities. Encoding all sequences of jer letter plus [i] letter this way removes responsibility for identifying and dealing with ambiguities from the data entry stage, and transfers it to the analysis stage, which has both practical and theoretical advantages⁴⁸.

⁴⁸ From a practical perspective, as is noted above, this separation permits data entry to be performed by less experienced or less knowledgeable persons. From a theoretical perspective, all transcription is inevitably interpretive, but it is nonetheless possible to minimize the degree of interpretation, treating transcription primarily as a reproduction of the information, including any original ambiguities, rather than as an analysis that seeks to resolve the ambiguities.

3.4.1.4. *I-type letters*

Modern Cyrillic writing includes U+0456 (і) and U+0457 (ї) letters; these are regarded by native grammarians as single letters of the alphabet, and are encoded as unit characters in modern character sets. The situation is more complicated in early Cyrillic because both single and double dot occur freely over a large number of base characters, which means that these nonspacing marks must also be included separately in the character set. Furthermore, modern Cyrillic typography regards AFII 047/107-10055 “Cyrillic capital letter I” (dotless) and AFII 047/167-10103 “Cyrillic small letter I” (dotted) as upper- and lower-case counterparts of the same letter. But the number of dots is variable in early Cyrillic writing, and usually conveys no informational difference. As far as sound and meaning are concerned, early Cyrillic “i” with zero, one, or two dots represents the same character. For reasons discussed below, the present inventory allows *i*-type letters to be encoded either as single characters or as sequences of spacing dotless character plus nonspacing superscript dots.

Ukrainian is the only modern Slavic language that includes both U+0456 (і) and U+0438 (и), and these letters have different sounds and distinguish meaning in Ukrainian. In early Cyrillic these text elements represent the same sound, but they have different numerical values, and can therefore represent different (numerical) meanings. This quasi-characterhood (present in numerical contexts and absent in alphabetic ones) imposes certain processing costs; for most textual or linguistic analysis of early Cyrillic electronic texts, U+0456 (і) and U+0438 (и) will require normalization, but this normalization cannot be automated completely in a plain-text environment, since numerals must be treated exceptionally⁴⁹.

⁴⁹ Both letters occurred in pre-revolutionary Russian orthography, where they conveyed the same sound, and their distribution was governed by their environment—except for one minimal pair: миръ ‘peace’ мѣръ ‘world’ (cf. also мѣро ‘myrrh’). Automated conflation of alphabetic “и” and “і” in early Cyrillic, without conflation of the same characters in a numerical context, can be achieved in an SGML environment by wrapping an appropriate SGML tag around numbers, such as the <num> tag in P3.

One additional complication within early Cyrillic is a third *i*-type letter: “ı”. This is not a naturally-occurring distinct text element of early (or modern) Cyrillic writing; it was introduced into Cyrillic typography as a way of representing a graphically distinctive third Glagolitic *i*-type letter when typesetting Glagolitic documents in Cyrillic transliteration. From the perspective of the productive Unicode definition of characterhood, this third *i*-type letter is not a separate character from “ı” even in Glagolitic, since it does not represent a distinctive sound or meaning⁵⁰.

Given the Unicode decision to treat early Cyrillic as a font change from modern Cyrillic, it would be impractical (even if structurally appropriate) to *require* that what looks like “ı” be encoded as a single character in modern texts and as a sequence of characters in early ones. As with Latin “á”, or early Cyrillic jery (§ 3.4.1.3), UCS provides more than one way of encoding “ı” and “ı̇”, with the decision among them left to the individual editor.

The present inventory does not create any new *i*-type characters beyond the three already present in UCS: U+0438 (и), U+0456 (“ı” or “ı̇”), and U+0457 (ї). It does, however, supply several new glyphs. As was noted above, U+0438 (“octal i”) may be represented either by slanted AFII 047/132-10074 (и) or by a straight “и”; intermediate glyphs may be created as needed. A new glyph for the superscript variant of U+0438 that looks like a double grave accent (¨) is also proposed. New “decimal i” glyphs for dotted and dotless shapes are mapped to U+0456, as is the Glagolitic “third i” (ı). For early Cyrillic, U+0456 should be considered unspecified with respect to superscript dots, but users who wish to encode dots as separate characters will probably want to regard U+0456 as dotless in early Cyrillic, even though it normally has a single dot (lower-case only) in modern Cyrillic.

⁵⁰ Within the basic Unicode strategy, transliteration between Cyrillic and Glagolitic should be regarded as a mapping between different character inventories, rather than as a font change. This type of transliteration is an established Slavistic tradition, so that Cyrillicized Glagolitic is culturally a Cyrillic writing system in its own right.

3.4.1.5. *ı̇ and ı̈*

The status of U+0439 (ı̇) and U+045E (ı̈) is comparable to that of U+0456 (ı) and U+0457 (ї). These items are traditionally treated as independent characters in modern Cyrillic and are available as such in the UCS Cyrillic inventory. But U+0306 (¨) combines freely with a large number of bases in early Cyrillic writing, and therefore also needs to be included in UCS as a separate nonspacing mark. This provides multiple ways of encoding “ı̇” and “ı̈”, with the decision in individual cases resting with the editor. The most natural approach would seem to be to use the separable non-spacing mark if the inventory of combinations is greater than that found in modern Slavic Cyrillic writing systems.

3.4.1.6. *Ligatures*

The inventories proposed here include no ligatures not already present in UCS. Ligatures are a common feature of early Cyrillic writing, and ligated glyphs will be needed for fine typographic reproduction, but ligatures do not fulfill the productive Unicode requirements for characterhood, and the process of ligation would in any case be too productive for it to be practical to encode the results as independent characters. This means that ligatures cannot be encoded in plain text. Fancy text can incorporate ligature tags, which can then be used by the rendering engine to provide a ligated glyph where necessary (and where available). The inventory of ligated glyphs could be quite large, and will probably need to be accumulated slowly, since it is impractical to try to envision all conceivable members of the set.

As was noted earlier, a common way of rendering English ligatures (e.g., “ff”) is for the rendering engine to ligate certain combinations automatically, while providing the user with a mechanism for blocking ligation in exceptional cases. This is appropriate for such English examples, since ligation (in fonts that support it) is the statistical norm for certain character sequences. Because ligation is exceptional for most early Cyrillic, the reverse fancy-text encoding scheme is more appropriate: ligation should take place only when explicitly encoded.

3.4.1.7. *Other composite characters*

The current UCS inventory includes U+0477 (ѣ), as a separate character from U+0475 (ѣ, ižica) with

superscript U+030F (ˆ, kendema, double grave). Other composite characters are U+047D (̃) and U+047F (̄). While there is some justification for these decisions (for example, the latter is listed as a separate letter of the alphabet in a modern Church Slavonic grammar used in Orthodox seminaries [Alipij 17]), the superscripted portions of all of these characters will need to be available as separate characters themselves, which means that the combined characters merely provide encoding alternatives⁵¹. If we were developing a general character set for early Cyrillic, it might be best not to include any of these composite text elements as individual coded characters, but it is not now possible to remove them from UCS.

The components of U+0477 (̂) and U+047D (̃) are available in plain text, which means that these text elements may be encoded in plain text either as composite character sequences or as precomposed (unitary) characters. U+047F (̄) can be encoded only as a precomposed (unitary) character in plain text; if it is to be encoded as a sequence of U+0461 (w) and U+0442 (т), this must be done in fancy text, with a tag indicating the superscription.

3.4.2. Superscript letters

Early Cyrillic writing is characterized by the widespread use of superscript letters. Because superscript letters have the same semantics, and essentially the same form, as nonsuperscript letters (but see below), I view the two as identical on both the character and the glyph levels⁵². This means that superscription must be encoded with a tag, which means that it can be used only in fancy text.

⁵¹ Superscript letters often have different forms than regular letters, but they nonetheless represent the same information as their nonsuperscript equivalents and are encoded using the same characters, with their shape and position normally indicated with tags in fancy text. The presence of these particular combinations in the UCS inventory means that they, but not other combinations, can also be encoded in plain text. Enriching plain text by including precombined characters is risky, because it is impractical — and perhaps impossible — to determine an exhaustive inventory. Additionally, processing routines will need to recognize the equivalence of, for example, omega plus superscript “т” encoded as two characters and as one.

⁵² This analysis depends on the font model described in § 2.2.3.1. More recent models would support the inclusion of in-line and superscript glyphs in the same font.

A character tagged as superscripted will normally be interpreted by the rendering engine by reducing its size and centering it above the preceding character. In many cases, a superscript tag should also be interpreted by the rendering engine as a request to change glyph (e.g., superscript U+0445 [x] might be wider and flatter than regular U+0445). This is especially clear with variant superscript letterforms, such as “recumbant r” (ꝛ), which may occur in the same font a regular superscript “r”. Multiple superscription (a superscript letter above a superscript letter above a base letter) is rare, but it does occur, and encoding superscription as a fancy-text tag enables the user to record such forms simply by nesting tags.

Superscript letters are entered into the backing store following the letters over which they will be rendered for two reasons:

- (1) They are most commonly intended to be read in this order.
- (2) This ordering corresponds to the Unicode system of encoding nonspacing diacritics, and observing it here will provide a uniform protocol for entering superscripted material of any kind.

3.4.3. Modern and early Cyrillic

As is noted in § 2.2.3.2, above, Unicode regards early Cyrillic as a font change from modern Cyrillic, which is consistent with the notion that Unicode is an encoding system for scripts. If modern Macedonian and Ukrainian are combined in a single Cyrillic script, there can be no justification for not combining early and modern Cyrillic similarly. But while the appropriateness of such unification seems unassailable, at least in the context of a script-encoding architecture, its implementation leads to a small number of difficult situations that require special attention.

3.4.3.1. Case

Upper and lower case is not a significant distinction for much early Cyrillic writing. Large or decorative letters were available for special purposes (headings, the mediæval equivalent of drop caps, etc.), but these uses are not fully analogous to the modern distinction between upper and lower case,

and might better be compared to changes in typeface or the use of swash variants.

Modern case distinctions are nonetheless maintained in the early Cyrillic character inventory for two reasons. First, such distinctions are common in normalized early Cyrillic printed editions, and although case may not have functioned in any significant way for many mediæval Slavic scribes, it is a relevant category for modern Slavic philologists. Second, because UCS regards Cyrillic as a single script, and because case distinctions are needed for modern Cyrillic and are also already present for some early Cyrillic characters, it seemed most consistent to maintain case distinctions throughout the Cyrillic inventory, except for a small number of early Cyrillic items that are essentially ambiguous with respect to case.

3.4.3.2. Alphabetic order

It is unclear to what extent alphabetic order can be considered a meaningful concept for early Cyrillic writing. Different mediæval grammatical treatises are not entirely consistent in the letters they list and the order in which they list them, and with the exception of alphabetic acrostic texts and the use of letters to represent numerical values, no source explicitly addresses or employs anything comparable to our modern concept of alphabetical order as a means of organizing data.

The order in which letters of the alphabet are listed in original mediæval materials shows some dependence on two related principles: Greek alphabetic order and Cyrillic numerical value. But these principles are not observed strictly, and cannot in any case be applied to those Cyrillic letters that have no Greek equivalent or Cyrillic numerical value. Modern handbooks of Old Church Slavonic follow these same principles, but may also observe features of modern Cyrillic alphabetic order. Furthermore, many handbooks of Old Church Slavonic present Glagolitic and Cyrillic contrastively, sometimes arranging the Glagolitic letters in Glagolitic numerical order and then listing their Cyrillic counterparts alongside them in that same Glagolitic order. In general, different handbooks may follow different orders for what they regard as letters of the alphabet, and there is no consensus at all about a standard order for glyphic variants.

The present study follows an original order that is partially derived from existing references and partially arbitrary⁵³.

3.4.3.3. я, ѣ, and ѡ

The UCS inventory includes U+044F (imaged as modern Cyrillic "я" in *Unicode*) and U+0467 (ѡ), but nothing imaged as early Cyrillic "ѣ"⁵⁴. This absence of a separate "ѣ" character invites us to treat this item as a font change variant of modern "я", which is the solution adopted in this paper.

From a palæographic perspective, however, modern "я" is descended not from early "ѣ", but from early "ѡ", and since early and modern Cyrillic are regarded as a font change, it might appear palæographically more sensible to unify modern "я" and early "ѡ", while assigning early "ѣ" to a separate character cell. But two other perspectives are also relevant, one of which overrides palæographic considerations.

Since the two early Cyrillic letters represented the same sound in early East Slavic writing, and this sound is comparable to that of modern "я", from a Russian or Ukrainian or Belarusian perspective one could have unified either of the early letters with modern "я" and assigned the remaining early letter to a separate character cell. From an East Slavic phonetic perspective, then, there is no reason to prefer one decision over the other.

In early Cyrillic writing from South Slavic areas, however, only "ѣ" was pronounced like modern "я", while "ѡ" was pronounced quite differently. Thus, the palæographic evidence favors "я" = "ѡ", the South Slavic linguistic evidence favors "я" = "ѣ", and the East Slavic linguistic evidence is ambivalent. The South Slavic linguistic evidence is controlling because of the basic Unicode principle that differences in sound or meaning are more important than differences in form for identifying characters.

⁵³ As was noted in § 2.2.4.2, above, the order of presentation in the present work is not a proposal for the order of characters or glyphs within their respective sets.

⁵⁴ Note that the images are intended to be illustrative. Characters are defined by the distinctions in sound and meaning that they represent, irrespective of their physical appearance.

3.4.3.4. U-type letters

UCS contains two early Cyrillic letterforms that corresponds to the sound [u]: U+0443 (early "Ѹ" or modern "y") and U+0479 (ѸѸ). Although these are essentially allographs in most early Cyrillic writing, in the earliest documents U+0443 (Ѹ) corresponds in sound and meaning not to U+0479 (ѸѸ), but to U+0475 (Ѹ). Accordingly, U+0475 (Ѹ), U+0479 (ѸѸ), and U+0443 (Ѹ) are properly encoded as different characters, with the latter character merged during processing with one or the other of the former two, depending on the orthography of a particular source. U+0479 (ѸѸ) is a composite character and is equivalent to a sequence of U+043E (Ѹ) and U+0443 (Ѹ).

There is one other common related letterform traditionally distinguished in orthographic analysis, AFII 052/326-10966 "Ѹ", which is not represented in UCS⁵⁵. This is included in the present inventory as an alternative glyphic representation of U+0479 (ѸѸ). A superscript *u*-type glyph that resembles a breve (˘) is mapped to U+0442 (Ѹ).

3.4.3.5. Jers

Back jer (U+044A "Ѹ") and front jer (U+044C "Ѹ") are distinctive graphemes in modern Cyrillic and in much early Cyrillic writing, and are represented as separate characters in UCS. In some other early Cyrillic, however, they function as positional variants, which do not distinguish sound or meaning, and the distinction between them must therefore be neutralized during processing for such manuscripts. Furthermore, even in those early manuscripts where back and front jer normally do function as different characters, the use of the wrong jer letter (from an etymological perspective) is one of the most common deviations from canonic Old Church Slavonic norms. This means that character-based processes, such as indexing, may benefit from neutralizing the difference between the two original jer letters even in those manuscripts where their distribution is not fundamentally allographic.

⁵⁵ Historically, the spelling "ѸѸ" is based on the Greek digraph "ου" (phonetic [u]). The form "Ѹ" reflects a ligature, also found in Greek, in which the second component (epsilon) is written atop the first (omicron). Greek "ου" outside of the digraph in question was probably pronounced [ū] when Cyrillic was first elaborated.

Middle Balkan Slavic writing also knows an intermediate, or neutral, jer (Ѹ). Because neutral jer occurs in writing where the distinction between front and back jer letters is not associated with a difference in sound or meaning, one can regard the relevant writing systems as containing only one functional jer character. They are nonetheless encoded with two characters, since U+044A (Ѹ) and U+044C (Ѹ) are part of the script-based character inventory, but there was no clear justification for creating a third character-level item to represent neutral jer (Ѹ). Accordingly, the latter glyph is arbitrarily mapped to U+044C (Ѹ).

Despite the fact that it is written above the baseline (either between letters or above a letter), paerok (U+033E, AFII 052/317-10959, "˘") is functionally and alphabetic (rather than diacritic or accentual) character, since it essentially replaces jer letters⁵⁶. Erik (AFII 052/320-10960 "˘") does not represent a different sound or meaning than paerok, and can be treated as a glyphic variant of the latter.

3.4.4. Other special problems of early Cyrillic

3.4.4.1. Nasal vowels

The normalized Old Church Slavonic Cyrillic orthography found in most handbooks and grammars uses four nasal vowel letters, characterized by the linguistic parameters front / back and nonjotated / jotated:

- (1) U+0467 (Ѹ): front nonjotated nasal vowel (/ɛ̃/)
- (2) U+0469 (Ѹ): front jotated nasal vowel (/jɛ̃/)
- (3) U+046B (Ѹ): back nonjotated nasal vowel (/ɔ̃/)
- (4) U+046D (Ѹ): back jotated nasal vowel (/jɔ̃/)

⁵⁶ U+033E (˘) may be written after a consonant even where there is no etymological jer, but the same is true of jer letters. That is, in certain writing systems both jer letters, paerok, and erik all occur in the same positions and with the same function. While paerok and erik may be written above a letter, they are primarily spacing characters, rather than nonspacing superscripts. The identification of U+033E (˘) as NON-SPACING VERTICAL TILDE in UCS is erroneous, since paerok is fundamentally a spacing character. The Unicode alias CYRILLIC PALATALIZATION is similarly erroneous, since paerok may replace both non-palatalizing U+044A (Ѹ) and palatalizing U+044C (Ѹ), and may also appear where no jer letter is justified etymologically or linguistically. Palaeographically, U+033E (˘) is descended from U+044A (Ѹ).

For reasons discussed below, one further nasal vowel character is required.

Regardless of the spelling, the distribution of jotted and nonjotted front nasal letterforms in the canonic Cyrillic Old Church Slavonic manuscripts is that the nonjotted letterform (U+0467 "␣" in the system above) is written after consonants and the jotted form (U+0469 "␣̣" in the preceding system) is written elsewhere. Spellings like *зѣмѣа* (with U+0469 "␣̣" after a consonant) in normalized reproductions of Old Church Slavonic manuscripts are based on Russian Church Slavonic orthography. With a small number of exceptions, such spellings do not occur in the Old Church Slavonic canon (Lunt 21-22)⁵⁷.

The current UCS inventory of nasal vowel letterforms is based on Old Church Slavonic handbooks, rather than primary sources, and several other nasal vowel letterforms (␣, ␣̣, ␣̂, ␣̃, ␣̄, ␣̅) also require consideration. The following distributions of front nasal vowel letterforms can be observed (Trubetzkoy 40, Diels 31):

Manuscript	nonjotted	jotted
<i>Savvina Kniga</i>	␣, ␣̣ (rarely)	␣̣̣
<i>Zograph Folia</i>	␣̣̣	␣̣̣̣
<i>Suprashiensis Sluck Psalter</i>	␣̣̣̣	␣̣̣̣̣
<i>Hilandar Folia</i>	␣̣̣̣̣	␣̣̣̣̣̣
<i>Ostromir Gospel</i>	␣̣̣̣̣̣	␣̣̣̣̣̣̣
Preslav ceramic inscription ⁵⁸	␣̣̣̣̣̣̣	␣̣̣̣̣̣̣̣

Table 1. Distribution of front nasal vowel letterforms

Additionally, other manuscripts use only one front nasal vowel letter both after consonants and elsewhere: U+0467 (␣) in the *Undol'skij Folia* and "␣" in the *Cyrillic Macedonian Folium* (with two examples of U+0467 "␣", both after consonants). The form "␣" may represent either etymological front or back nasal vowels, and is treated as a variant of U+0467 (␣) here primarily for palaeographic reasons (Karskij 208).

⁵⁷ The same obtains for U+0454 (␣̂) and U+0465 (␣̃). In Old Church Slavonic, the former indicates the sound /e/ or letter "ѣ" after a consonant, while the latter indicates that the preceding letter is not a consonant.

⁵⁸ Andrej Bojadžiev, personal communication.

The present inventory retains the two UCS front nasal vowel characters and proposes the addition of "␣" as a third character, with the remaining letterforms mapped as follows:

Character	Letterforms
␣ (new)	␣, ␣̣
U+0467 (␣̣̣)	␣̣̣, ␣̣̣̣, ␣̣̣̣̣
U+0469 (␣̣̣̣̣)	␣̣̣̣̣, ␣̣̣̣̣̣

Table 2. Front nasal vowel characters and glyphs

This arrangement makes it possible to represent all of the opinions in Table 1 on the character level⁵⁹.

Nasal vowel letters in South Slavic and East Slavic manuscripts after the Old Church Slavonic period do not represent nasal vowel sounds; for example, East Slavic U+0467 (␣̣̣) (= AFII 052/337-10975 "␣̣̣") indicates the sound [ja] (otherwise associated with U+044F "␣̣̣̣" [= AFII 052/335-10973 "␣̣̣̣̣"]). That these neutralizations will need to be performed at the processing, rather than encoding, level is a consequence of the script-based nature of UCS.

Finally, a blended jus (␣̅) is widespread in Middle Bulgarian manuscripts, where confusion of U+0467 (␣̣̣) and U+046B (␣̅) is characteristic. As far as I have been able to determine, the letterform "␣̅" does not represent a different sound or meaning from the letterform AFII 047/325-10197 (␣̅) in those manuscripts in which it occurs. In those manuscripts, either AFII 047/325-10197 (␣̅) is completely lacking, or it is used where either etymological front nasal or etymological back nasal (jotted and nonjotted) are expected, much like the blended jus. Because neither front / back nor jotation is a significant feature of nasal vowel letters in these manuscripts, I have treated the blended jus ("␣̅") as a glyphic variant of U+046B (␣̅)⁶⁰.

⁵⁹ Note that the character U+0467 (␣̣̣) may represent a jotted front nasal vowel sound in some sources, a nonjotted sound in others, and nonnasal sounds in others. Because differences between jotted and nonjotted front vowel letters are very rarely significant for lexical or textual analysis, the simplest method of comparing manuscripts may involve conflating all front nasal vowel letters.

⁶⁰ See Karskij 167 and 208-09 for details.

3.4.4.2. O-type letters

UCS distinguishes several early Cyrillic *o*-type letters, although only two, U+043E (o) and U+0461 (w) satisfy the productive definition of characterhood (because they represent different numerical values). Other variants, such as U+047B (o), do not normally function as distinctive characters, although their distribution may be governed by orthographic norms. These three letterforms sometimes do reflect differences in pronunciation (usually in accentuation or in vowel height), but the fact that these differences are not normally relevant to lexical or textual analysis (due to the very small number of potential minimal pairs), plus the enormous variety in their use (Stadnikova), suggests that they cannot easily be assigned stable and independent sound or meaning. For these reasons I have treated all additional [o] letterforms as instances of basic U+043E "o"⁶¹. Because the differences among them will be encoded at the SGML entity and glyph levels, this information is accessible, should it prove necessary for analytical purposes.

3.4.4.3. E-type letters

There are four *e*-type letters in UCS, all of which are needed as distinct characters for encoding modern Cyrillic materials: U+0435 (e), U+0454 (e), U+044D (e), and U+0451 (e)⁶². Early Cyrillic also includes U+0465 (e), and it requires several other letterforms for purposes of orthographic analysis. In some instances, variant *e*-type letterforms may reflect differences in sound (such as the use of broad "e", anchor "e", or epsilon "e" as replacements for jotted U+0465 (e) in some manuscripts), although this usage is inconsistent (in some manuscript epsilon "e" may be the only *e*-type letter), and in any case

⁶¹ Most interesting are the various ocular "o" letters, used in the root for "eye", with one (o), two (o, oo), or many dots (ooo), depending on whether the wordform is singular, dual, or used in the epithet "many-eyed".

⁶² U+0451 (e) is sometimes spelled "e" and sometimes "e" in modern Russian. Encoding can either transcribe the spelling that occurs in the original source (reflecting a neutralization found in standard Russian orthography) or restore the spelling "e" where it is justified by sound and meaning. The former is more faithful to the distinction between transcription and interpretation (although this distinction is never absolute), while the latter may simplify subsequent processing.

these differences are not normally important for lexical or textual analysis.

These letterforms are largely positionally determined, and the most efficient encoding strategy is to register all of them as independent glyphs, associated with basic early Cyrillic U+0454 (e). (Large "e" [Karskij 186] does not require registration as a separate glyph, since it can be encoded as a regular upper-case form). As with *o*-type letters, the distinctions are available at the SGML entity and glyph levels; for example, it should be possible to create an ad hoc association of the epsilon "e" glyph with U+0465 (e), rather than U+0454 (e), should that be required. Reversed *e*-type letterforms, such as tilted "e", are treated as variants of U+044D ("e").

3.4.5. Numbers

Numerical function is often indicated by surrounding the letter (e.g., U+0432 "R") with periods (R.) or middle dots (R·), by crowning it with titlo (R̄), or by a combination of the preceding, but none of these features is obligatory. SGML provides a method for marking numbers by wrapping them in markup tags, such as <num>R</num>, which would enable an application to identify numbers and treat them differently than letters used in nonnumerical functions.

This situation differs from that of the quadratic letter form discussed in § 2.2.7.3, above, in which the glyph "u" sometimes corresponds to U+0432 (R) and sometimes to U+0434 (A). In the latter case, a single glyph corresponds to two different characters, but in the case of numbers, the text element in question is most properly considered both the same glyph and the same character whether it represents alphabetic or numerical meaning. Since characters do not represent specific meaning, but instead serve as discriminators of meaning, there is no inherent justification for treating "R" with numerical function as a different character than "R" with alphabetic function. Markup provides a way of encoding the different functions of this single character without impinging on the integrity of the glyph or character inventories.

3.4.6. Glagolitic

Unicode regards Glagolitic and Cyrillic as different scripts, which means they are encoded with

different characters, rather than treated as different fonts (*Unicode 45*). This decision is based on several factors:

- (1) Glagolitic and Cyrillic look very different. Treating them as different subsets of Unicode, rather than as font changes applied to a common inventory of characters, is comparable to treating Latin-alphabet Serbocroatian and Cyrillic-alphabet Serbocroatian as different character codings, rather than different fonts.
- (2) Glagolitic and Cyrillic began with roughly comparable inventories (with a small but significant number of exceptions), but Cyrillic has been extended considerably as a means of representing both Slavic and non-Slavic languages. Changing from one Latin font to another is normally possible irrespective of the language being represented, while changing from Cyrillic to Glagolitic would not be considered merely a font change by most persons who use the former.
- (3) *Unicode* does not raise this argument, but Cyrillic and Glagolitic letters had both alphabetic and — in most cases — numeric values, and the numeric values of alphabetically identical Cyrillic and Glagolitic units do not coincide. This means that when the signs in question are applied to the same linguistic material, their information content is different, suggesting that they represent different scripts.

One engineering argument in favor of treating Cyrillic and Glagolitic as the application of different fonts to a common character inventory is that transliteration then becomes largely a simple font change, rather than a reencoding (Miklas 1993:9). Additionally, a shared encoding would simplify processing Cyrillic and Glagolitic texts together, such as for building a word list based on a mixed Cyrillic and Glagolitic corpus. Although these arguments have merit, they are considerably weakened by the fact that neither culturally correct transliteration nor joint processing of Glagolitic and Cyrillic electronic texts could be reduced entirely to the level of font change, in the former case because of differences in the numeric value of the letters and in the latter case because of the lack of complete isomorphism

between the two alphabets. Thus, at least a minimal filtering mechanism would always be required for transliteration and joint processing.

3.4.7. Greek

The Cyrillic alphabet was based on Greek, and many basic letterforms in early Cyrillic manuscripts are similar to those of contemporaneous Greek writing. Additionally, early Cyrillic included a small number of Greek letters that at first were used almost exclusively in Greek words (e.g., U+0471 [Ψ], U+046F [ξ]), but that are nonetheless traditionally considered part of the early Cyrillic alphabet. Note that the preceding are Cyrillic characters in UCS, and are encoded differently from their Greek counterparts.

Later Cyrillic writing adopted a number of additional Greek letterforms (e.g., “*α*”, “*ε*”), which may be used either as basic shapes or as variants alongside more traditional Cyrillic shapes. The coexistence of “*α*” and “*α*” or “*ε*” or “*ε*” cannot be encoded in plain text, since the differences among these items do not satisfy the productive Unicode requirements for characterhood, but these Greek forms are included separately in the glyph inventory. As far as I have been able to determine, the difference between “*α*” and “*α*” never represents a difference in sound or meaning. This difference is thus primarily palaeographic, and the different glyphs are included as a convenience to font designers, rather than because this distribution is likely to be an object of study for anyone other than palaeographers. The more complicated distribution of “*ε*” and “*ε*” was discussed in § 3.4.4.3, above.

3.5. Early Cyrillic microsystems

As was discussed in § 2.2.2.3, above, early Cyrillic is functionally not a consistent semiotic system, but a set of semiotic microsystems based on a shared inventory of signs. Local modifications within individual documents take two basic forms: the neutralization of certain character distinctions (where different UCS characters do not discriminate sound or meaning in a particular source) and the “promotion” of certain glyph distinctions to character distinctions in others (where different letterforms that normally do not discriminate sound or meaning do

so in a particular source). It is sensible to store this information with the document, where it can be accessed by a processing application without user intervention.

TEI-conformant documents are required to have a TEI header, which travels with the document and is designed to store, among other things, information about encoding schemes. One practical solution to the problem described above is to incorporate the necessary information into the TEI header in a structured format that would be available to text processing routines. An alternative solution might be to construct a separate WSD for each document, and to encode the information in question there. While the original discussion of the WSD in § 2.2.7.2, above, was based on an assumption that a single WSD would underly transcriptions of all early Cyrillic writing, the present discussion suggests instead that each early Cyrillic monument could be regarded as a separate writing system (or set of writing systems), with each such system documented in a separate WSD.

The most complex aspect of managing the variety of attested systems is dealing with multiple systems simultaneously, such as when assembling a common dictionary for a set of manuscripts or when comparing variants to prepare a critical edition. The range of neutralizations affecting vowel letters in post-OCS manuscripts is extremely broad; for example, Old Church Slavonic U+0467 (Ѡ) may correspond to U+044F (Ѡ) in East Slavic, to U+0454 (Ѡ) in Serbian, and to U+046B (Ѡ) in Middle Bulgarian. This means that East Slavic U+044F (Ѡ) may correspond to Serbian Church Slavonic U+0454 (Ѡ), but only when they both correspond to Old Church Slavonic U+0467 (Ѡ), and East Slavic U+044F (Ѡ) that does not correspond to Old Church Slavonic U+0467 (Ѡ) will not correspond to Serbian Church Slavonic U+0454 (Ѡ). It is clear that simple global conflation of vowel characters across a corpus may lead to false matches; if all instances of U+044F (Ѡ) are conflated with U+0467 (Ѡ) in an East Slavic document, it becomes easier to process the document in isolation, but impossible to collate it effectively against a Serbian Church Slavonic copy. Furthermore, the environment in which a character occurs is also relevant, so that, for example, U+044E (Ѡ) and U+0479 (Ѡ) are interchangeable in all early Cyrillic writing

after palatal consonants, such as U+0448 (Ѡ), but not after nonpalatal consonants⁶³.

4. A proposal for a standardized approach to encoding early Cyrillic texts

4.1. Compatibility with existing early Cyrillic character inventories

One of the most important considerations in developing a new character set is the preservation of legacy data, which means ensuring backward compatibility with existing standardized sets. In practice, this means that a character present in a previous set should be omitted only in one of two situations:

- (1) The character is spurious, i.e., is designed to represent information that in fact will never occur. Removing such a character should not cause data conversion problems, since there should be no existing encoded data that uses this erroneous character.
- (2) The new character set provides an alternative method of encoding the same information. For example, from a technical standpoint, precomposed characters (base plus nonspacing mark) from old character sets do not need to be retained as precomposed units in a new character set, as long as the component parts are available to represent the same information.

The character set proposed below includes all UCS early Slavic Cyrillic characters, and therefore all characters from ISO DIS 6861.2 (apparently used in the development of Unicode). Certain characters from RKA-2 have not been encoded when their omission is supported by one of the two principles noted above.

⁶³ Similar problems occur with abbreviations, superscription, and other variations in spelling. It is likely to prove impossible to automate textual comparisons completely, except by normalizing texts comprehensively during input. The most effective way to maintain a maximal distinction between data and interpretation is to encode this normalization in markup, rather than content. In any case, computer-assisted lexical or textual analysis may not provide a fully automated solution, and may require human intervention, but it should nonetheless afford significant advantages.

4.1.1. The Mašinnyj fond

RKA-2 contains several characters whose existence is undocumented in standard palaeography manuals, and which accordingly are not included in the inventory proposed below⁶⁴:

- (1) Soft *k* "kako mjagkoe"
- (2) Soft *g* "glagol' mjagkaja"
- (3) Soft *x* "xer mjagkaja"
- (4) Soft *r* "rcy mjagkoe"

Soft *k*, *g*, *x*, and *r* may be spelled as regular U+043A (к), U+0433 (р), U+0445 (х), and U+0440 (р), respectively, with a superscripted U+0484 (҃), and are not therefore needed as precomposed (unitary) characters⁶⁵. They differ from soft *l* and soft *n* in that the latter two sounds may be spelled in manuscript sources either as U+043B (л) and U+043D (н), respectively, with superscripted U+0484 (҃), or by means of ligation with a PALATAL HOOK, as AFII 052/325-10965 "л҃" and AFII 052/356-10222 "н҃", respectively. Since Unicode already provides U+0459 (љ) and U+045A (њ) for encoding palatal *l* and *n*, respectively, in modern Cyrillic writing, these same characters can also be employed to represent early Cyrillic AFII 052/325-10965 "л҃" and AFII 052/356-10222 "н҃", respectively. This would be a font difference; a modern Cyrillic font would use the modern glyphs, as imaged in *Unicode*, while an early Cyrillic font would use the early glyphs. There are no comparable unitary representations of soft *k*, *g*, *x*, or *r* in early or modern Cyrillic writing⁶⁶.

⁶⁴ Letter names are reproduced as they appear in Andrijuščenko.

⁶⁵ The palatalization mark U+0484 (҃) occasionally occurs over other consonants, as well (Leskien § 3).

⁶⁶ While the distinction between palatal and nonpalatal consonants does discriminate meaning (in a very small number of situations, in practice), the palatal feature is marked inconsistently in the sources, with ambiguities resolved by content. This means that a character-level (lexical, grammatical, or textological) analysis that depends on the presence of a linguistic palatal feature cannot be automated, since this linguistic feature is often present even in the absence of any graphic indicator of palatalization.

Because UCS already includes palatal /n/ and palatal /l/ as characters, I have proposed adding the other two attested early Cyrillic text elements with a palatalization hook (palatal /d/ "л҃" and palatal /m/ "л҃") to the character inventory. This solution has the advantage of treating the palatal hook uniformly.

The following RKA-2 characters are functionally glyphic variants of other characters, and should not be distinguished in a character set:

- (1) Hard and soft *n* that look like "н" and "н" are encoded differently in RKA-2. The character set inventory proposed here makes no statement about the angle of the cross stroke. The two extreme glyph images (completely horizontal and completely slanted cross stroke) are included in the glyph inventory; intermediate forms may be associated with one or the other, or additional glyphs may be registered.
- (2) *l* that looks like "л" is encoded differently from *i* that looks like "л". As in the preceding situation, this is at most a glyphic difference, and both extreme shapes are encoded as a single character but different glyphs in the sets proposed here. The classification of intermediate shapes is not specified.

4.1.2. ISO DIS 6861.2

ISO DIS 6861.2 includes as an independent character "tuerde-titlo", a short three-legged "m", apparently intended to render superscript U+0442 (т). Because superscript letters are most appropriately considered variants of basic (nonsuperscript) characters, this item has not been included in the present inventory as an independent character. It is also not included as an independent glyph, since it seems more appropriate to treat it as a variant glyph instance of a basic three-legged nonsuperscript "m" glyph, at least within the model underlying the AFII inventory. The composite character U+047F (т̄) is included because it is already present in UCS, but superscript U+0442 (т) may occur over base letters other than U+0461 (w), and U+047F (т̄) might more properly be encoded as a sequence of base U+0461 (w) plus superscript U+0442 (т) (imaged as "т" where appropriate) in fancy text.

4.2. Inventory

The following inventory lists proposed early Cyrillic characters, glyphs, and SGML entities. UCS numbers are taken from *Unicode*, UCS names are taken from ISO/IEC 10646-1: 1993; most, but not all, of these names are identical to those in *Unicode*. AFII

numbers (in the form REFERENCE NUMBER-GLYPH NUMBER) and glyph descriptions are taken from *Register*, and are intended to serve as entity descriptions, as well. Entity names are my own. Shaded areas represent omissions from the current UCS and AFII inventories, which should be rectified. I have left the UCS and AFII number fields for these items blank, but I have proposed new UCS names and AFII glyph descriptions according to the naming strategies otherwise employed in these published inventories⁶⁷. Note that these naming strategies are not necessary consistent internally, with each other, or with traditional names in Slavic philology, but it is too late to propose broad renaming of large established inventories, which means that there is no opportunity to introduce more sensible systems of names.

Modern Cyrillic characters that are not used in early Cyrillic writing have been omitted from the following inventory. The present inventory does not

include nonspacing marks used in hymnographic and musical manuscripts, which will be discussed separately, after further study. It should be recalled that the images listed in the table are generalized, and Cyrillic characters may be mapped to different fonts to achieve different representations.

Patent errors in the existing UCS and AFII inventories, as well as proposed replacement AFII glyph descriptions, are discussed in footnotes⁶⁸. For backward compatibility, it is suggested that existing inappropriate or erroneous AFII glyph descriptions be retained as cross-references to new descriptions, but that the more informative naming strategy proposed here be employed for any future expansion of the inventory. UCS names have not been changed. Proposed SGML entity names are case sensitive and have a maximum length of eight, including an obligatory trailing -os (for "Old Slavonic"), in conformity with the general naming strategy for ISO registered standard entity sets.

⁶⁷ AFII traditionally distinguishes multiple variants merely by ordinal number, e.g., "Cyrillic capital letter X, second alternate". Wherever possible, I have replaced the numbers with descriptive adjectives, which supply more useful information than the numerical alternative and make the inventory more intuitive and accessible. Also wherever possible, I have moved these adjectives in front of the basic glyph name, e.g., favoring "Cyrillic capital letter TILTED X" over "Cyrillic capital letter x, tilted variant". The current AFII Cyrillic inventory is inconsistent in the placement of modifiers.

⁶⁸ *Register* is inconsistent in its use of capitalization; cf. 047/077-10047 "Cyrillic capital REVERSE E" (upper case "REVERSE") but 052/256-10926 "Cyrillic capital letter anchor E (early)" (lower case "anchor"). Because case is not significant in AFII names (unlike in entity names), I have silently normalized the usage by capitalizing descriptive terms following the word "letter" that are part the glyph name. I have also silently added "early" where it was overlooked in the current inventory.

Character, Glyph, and Entity Table

Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
А	0410	CYRILLIC CAPITAL LETTER A	052/301-10945	Cyrillic capital letter A, alternate (early) ¹	Aos	1
а	0430	CYRILLIC SMALL LETTER A	052/361-10993	Cyrillic small letter A, alternate (early)	aos	1
Ѡ		= 0410		Cyrillic capital letter ALPHA (early) ²	Alos	1
α		= 0430		Cyrillic capital letter ALPHA (early) <i>small</i>	alos	1
Б	0411	CYRILLIC CAPITAL LETTER BE	047/042-10018	Cyrillic capital letter BE	Bos	
б	0431	CYRILLIC SMALL LETTER BE		Cyrillic small letter BE, alternate (early) ³	bos	
Ѣ		= 0411		Cyrillic capital letter TILTED BE (early) ⁴	Blos	
ѣ		= 0431		Cyrillic small letter TILTED BE (early)	blos	
В	0412	CYRILLIC CAPITAL LETTER VE	047/043-10019	Cyrillic capital letter VE	Vos	2
в	0432	CYRILLIC SMALL LETTER VE	047/123-10067	Cyrillic small letter VE	vos	2
Ѧ		= 0412		Cyrillic capital letter LOOPY VE (early) ⁵	Vlos	2
ѧ		= 0432		Cyrillic small letter LOOPY VE (early)	vlos	2
Г	0413	CYRILLIC CAPITAL LETTER GHE	047/044-10020	Cyrillic capital letter GHE	Gos	3
г	0433	CYRILLIC SMALL LETTER GHE	047/124-10068	Cyrillic small letter GHE	gos	3
Ґ	0490	CYRILLIC CAPITAL LETTER GHE WITH UPTURN	047/102-10050	Cyrillic capital letter HARD G	HARDGos	3
г	0491	CYRILLIC SMALL LETTER GHE WITH UPTURN	047/162-10098	Cyrillic small letter HARD G	hardgos	
Д	0414	CYRILLIC CAPITAL LETTER DE	047/045-10021	Cyrillic capital letter DE	Dos	4
д	0434	CYRILLIC SMALL LETTER DE	047/125-10069	Cyrillic small letter DE	dos	4
Д		CYRILLIC CAPITAL LETTER SOFT D ⁶	052/116-10830	Cyrillic capital letter SOFT D (early)	SOFTDos	
д		CYRILLIC SMALL LETTER SOFT D	052/176-10878	Cyrillic small letter SOFT D (early)	softdos	
Є	0404	CYRILLIC CAPITAL LETTER UKRAINIAN IE	047/105-10053	Cyrillic capital letter YE	Eos	5

¹ Distinguished by a slanted stroke that runs from upper left to lower right, with a loop attached to its lower left side. It does not resemble uppercase 047/041-10017 (А) at all. Its lower-case counterpart is distinct from lower-case 047/121-10065 (а), which has a nearly vertical stroke on the right side (cf. the clearly slanted stroke in the early variant) and a marked curve at the top of this stroke over to the left (which may be absent or minimal in the early variant).

² Both upper- and lower-case Cyrillic alphas are distinguished by resembling lower-case Greek-alpha.

³ Only the lower case image of this early Cyrillic letter requires an alternate glyph. Its distinctive property is that it is identical in shape to its upper-case counterpart.

⁴ Distinguished from its untilted variant by being rotated ninety degrees clockwise.

⁵ Distinguished by completely rounded loops and a rounded left side, with no straight lines.

⁶ Golysenko 42.

Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
Є	0454	CYRILLIC SMALL LETTER UKRAINIAN IE	047/165-10101	Cyrillic small letter YE	eos	5
Є		= 0404		Cyrillic capital letter BROAD E (early) ⁷	E1os	5
є		= 0454		Cyrillic small letter BROAD E (early)	e1os	5
Є		= 0404	052/256-10926	Cyrillic capital letter ANCHOR E (early) ⁸	E2os	5
є		= 0454	052/336-10974	Cyrillic small letter ANCHOR E (early)	e2os	5
Є		= 0404		Cyrillic capital letter TILTED E (early) ⁹	E3os	5
є		= 0454		Cyrillic small letter TILTED E (early)	e3os	5
Є		= 0404		Cyrillic capital letter EPSILON (early) ¹⁰	E4os	5
є		= 0454		Cyrillic small letter EPSILON (early)	e4os	5
Ә	042D	CYRILLIC CAPITAL LETTER E	047/077-10047	Cyrillic capital letter REVERSE E	E5os	
ә	044D	CYRILLIC SMALL LETTER E	047/157-10095	Cyrillic small letter REVERSE E	e5os	
Ә		= 042D		Cyrillic capital letter TILTED REVERSE E (early) ¹¹	E6os	
ә		= 044D		Cyrillic small letter TILTED REVERSE E (early)	e6os	
Ж	0416	CYRILLIC CAPITAL LETTER ZHE	047/050-10024	Cyrillic capital letter ZHE	ZHos	
ж	0436	CYRILLIC SMALL LETTER ZHE	047/130-10072	Cyrillic small letter ZHE	zhos	
Ж		CYRILLIC CAPITAL LETTER RUTHENIAN DZHE ¹²		Cyrillic capital letter DE-ZHE (early Ruthenian)	DZHRros	
ж		CYRILLIC SMALL LETTER RUTHENIAN DZHE		Cyrillic small letter DE-ZHE (early Ruthenian)	dzhros	
З	0405	CYRILLIC CAPITAL LETTER DZE	047/106-10054	Cyrillic capital letter ZELO	DZos	6
з	0455	CYRILLIC SMALL LETTER DZE	047/166-10102	Cyrillic small letter ZELO	dzos	6
З		= 0405	052/241-10913	Cyrillic capital letter INVERTED ZELO (early) ¹³	DZ1os	6
з		= 0455	052/321-10961	Cyrillic small letter INVERTED ZELO (early)	dz1os	6
З		= 0405	052/303-10947	Cyrillic capital letter CROSSED ZELO (early) ¹⁴	DZ2os	6

⁷ Distinguished by a proportionally longer horizontal dimension than its regular counterpart.

⁸ Distinguished by counterclockwise rotation.

⁹ Distinguished by clockwise rotation.

¹⁰ Both upper- and lower-case Cyrillic epsilon are distinguished by indentation in the left side, as in Greek lower-case epsilon.

¹¹ Distinguished by counterclockwise rotation.

¹² Faulmann 185. This spelling is treated as a separate atomic character, rather than a nonce ligature, because it represents a distinctive affricate phoneme when used in Ruthenian (cf. Latin "u" and "w").

¹³ Distinguished by mirroring the normal variant across the y axis.

¹⁴ The current AFII name "Cyrillic capital letter ZE, second alternate (early)" is misleading. Although this glyph physically resembles "Cyrillic capital letter ZE, first alternate (early)" (052/076-10814), but with a cross stroke, and may sometimes be substituted for it, it is used primarily as a variant of "Cyrillic capital letter ZELO" (047/166-10102). The cross stroke may go halfway or all the way across; I have not proposed separate glyph identifiers for these variants. See Karskij 165, 190-91.

Image	UCS Code	UCS (Unicode and 10646) Name	Afii Code	Afii Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
z		= 0455	052/363-10995	Cyrillic small letter CROSSED ZELO (early)	dz2os	6
Z	0417	CYRILLIC CAPITAL LETTER ZE	052/076-10814	Cyrillic capital letter ZE, alternate (early) ¹⁵	Zos	7
z	0437	CYRILLIC SMALL LETTER ZE	052/156-10862	Cyrillic small letter ZE, alternate (early)	zos	7
И	0418	CYRILLIC CAPITAL LETTER I		Cyrillic capital letter STRAIGHT I (early) ¹⁶	IOCTos	8
И	0438	CYRILLIC SMALL LETTER I		Cyrillic small letter STRAIGHT I (early)	ioctos	8
И		= 0418	047/052-10026	Cyrillic capital letter I	IOCT1os	8
и		= 0438	047/132-10074	Cyrillic small letter I	ioct1os	8
~		= 0438		Cyrillic small letter KENDEMA-LIKE SUPERSCRIPIT I (early) ¹⁷	ioct2os	8
І	0406	CYRILLIC CAPITAL LETTER BYELORUSSIAN-UKRAINIAN I	047/107-10055	Cyrillic capital letter DECIMAL I ¹⁸	IDECos	10
і	0456	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I		Cyrillic small letter DOTLESS DECIMAL I (early) ¹⁹	idecos	10
І		= 0406		Cyrillic capital letter THIRD I (Glagolitic transcription) ²⁰	IDECGos	10
і		= 0456		Cyrillic small letter THIRD I (Glagolitic transcription)	idecgos	10
І̇		= 0406		Cyrillic capital letter DOTTED DECIMAL I (early)	IDEC2os	10
і̇		= 0456	047/167-10103	Cyrillic small letter DECIMAL I ²¹	idec2os	10

¹⁵ The current Afii name, "Cyrillic capital letter ZE, first alternate (early)", is called "first" because of the misleading "second alternate" discussed above. I have therefore removed the "first" from this name.

¹⁶ Distinguished from its modern counterpart, Afii 047/052-10026, by having a horizontal cross-stroke, while the modern variant slants from the lower left to the upper right corner. Intermediate representations should be associated with the glyph they resemble most closely. Intermediate images that must be distinguished may be registered separately.

¹⁷ Distinguished by its resemblance to a double grave accent.

¹⁸ *Register* identifies both 047/107-10055 (І) and 047/052-10026 (И) identically as "Cyrillic capital letter I" (likewise their lower-case counterparts, 047/167-10103 "і" and 047/132-10074 "и"). The present inventory modifies the current

Afii names by adding the word "decimal" to the "i" glyphs, so that they will be distinguished from the names of the "и" glyphs.

Modern Cyrillic typography regards 047/107-10055 "Cyrillic capital letter DECIMAL I" (which is dotless) and 047/167-10103 "Cyrillic small letter DECIMAL I" (which is dotted) as upper and lower case counterparts of the same letter. The number of dots is variable in early Cyrillic writing, and I have arranged the glyphs accordingly.

¹⁹ Distinguished by the absence of any superscript dots.

²⁰ Distinguished by the absence of any superscript dots, and by a tail that curves upward and to the right.

²¹ Distinguished by a single superscript dot.

Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
Ї	0407	CYRILLIC CAPITAL LETTER YI (Ukrainian) ²²	047/110-10056	Cyrillic capital letter I WITH TWO DOTS	IDEC3os	10
ї	0457	CYRILLIC SMALL LETTER YI (Ukrainian)	047/170-10104	Cyrillic small letter I WITH TWO DOTS	idec3os	10
Й	0419	CYRILLIC CAPITAL LETTER SHORT I ²³	047/053-10027	Cyrillic capital letter SHORT I	Jos	
й	0439	CYRILLIC SMALL LETTER SHORT I	047/133-10075	Cyrillic small letter SHORT I	jos	
Ј	0408	CYRILLIC CAPITAL LETTER ЈЕ	047/111-10057	Cyrillic capital letter ЈЕ	J1os	
ј	0458	CYRILLIC SMALL LETTER ЈЕ	047/171-10105	Cyrillic small letter ЈЕ	j1os	
К	041A	CYRILLIC CAPITAL LETTER КА	047/054-10028	Cyrillic capital letter КА	Kos	20
к	043A	CYRILLIC SMALL LETTER КА	047-134/10076	Cyrillic small letter КА	kos	20
Л	041B	CYRILLIC CAPITAL LETTER ЕЛ	047/055-10029	Cyrillic capital letter ЕЛ	Los	30
л	043B	CYRILLIC SMALL LETTER ЕЛ	047/135-10077	Cyrillic small letter ЕЛ	los	30
Л̆	0409	CYRILLIC CAPITAL LETTER LЈЕ	052/245-10917	Cyrillic capital letter SOFT L (early)	SOFTLos	
л̆	0459	CYRILLIC SMALL LETTER LЈЕ	052/325-10965	Cyrillic small letter SOFT L (early)	softlos	
М	041C	CYRILLIC CAPITAL LETTER ЕМ	047/056-10030	Cyrillic capital letter ЕМ	Mos	40
м	043C	CYRILLIC SMALL LETTER ЕМ	047/136-10078	Cyrillic small letter ЕМ	mos	40
М̆		CYRILLIC CAPITAL LETTER SOFT М ²⁴		Cyrillic capital letter SOFT М (early) ²⁵	SOFTMos	
м̆		CYRILLIC SMALL LETTER SOFT М		Cyrillic small letter SOFT М (early)	softmos	
Н	041D	CYRILLIC CAPITAL LETTER ЕН	047/057-10031	Cyrillic capital letter ЕН	Nos	50
н	043D	CYRILLIC SMALL LETTER ЕН	047/137-10079	Cyrillic small letter ЕН	nos	50
Н̆		= 041D		Cyrillic capital letter SLANTED ЕН (early) ²⁶	N1os	50
н̆		= 043D		Cyrillic small letter SLANTED ЕН (early)	n1os	50
Н̇	040A	CYRILLIC CAPITAL LETTER NЈЕ		Cyrillic non-Slavic capital letter SLANTED PALATAL ЕН (early)	SOFTNos	
н̇	045A	CYRILLIC SMALL LETTER NЈЕ		Cyrillic non-Slavic small letter SLANTED PALATAL ЕН (early)	softnos	

²² Doubly dotted "i" does not function as an independent character in early Cyrillic writing, but it is included as such in UCS because of its use in modern Cyrillic. In early Cyrillic electronic texts, doubly dotted "i" may be encoded either as one character or as a composite. Upper and lower case singly and doubly dotted "i" may be imaged as single glyphs or as composites.

²³ May be encoded as one or two characters for early Cyrillic and imaged as one or two glyphs. See also the discussion of the angle of the cross stroke for glyphs mapped to U+0418.

²⁴ Golyšenko 42.

²⁵ Distinguished by a palatalization hook attached to the upper right edge.

²⁶ Distinguished from its modern counterpart by having a crossstroke that slants from the upper left to the lower right, while the modern variant has a straight crossstroke.

Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
О	041E	CYRILLIC CAPITAL LETTER O	047/060-10032	Cyrillic capital letter o	Oos	70
о	043E	CYRILLIC SMALL LETTER O	047/140-10080	Cyrillic small letter o	oos	70
О	047A	CYRILLIC CAPITAL LETTER ROUND OMEGA	052/242-10914	Cyrillic capital letter BROAD o (early) ²⁷	BROADOos	70
о	047B	CYRILLIC SMALL LETTER ROUND OMEGA	042/322-10962	Cyrillic small letter BROAD o (early)	broadoos	70
⊙		= 041E	052/243-10915	Cyrillic capital letter OCULAR o (early) ²⁸	OOCos	
⓪		= 043E	052/323-10963	Cyrillic small letter OCULAR o (early)	oocos	
⊕		= 041E		Cyrillic capital letter BINOCULAR o (early) ²⁹	OBOCOs	
⓪		= 043E		Cyrillic small letter BINOCULAR o (early)	obocos	
⊕		= 041E		Cyrillic capital letter SPLIT BINOCULAR o (early) ³⁰	OBOCOs	
⓪		= 043E		Cyrillic small letter SPLIT BINOCULAR o (early)	oboclos	
⊕		= 043E		Cyrillic letter MULTIOCULAR o (early) ³¹	omocos	
И	041F	CYRILLIC CAPITAL LETTER PE	047/061-10033	Cyrillic capital letter PE	Pos	80
и	043F	CYRILLIC SMALL LETTER PE	047/141-10081	Cyrillic small letter PE	pos	80
Р	0420	CYRILLIC CAPITAL LETTER ER	047/062-10034	Cyrillic capital letter ER	Ros	100
р	0440	CYRILLIC SMALL LETTER ER	047/142-10082	Cyrillic small letter ER	ros	100
Ѡ		= 0440		Cyrillic small letter RECUMBANT R (early) ³²	rros	
С	0421	CYRILLIC CAPITAL LETTER ES	047/063-10035	Cyrillic capital letter ES	Sos	200
с	0441	CYRILLIC SMALL LETTER ES	047/143-10083	Cyrillic small letter ES	sos	200
Т	0422	CYRILLIC CAPITAL LETTER TE	047/064-10036	Cyrillic capital letter TE	Tos	300
т	0442	CYRILLIC SMALL LETTER TE	047/144-10084	Cyrillic small letter TE	tos	300
Ш		= 0422		Cyrillic capital letter THREE-LEGGED TE (early) ³³	Tlos	300

²⁷ Distinguished by greater width than regular 047/140-10080 (о). The small ticks at the top and bottom of the circle are common in early Cyrillic typefaces, but are not distinctive, and may be absent in some manuscripts that nonetheless distinguish narrow and wide letterforms. The decision as to whether this and other o-type letterforms are most appropriately viewed as variants of U+043E (о) or U+0461 (w) is somewhat arbitrary. Cf. Sreznevskij 196.

²⁸ Distinguished by a single dot inside a single circle.

²⁹ Distinguished by two dots inside a single circle.

³⁰ Distinguished by two touching circles, each with a single dot in the center.

³¹ Distinguished by a cluster of more than two touching circles, each with a single dot in the center. The number of circles is variable. This glyph lacks separate upper- and lower-case variants.

³² Distinguished by ninety-degree counter-clockwise rotation. Superscript only. No case distinction.

³³ Distinguished by three vertical strokes that reach the baseline. Intermediate forms, such as the "two-and-a-half-legged T" discussed above, may be rendered with this glyph if it is not important to retain the graphic difference between the two. If this difference is important, additional glyphs may be added to the inventory.

Image	UCS Code	UCS (Unicode and 10646) Name	AFLI Code	AFLI Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
III		= 0442		Cyrillic small letter THREE-LEGGED TE (early)	t2os	300
7		= 0442		Cyrillic CURSIVE TE (early) ³⁴	t3os	300
Ѐ	040B	CYRILLIC CAPITAL LETTER TSHE (Serbocroatian)		Cyrillic capital letter DJERV (early)	DJos	
Ӑ	045B	CYRILLIC SMALL LETTER TSHE (Serbocroatian)		Cyrillic small letter DJERV (early)	djos	
Ғ		= 040B	047/114-10060	Cyrillic capital letter SOFT T	DJ1os	
Һ		= 045B	047/174-10108	Cyrillic small letter SOFT T	dj1os	
У	0423	CYRILLIC CAPITAL LETTER U	047/065-10037	Cyrillic capital letter U	Uos	400
у	0443	CYRILLIC SMALL LETTER U	047/145-10085	Cyrillic small letter U	uos	400
Ѹ		= 0443		Cyrillic SUPERSCRIPED CURVED U (early) ³⁵	u1os	
Оу	0478	CYRILLIC CAPITAL LETTER UK	052/307-10951	Cyrillic upper-and-lowercase UK DIGRAPH (early) ³⁶	Ouos	400
OU		= 0478		Cyrillic uppercase UK DIGRAPH ³⁷	OUos	400
оу	0479	CYRILLIC SMALL LETTER UK	052/327-10967	Cyrillic lowercase UK DIGRAPH (early)	ouos	400
ОѸ		= 0478		Cyrillic upper-and-lower case UK DIGRAPH WITH IZHITSA (early)	Ou1os	400
OU		= 0478		Cyrillic upper case UK DIGRAPH WITH IZHITSA (early)	OU1os	400
оѸ		= 0479		Cyrillic lower case UK DIGRAPH WITH IZHITSA (early)	ou1os	400
Ѹ		= 0478	052/246-10918	Cyrillic capital letter UK LIGATURE (early)	OU2os	400
Ѹ		= 0479	052/326-10966	Cyrillic small letter UK LIGATURE (early)	ou2os	400
У	040E	CYRILLIC CAPITAL LETTER SHORT U (Byelorussian) ³⁸	047/116-10062	Cyrillic capital letter SHORT U	USos	
у	045E	CYRILLIC SMALL LETTER SHORT U (Byelorussian)	047/176-10110	Cyrillic small letter SHORT U	usos	
Ф	0424	CYRILLIC CAPITAL LETTER EF	047/066-10038	Cyrillic capital letter EF	Fos	500
ф	0444	CYRILLIC SMALL LETTER EF	047/146-10086	Cyrillic small letter EF	fos	500
Ѳ	0472	CYRILLIC CAPITAL LETTER FITA	047/243-10147	Cyrillic capital letter FITA	THos	9
ѳ	0473	CYRILLIC SMALL LETTER FITA	047/323-10195	Cyrillic small letter FITA	thos	9
Х	0425	CYRILLIC CAPITAL LETTER HA	047/067-10039	Cyrillic capital letter KHA	Xos	600

³⁴ No case distinction.³⁵ Superscript only.³⁶ Sic. *Register* usually identifies alphabetic glyphs as "capital" or "small", but in other places uses "uppercase" and "lowercase" instead.³⁷ Used in decorative titles and other text that may be written entirely in upper case.³⁸ May be encoded as one or two characters for early Cyrillic and imaged as one or two glyphs.

Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
x	0445	CYRILLIC SMALL LETTER HA	047/147-10087	Cyrillic small letter KHA	xos	600
Ω	0460	CYRILLIC CAPITAL LETTER OMEGA	052/250-10920	Cyrillic capital letter OMEGA (early)	OMos	800
ω	0461	CYRILLIC SMALL LETTER OMEGA	052/330-10968	Cyrillic small letter OMEGA (early)	omos	800
Ω̇	047C	CYRILLIC CAPITAL LETTER OMEGA WITH TITLO ³⁹	052/300-10944	Cyrillic capital letter OMEGA with TITLO (early)	OMTITos	
ω̇	047D	CYRILLIC SMALL LETTER OMEGA WITH TITLO	052/360-10992	Cyrillic small letter OMEGA with TITLO (early)	omtitos	
Ω̇̆	047E	CYRILLIC CAPITAL LETTER OT ⁴⁰	052/277-10943	Cyrillic capital letter OMEGA with 't' diacritic (early)	OMTos	
ω̇̆	047F	CYRILLIC SMALL LETTER OT	052/357-10991	Cyrillic small letter OMEGA with 't' diacritic (early)	omtos	
Ш	0429	CYRILLIC CAPITAL LETTER SHCHA	052/273-10939	Cyrillic capital letter SHTE (early) ⁴¹	SHTos	
ш	0449	CYRILLIC SMALL LETTER SHCHA	052/353-10987	Cyrillic small letter SHTE (early)	shtos	
Щ	0426	CYRILLIC CAPITAL LETTER TSE	047/070-10040	Cyrillic capital letter TSE	TSos	900
щ	0446	CYRILLIC SMALL LETTER TSE	047/150-10088	Cyrillic small letter TSE	tsos	900
Ш̆		= 0426		Cyrillic capital letter REVERSED TSE (early) ⁴²	TS1os	900
ш̆		= 0446		Cyrillic small letter REVERSED TSE (early)	ts1os	900
Д̆	040A	CYRILLIC CAPITAL LETTER DZHE	047/241-10145	Cyrillic capital letter HARD DJ	DZos	
д̆	045A	CYRILLIC SMALL LETTER DZHE	047/321-10193	Cyrillic small letter HARD DJ	dzos	
Ч̆	0427	CYRILLIC CAPITAL LETTER CHE	052/265-10933	Cyrillic capital letter CUPPED CHE (early) ⁴³	CHos	90
ч̆	0447	CYRILLIC SMALL LETTER CHE	052/345-10981	Cyrillic small letter CUPPED CHE (early)	chos	90
Ч̇		= 0427	047/071-10041	Cyrillic capital letter CHE	CH1os	90
ч̇		= 0447	047/151-10089	Cyrillic small letter CHE	ch1os	90
Ѡ	0480	CYRILLIC CAPITAL LETTER KOPPA	052/266-10934	Cyrillic capital letter KOPPA (early)	KOPos	90
ѡ	0481	CYRILLIC SMALL LETTER KOPPA	052/346-10982	Cyrillic small letter KOPPA (early)	kopos	90
Ш̈	0428	CYRILLIC CAPITAL LETTER SHA	047/152-10090	Cyrillic capital letter SHA	SHos	

³⁹ Functionally a sequence of the preceding plus two non-spacing superscript marks, but encoded as a separate character in UCS. The use of this composite character is deprecated in favor of a three-character encoding. Despite the UCS and AFII names, neither of the superscript marks is a titlo, which is an abbreviation mark.

⁴⁰ Functionally a sequence of U+0460 "CYRILLIC CAPITAL LETTER OMEGA" and U+0422 "CYRILLIC CAPITAL LETTER TE", with the superscription encoded through rich text markup. This atomic character is listed here because it is

already present in UCS, but its use is deprecated in favor of a two-character encoding.

⁴¹ Distinguished from its modern counterpart, 047/073-10043 (Ш), by having a centered descender.

⁴² Distinguished from the preceding by mirroring across the y axis.

⁴³ Distinguished from the following glyph by having a vertical stroke that is attached to the lowest part of the cup, rather than the right edge. Despite the names, which focus on the cup, the cup itself is identical in both variants.

Ш	0448	CYRILLIC SMALL LETTER SHA	047/072-10042	Cyrillic small letter SHA	shos	
Ђ	042A	CYRILLIC CAPITAL LETTER HARD SIGN	047/074-10044	Cyrillic capital letter ER	BJERos	
Ѣ	044A	CYRILLIC SMALL LETTER HARD SIGN	047/154-10092	Cyrillic small letter ER	bjeros	
Ѣ	042B	CYRILLIC CAPITAL LETTER YERU ⁴⁴	052/252-10922	Cyrillic capital letter YERY WITH YER HANDLE (early) ⁴⁵	JERYBos	
ѣ	044B	CYRILLIC SMALL LETTER YERU	052/332-10970	Cyrillic small letter YERY WITH YER HANDLE (early)	jerybos	
Ѥ		= 042B	047/075-10045	Cyrillic capital letter ERY ⁴⁶	JERYos	
ѥ		= 044B	047/155-10093	Cyrillic small letter ERY	jeryos	
Ѧ	042C	CYRILLIC CAPITAL LETTER SOFT SIGN	047/076-10046	Cyrillic capital letter SOFT SIGN	FJERos	
ѧ	044C	CYRILLIC SMALL LETTER SOFT SIGN	047/155-10094	Cyrillic small letter SOFT SIGN	fjeros	
Ѩ		= 042C	052/251-10921	Cyrillic capital letter NEUTRAL YER (early) ⁴⁷	NJERos	
ѩ		= 044C	052/331-10969	Cyrillic small letter NEUTRAL YER (early)	njeros	
Ѫ	033E	NON-SPACING VERTICAL TILDE ⁴⁸	052/317-10959	Cyrillic PAEROK sign (early)	paerokos	
ѫ		= 033E	052/320-10960	Cyrillic ERIK sign (early) ⁴⁹	erikos	
Ѭ	0462	CYRILLIC CAPITAL LETTER YAT	047/242-10146	Cyrillic capital letter YAT (early)	JATos	
ѭ	0463	CYRILLIC SMALL LETTER YAT	047/322-10194	Cyrillic small letter YAT (early)	jatos	
Ѯ		CYRILLIC CAPITAL LETTER IOTIFIED YAT	052/253-10923	Cyrillic capital letter JOTATED (IOTIZED) YAT (early) ⁵⁰	JATJos	
ѯ		CYRILLIC SMALL LETTER IOTIFIED YAT	052/333-10971	Cyrillic small letter JOTATED (IOTIZED) YAT (early)	jatjos	
Ѱ	042E	CYRILLIC CAPITAL LETTER YU	047/100-10048	Cyrillic capital letter YU	JUos	
ѱ	044E	CYRILLIC SMALL LETTER YU	047/160-10096	Cyrillic small letter YU	juos	
Ѳ		= 042E		Cyrillic capital letter REVERSE YU ⁵¹	JURos	
ѳ		= 044E		Cyrillic small letter REVERSE YU	juros	

⁴⁴ The main function of ISO names is to identify items uniquely and unambiguously, and names are designed to be uniform across ISO standards. The culturally erroneous and embarrassing minomer YERU is part of existing ISO standards, and was retained in ISO 10646-1: 1993 because uniform names across multiple standards were considered more important than culturally correct names.

⁴⁵ Distinguished from the following glyph by having a back jer shape (Ђ) as its first component.

⁴⁶ Sic. Register writes ERY in signature 47 and YERY in signature 52.

































⁴⁷ Distinguished by a smaller flag than back jer (Ђ). In practice, it is often difficult to determine whether a specific letterform is intended to represent front, back, or neutral jer in shape.

⁴⁸ This is misnamed; paerok is a spacing character.


⁴⁹ Distinguished from the preceding by lacking the uppermost diagonal stroke.

⁵⁰ Distinguished from the preceding by the presence of a jotation bar. Cf. Karskij 205.


⁵¹ Distinguished from the preceding by mirroring across the y axis.



Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
	042F	CYRILLIC CAPITAL LETTER YA	052/255-10925	Cyrillic capital letter JOTATED (IOTIZED) A (early)	JAos	
	044F	CYRILLIC SMALL LETTER YA	052/335-10973	Cyrillic small letter JOTATED (IOTIZED) A (early)	jaos	
	0464	CYRILLIC CAPITAL LETTER IOTIFIED E	052/254-10924	Cyrillic capital letter JOTATED (IOTIZED) E (early)	JEos	
	0465	CYRILLIC SMALL LETTER IOTIFIED E	052/334-10972	Cyrillic small letter JOTATED (IOTIZED) E (early)	jeos	
	0466	CYRILLIC CAPITAL LETTER LITTLE YUS	052/257-10927	Cyrillic capital letter MINOR YUS (early)	JUSlos	900
	0467	CYRILLIC SMALL LETTER LITTLE YUS	052/337-10975	Cyrillic small letter MINOR YUS (early)	juslos	900
		CYRILLIC CAPITAL LETTER THIRD YUS	052/260-10928	Cyrillic capital letter THIRD YUS (early) ⁵²	JUS3os	900
		CYRILLIC SMALL LETTER THIRD YUS	052/340-10976	Cyrillic small letter THIRD YUS (early)	jus3os	900
		= CYRILLIC CAPITAL LETTER THIRD YUS		Cyrillic capital letter FOURTH YUS (early) ⁵³	JUS4os	900
		= CYRILLIC SMALL LETTER THIRD YUS		Cyrillic small letter FOURTH YUS (early)	jus4os	900
		= CYRILLIC CAPITAL LETTER THIRD YUS		Cyrillic capital letter FIFTH YUS (early) ⁵⁴	JUS5os	900
		= CYRILLIC SMALL LETTER THIRD YUS		Cyrillic small letter FIFTH YUS (early)	jus5os	900
		= CYRILLIC CAPITAL LETTER THIRD YUS		Cyrillic capital letter EPIGRAPHIC YUS (early) ⁵⁵	JUSEos	900
		= CYRILLIC SMALL LETTER THIRD YUS		Cyrillic small letter EPIGRAPHIC YUS (early)	juseos	900
	046A	CYRILLIC CAPITAL LETTER BIG YUS	047/245-10149	Cyrillic capital letter YUS (early)	JUSBos	
	046B	CYRILLIC SMALL LETTER BIG YUS	047/325-10197	Cyrillic small letter YUS (early)	jusbos	
		= 046A		Cyrillic capital letter BLENDED YUS (early) ⁵⁶	JUSBlos	
		= 046B		Cyrillic small letter BLENDED YUS (early)	jusblos	
	0468	CYRILLIC CAPITAL LETTER IOTIFIED LITTLE YUS	052/261-10929	Cyrillic capital letter JOTATED (IOTIZED) MINOR YUS (early)	JUSJlos	
	0469	CYRILLIC SMALL LETTER IOTIFIED LITTLE YUS	052/341-10977	Cyrillic small letter JOTATED (IOTIZED) MINOR YUS (early)	jusjlos	
		= 0468		Cyrillic capital letter JOTATED (IOTIZED) THIRD YUS (early) ⁵⁷	JUSJ3os	

⁵² Distinguished from the preceding by the presence of a horizontal baseline stroke and the absence of the centered vertical stroke that descends from the middle of the glyph.

⁵³ Distinguished from 052/257-10927 () by a horizontal baseline stroke.

⁵⁴ Shaped like an inverted "V" with a shorter diagonal stroke running from the midpoint of one of the main diagonals to the center of baseline, parallel to the other diagonal. The direction of slant of this short diagonal stroke is not significant.

⁵⁵ Distinguished by a small v-shaped form that replaces the horizontal stroke and vertical descending line in 052/257-10927 ()

⁵⁶ Distinguished by resembling a back nasal 047/245-10149 () with a front nasal 052/257-10927 () tucked underneath. The front nasal portion may resemble any of the front nasal glyphs depicted here.

⁵⁷ Distinguished from the preceding by the presence of a horizontal baseline stroke and the absence of the centered vertical stroke that descends from the middle of the glyph.

Image	UCS Code	UCS (Unicode and 10646) Name	AFLI Code	AFLI Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
Ѣ		= 0469		Cyrillic small letter JOTATED (IOTIZED) THIRD YUS (early)	jus3os	
Ѣ̅	046C	CYRILLIC CAPITAL LETTER IOTIFIED BIG YUS	052/262-10930	Cyrillic capital letter JOTATED (IOTIZED) MAJOR YUS (early)	JUSJBos	
ѣ	046D	CYRILLIC SMALL LETTER IOTIFIED BIG YUS	052/342-10978	Cyrillic small letter JOTATED (IOTIZED) MINOR YUS (early)	jusjbos	
Ѥ	046E	CYRILLIC CAPITAL LETTER KSI	052/263-10931	Cyrillic capital letter KSI (early)	KSios	60
ѥ	046F	CYRILLIC SMALL LETTER KSI	052/343-10979	Cyrillic small letter KSI (early)	ksios	60
Ѧ	0470	CYRILLIC CAPITAL LETTER PSI	052/264-10932	Cyrillic capital letter PSI (early)	PSios	700
ѧ	0471	CYRILLIC SMALL LETTER PSI	052/344-10980	Cyrillic small letter PSI (early)	psios	700
Ѩ	0474	CYRILLIC CAPITAL LETTER IZHITSA	047/244-10148	Cyrillic capital letter IZHITSA (early)	IZHos	400
ѩ	0475	CYRILLIC SMALL LETTER IZHITSA	047/324-10196	Cyrillic small letter IZHITSA (early)	izhos	400
Ѫ	0476	CYRILLIC CAPITAL LETTER IZHITSA WITH DOUBLE GRAVE ACCENT ⁵⁸	052/113-10827	Cyrillic capital letter IZHITSA WITH DOUBLE GRAVE (early) ⁵⁹	IZHKos	
ѫ	0477	CYRILLIC SMALL LETTER IZHITSA WITH DOUBLE GRAVE ACCENT	052/173-10875	Cyrillic small letter IZHITSA WITH DOUBLE GRAVE (early)	izhkos	
Ѭ		CYRILLIC CAPITAL LETTER YN	052/272-10938	Cyrillic capital letter YN (early Romanian)	YNos	
ѭ		CYRILLIC SMALL LETTER YN	052/352-19876	Cyrillic small letter YN (early Romanian)	ynos	

Numerical Diacritics⁶⁰

Ѯ	0482	CYRILLIC THOUSANDS SIGN ⁶¹	052/315-10957	Cyrillic thousand mark (early)	mult4os	
ѯ		CYRILLIC TMA (10,000 SIGN) ⁶²	052/316-10958	Cyrillic ten-thousand mark (early)	mult5os	
Ѱ		CYRILLIC LEGION (100,000 SIGN)	052/375-11005	Cyrillic hundred-thousand mark (early)	mult6os	
ѱ		CYRILLIC LEODOR (1,000,000 SIGN)	052/376-11006	Cyrillic million mark (early)	mult7os	
Ѳ		CYRILLIC VORON (10,000,000 SIGN)		Cyrillic ten-million mark (early)	mult8os	
ѳ		CYRILLIC KOLODA (100,000,000 SIGN)		Cyrillic hundred-million mark (early)	mult9os	









⁵⁸ Deprecated in favor of encoding as a sequence of *izica* plus *kendema*.

⁵⁹ Distinguished from the preceding by the presence of a double grave superscript mark.

⁶⁰ Karskij 219.

⁶¹ Written before the numeral (letter).

⁶² Drawn around the numeral (letter), as are all higher numerical diacritics.

Image	UCS Code	UCS (Unicode and 10646) Name	AFII Code	AFII Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
<i>Abbreviation Marks⁶³</i>						
	0483	COMBINING CYRILLIC TITLO	052/367-10999	Cyrillic titlo (non-spacing abbreviation mark) (early)	titos	
		= 0483		Cyrillic medium titlo (non-spacing abbreviation mark) (early)	tit1os	
		= 0483		Cyrillic long titlo (non-spacing abbreviation mark) (early)	tit2os	
		= 0483	052/310-10952	Cyrillic titlo-operator (non-spacing abbreviation mark) (early) ⁶⁴	pokos	
<i>Non-Spacing Accent Marks and other Superscripted Characters⁶⁵</i>						
	0484	COMBINING CYRILLIC PALATALIZATION		Cyrillic PALATALIZATION HOOK, COMPLETE (early) ⁶⁶	palos	

⁶³ No case distinctions are proposed for superscript abbreviation and other non-spacing marks. In some instances, upper- and lower-case variants are available in the AFII inventory, and distinct upper-case glyphs can be created if a need should arise. It is assumed here that resizing and positioning can be performed algorithmically, based on environment, and that distinctive upper- and lower-case glyphs will not be required in the inventory.

⁶⁴ The standard name for this image in Slavic paleography is pokrytie, but the AFII glyph description that bears this name is erroneously attached to 052/371-11001 "Cyrillic POKRYTIE; also, RTSY-TITLO (non-spacing abbreviation mark)". The latter is imaged as a superscript 047/142-10082 "Cyrillic small letter ER" with a curved mark above it. Pokrytie is normally written over a word to indicate abbreviation, and it may have a superscript letter (any letter) underneath it.



Combinations of pokrytie and superscript letters are often called by the Church Slavonic letter name plus the word "titlo", so that RTSY-TITLO is an acceptable name for the image with the superscript ER (known as RTSY in Church Slavonic).

⁶⁵ Some non-spacing diacritics and punctuation are present in generic registered entity sets, such as ISONUM, ISODIA, etc. (documented, but without

images, in Goldfarb 513 ff). They are duplicated here so as to provide a complete, coordinated early Cyrillic inventory. Additionally, just as graphically similar Latin, Greek, and Cyrillic alphabetic glyphs may be imaged identically or differently, at the discretion of the font designer, so may non-alphabetic glyphs. If Latin, Greek, and Cyrillic are encoded in a single multiple-script font, the provision of distinct entity names will simplify maintaining these graphic distinctions.

⁶⁶ *Register* contains left and right upper- and lower-case Cyrillic palatalization hooks (052/313-10955, 052/314-10956, 052/373-11003, 052/373-11004). As far as I have been able to determine, the left hooks do not occur at all in Slavic Cyrillic writing. The right hooks can be used to compose soft *d*, *m*, *n*, and *l*, which are also available as atomic glyphs (soft *d*, *n*, and *l* are already present in *Register*; soft *m* is proposed here).

Palatals are also commonly represented in early Cyrillic writing by a curved superscript diacritic above the regular base letter. This diacritic may resemble 052/311-10953 "Cyrillic INVERTED BREVE (non-spacing stress mark) (early)", or it may lean slightly to the right (as in the image here).

Image	UCS Code	UCS (Unicode and 10646) Name	AFIG Code	AFIG Glyph Description (SGML Entity Description)	SGML Entity	Num. Value
'	0485	COMBINING CYRILLIC DASIA PNEUMATA	052/366-10998	Cyrillic ROUGH BREATHING SIGN (early) ⁶⁷	roughos	
'	0486	COMBINING CYRILLIC PSILI PNEUMATA	052/306-10950	Cyrillic SMOOTH BREATHING SIGN (early)	smoothos	
^	0301	NON-SPACING ACUTE	000/302-194	Acute (l.c. non-spacing accent)	acuteos	
˘	0300	NON-SPACING GRAVE	000/301-193	Grave (l.c. non-spacing accent)	graveos	
ˆ	030B	NON-SPACING DOUBLE ACUTE	000/315-205	Double acute (l.c. non-spacing accent)	dacuteos	
˘˘	030F	NON-SPACING DOUBLE GRAVE	043/342-9186	Double grave (l.c. non-spacing accent)	dgraveos	
˘		CYRILLIC NON-SPACING LONGA		Cyrillic LONGA (non-spacing diacritic), variant (early) ⁶⁸	longaos	
^	0302	NON-SPACING CIRCUMFLEX	000/303-195	Circumflex (l.c. non-spacing accent)	circos	
˘	0311	NON-SPACING INVERTED BREVE	052/311-10953	Cyrillic INVERTED BREVE (non-spacing stress mark) ⁶⁹	ibreveos	
˘	0306	NON-SPACING BREVE	000/306-198	Breve (l.c. non-spacing accent)	breveos	
—	0304	NON-SPACING MACRON	000/305-197	Macron (l.c. non-spacing accent)	macronos	
.	0307	NON-SPACING DOT ABOVE	000/307-199	Over-dot (l.c. non-spacing accent)	odotos	
¨	0308	NON-SPACING DIÆRESIS	000/310-200	Diaeresis (l.c. non-spacing accent)	diaeros	

Punctuation Used in Early Cyrillic Writing⁷⁰

	0020	SPACE		no image required	spaceos	
'	02BC	MODIFIER LETTER APOSTROPHE	000/271-185	Quotation mark, ending, single	apostos	
+	002B	PLUS SIGN	000/053-43	Plus sign	plusos	
,	002C	COMMA	000/054-44	Comma	commaos	

⁶⁷ Register also includes two variants of "Cyrillic dasia pneumata (non-spacing) (early)": 052/114-10828 and 052/236-10910, as well as two variants of "Cyrillic psili pneumata (non-spacing) (early)": 052/174-10876 and 052/237-10911. These palaeographic terms are synonymous with smooth and rough breathing. The glyphs used here correspond most closely to the images that are most common in Cyrillic manuscripts.

Combinations of breathing plus accent marks are sometimes known by a single name (e.g., a combination of smooth breathing plus acute is called *iso*). These are not included here as composite characters, since they can be composed dynamically from their components as needed.

⁶⁸ Longa is the Latin equivalent of macron, a length mark, and may vary from horizontal position to a small rotation clockwise from the horizontal. I have introduced an alternate glyph because the image depicted in 052/312-10954 "Cyrillic longa (non-spacing diacritic)" is more vertical the images usually associated with the term longa in Slavic palaeography, which risks confusion with a grave accent, and also may not easily be recognized by Slavists as representing a longa.

⁶⁹ Distinguished from the following glyph by mirroring across the *x* axis.

⁷⁰ Other major section marks, consisting of dots, crosses, and dashes, should be composed dynamically as needed.

<i>Image</i>	<i>UCS Code</i>	<i>UCS (Unicode and 10646) Name</i>	<i>AFII Code</i>	<i>AFII Glyph Description (SGML Entity Description)</i>	<i>SGML Entity</i>	<i>Num. Value</i>
.	002D	PERIOD	000/056-46	Period	periodos	
:	003A	COLON	000/072-58	Colon	colonos	
;	03D7	GREEK QUESTION MARK	046/042-9762	Greek question mark (Erotimatiko)	semios	
×	203B	REFERENCE MARK	042/050-8744	kome symbol	refmkos	
.	00B7	MIDDLE DOT	000/267	Dot, Centered	middotos	

Acknowledgements

I am grateful to the following colleagues for their generous comments and suggestions on earlier versions of this paper, and for stimulating discussion of the issues underlying the present work: Glenn Adams, Per Ambrosiani, Lloyd Anderson, Joe Becker, Barbara Beeton, Andrej Bojadžiev, Milena Dobrova, Harry Gaylord, L.C.J. Jacobson, Jouko Lindstedt, Anisava Miltenova, Casey Palowitch, Michael Spring, Oscar Swan, Cynthia Vakareliyska, and Dimitri Vulis. I have not always followed their advice, and they are not responsible for errors of fact or judgment.

Works cited

Standards and documents

AFII-REGISTRY/0002

Registry Collections. Series number 1. *Latin Publishing (Western)*. September 28, 1989.

AFII-REGISTRY/0004 Cyrillic

Registry Collections. Series number 2. *Cyrillic*. September 28, 1989.

Alternative Encoding

See Brjabrin.

Apple Cyrillic

See Adobe 13.

Basic Encoding

See Brjabrin.

CP 1251 = Microsoft Cyrillic Windows 3.1 Character Set

See Adobe 15.

CP 866 = Microsoft's basic code page for Russified MS-DOS

CECP 500 = DKOI-8 = Russified EBCDIC

Other EBCDIC-based Cyrillic extended code pages are CECP 037, JNET, and FORTRAN mappings.

GOST 19768-74 = "old" KOI-8.

See Clews 66.

GOST 19768-87 = "new" KOI-8.

Equivalent to ISO 8859-5: 1988.

ISO 2202: 1986

Information processing – ISO 7-bit and 8-bit coded character sets – Code extension techniques.

ISO 8859-5: 1988 = ECMA-113

Information processing – 8-bit single-byte coded graphic character sets – Part 5: Latin/Cyrillic Alphabet.

ISO 8879: 1986

Information processing – Text and Office Systems – Standard Generalized Markup Language. (Includes Amendment 1.)

ISO/IEC 9541-1: 1991

Information Technology – Font information interchange – Part 1: Architecture.

ISO/IEC 10646-1: 1993

Information Technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.

ISO DIS 6861.2

Information and documentation – Cyrillic alphabet coded character set for historic Slavonic languages and European

non-Slavic languages written in a Cyrillic script, for bibliographic information interchange¹.

N 915

ISO/IEC JTC1/SC2/WG2. N 915. *Character Glyph Model*. 27 September 1993.

P3

Association for Computers and the Humanities (ACH), Association for Computational Linguistics (ACL), and Association for Literary and Linguistics Computing (ALLC). *Guidelines for Electronic Text Encoding and Interchange. (TEI P3)*. Edited by C.M. Sperberg-McQueen and Lou Burnard. 1994.

Register

International Glyph Register. Volume 1: Alphabetic Scripts and Symbols. Rochester: Association for Font Information Interchange. 1993.

RKA-2

Rasširennyj kirilliceskij alfavit 2. Version 7. November 1991. Cited from Andruščenko.

TEI TRI W4

Harry E. Gaylord. *Character Entities and Public Entity Sets*. 1992.

Literature

Adobe

Adobe Standard Cyrillic Font Specification. Technical Note #5013. 25 February 1993. URL file://ftp.adobe.com/pub/adobe/DeveloperSupport/TechNotes/5013.Cyrillic_Font_Spec.ps.

Alipij

Ieromonax Alipij (Gamanovič). *Grammatika cerkovno-slavjanskago jazyka*. Jordanville. 1964.

Andruščenko

V.M. Andruščenko. "Ob organizaciji arxiva istočnikov Mašinnogo fonda ruskogo jazyka, ix razmetke i kommentirovanii". Unpublished.

Birnbaum 1988

David J. Birnbaum. "Computers and the Study of Orthographically Complex Manuscripts". In: Eric Johnson, ed., *Proceedings of the Third International Conference on Symbolic and Logical Computing*. Madison, SD: Dakota State College. 1988. 63-85.

Birnbaum 1989

David J. Birnbaum. "Issues in Developing International Standards for Encoding Non-Latin Alphabets". In: Eric Johnson, ed., *Proceedings of the Fourth International Conference on Symbolic and Logical Computing*. Madison, SD: Dakota State University. 1989. 41-54.

¹ This draft included both Cyrillic and Glagolitic, but the former was removed before final balloting. The final official standard, ISO 6861, which apparently has not yet been published, is exclusively a Glagolitic encoding standard. It most likely bears a different title, but I have not been able to obtain a copy for verification.

- Brjabrin
V.M. Brjabrin, I.Ja. Landau, and M.E. Nemenman. "O sisteme kodirovanija dlja personal'nyx èvm". *Mikroprocesornye sredstva i sistemy*, 4, 1986, 61-63.
- Clews
John Clews. *Language automation worldwide: the development of character set standards*. (British Library R&D reports, 5962) Harrogate: SESAME, 1988.
- Diels
Paul Diels. *Altkirchenslavische Grammatik. I. Teil: Grammatik*. Second edition. Heidelberg: Carl Winter, 1963.
- Faulmann
Carl Faulmann. *Das Buch der Schrift*. 2nd, expanded and corrected, edition. Vienna, 1880.
- Goldfarb
Charles F. Goldfarb. *The SGM Handbook*. Oxford: Clarendon, 1990.
- Golyšenko
V.S. Golyšenko. *Mjagkost' soglasnyx v jazyke vostočnyx slavjan XI-XII vv.* Moscow: Nauka, 1987.
- Hansack
Ernst Hansack. "Paläoslavistik und Computer: die Erschließung des kirchenslavischen Erbes". *Anzeiger für slavische Philologie*, xxii/1, 1993, 89-96.
- Karskij
E.F. Karskij. *Slavjanskaja kirillovskaja paleografija*. Leningrad: Akademija Nauk SSSR, 1928. (Photographic facsimile: Moscow: Nauka, 1979).
- Kornai
Andreas Kornai. "Cyrillic Encoding FAQ", Version 1.3, 13 March 1993. URL file://infomeister.osc.edu/pub/central_eastern_europe/russian/doc/encoding.faq.
- Leskien
August Leskien. *Handbuch der althbulgarischen (altkirchenslavischen) Sprache. Grammatik, Texte, Glossar*. 10th, corrected and expanded edition (by J. Schröpfer). Heidelberg: Carl Winter Universitätsverlag, 1871/1990.
- Lunt
Horace G. Lunt. *Old Church Slavonic Grammar*. Sixth, revised edition. The Hague: Mouton, 1974.
- Mathiesen
Robert C. Mathiesen. "The Determination of Norms. (A Problem in the Diachronic Study of Church Slavonic.)" *American Contributions to the Eighth International Congress of Slavists, Zagreb and Ljubljana, September 3-9, 1978. Volume 1. Linguistics and Poetics*. (Henrik Birnbaum, editor). Columbus: Slavica, 1978, 483-94.
- Miklas 1988
Heinz Miklas. "Zur Struktur des kyrillisch-altkirchenslavischen (althbulgarischen) Schriftsystems". *Palaeobulgarica / Starobălgaristika*, 3, 1988, 52-65.
- Miklas 1989
Heinz Miklas. "Paläographische und graphematische Aspekte der kyrillischen Schriftentwicklung in Bulgarien (bis zum 14. Jahrhundert)". In: *Kulturelle Traditionen in Bulgarien*. (= *Abhandlungen der Akademie der Wissenschaften in Göttingen, phil.-hist. Kl. Dritte Folge*, Bd. 77) Göttingen, 1989, 68-90.
- Miklas 1993
Xajnc Miklas (= Heinz Miklas). "Ot preslavskija sâbor do preslavskata škola. Vâprosi na grafematikata". *Palaeobulgarica / Starobălgaristika*, xvii/3.3-12.
- Pamjatniki
Pamjatniki literatury drevnej rusi. XI-načalo XII veka. Moscow: Xudožestvennaja literatura, 1978.
- Robinson and Solopova
Peter Robinson and Elizabeth Solopova. "Guidelines for Transcription of the Manuscripts of the Wife of Bath's Prologue". *The Canterbury Tales Project Occasional Papers*, 1, 1993, 19-52.
- Stadnikova
E.V. Stadnikova. "Vlijanie akcentnoj sistemy na fonologičeskiju (na materiale istorii dvux fonem 'tipa o' v russkom jazyke)". Kandidat dissertation. Moscow, 1984.
- Trubetzkoy
Nikolaus S. Trubetzkoy. *Altkirchenslavische Grammatik. Schrift-, Laut- und Formensystem*. Second edition. Graz: Hermann Böhlau, 1968.
- Unicode
The Unicode Consortium. *The Unicode Standard, Worldwide Character Encoding, Version 1.0, Volume 1*. Reading: Addison-Wesley, 1991.
- Worth
Dean S. Worth. *The Origins of Russian Grammar. Notes on the state of Russian philology before the advent of printed grammars*. Columbus: Slavica, 1983.



David J. Birnbaum is assistant professor of Slavic Languages and Literatures at the University of Pittsburgh, 1417 Cathedral of Learning, Pittsburgh, PA 15260 USA. He has published in the areas of Slavic linguistics, manuscript studies, and humanities computing, and is an editor of the *Proceedings of the First International Conference on the Computer Processing of Medieval Slavic Manuscripts*, to be published early in 1996 by the Bulgarian Academy of Sciences. He earned an A.B. at Brown University (1976), A.M. degrees at the Ohio State University (1978) and Harvard University (1980), and a Ph.D. in Slavic linguistics at Harvard University (1988).