Frequently Asked Questions and Selected Resources on Cyrillic Multilingual Computing

Kevin S. Hawkins

SUMMARY. The persistence of multiple standards, both *de jure* and *de facto*, for handling text on the computer is perhaps the most perplexing problem facing those who wish to use a computer in more than one language, or even to provide access to data across more than one operating system and application platform. Using Cyrillic as an example, this article attempts to answer questions commonly asked by non-expert computer users who wish to work in more than one language.

Kevin S. Hawkins, MS, is Electronic Publishing Librarian, Scholarly Publishing Office, University Library, University of Michigan.

Address correspondence to: Kevin S. Hawkins, 300 Hatcher Graduate Library North, 920 North University Avenue, Ann Arbor, MI 48109-1205 USA (E-mail: kshawkin@ umich.edu).

The author wishes to thank David Dubin and the other members of the GSLIS Research Writing Group at the University of Illinois at Urbana-Champaign for their comprehensive advice, and Troy Williams, Benjamin Rifkin, Michael Brewer, David Dubin, Tatiana Poliakevitch, Irina Roskin, Peter Houtzagers, and Uladzimir Katkouski for their specific corrections and suggestions.

[Haworth co-indexing entry note]: "Frequently Asked Questions and Selected Resources on Cyrillic Multilingual Computing." Hawkins, Kevin S. Co-published simultaneously in *Slavic & East European Information Resources* (The Haworth Information Press, an imprint of The Haworth Press, Inc.) Vol. 6, No. 2/3, 2005, pp. 3-21; and: *Virtual Slavica: Digital Libraries, Digital Archives* (ed: Michael Neubert) The Haworth Information Press, an imprint of The Haworth Press, Inc., 2005, pp. 3-21. Single or multiple copies of this article are available for a fee from The Haworth Document Delivery Service [1-800-HAWORTH, 9:00 a.m. - 5:00 p.m. (EST). E-mail address: docdelivery@haworthpress.com].

Available online at http://www.haworthpress.com/web/SEEIR

doi:10.1300/J167v06n02_02

KEYWORDS. Character encoding, character sets, keyboard layouts, multilingual computing, fonts, computers, Cyrillic

INTRODUCTION

Nearly everyone who has used a computer has experienced what are often called "font problems" or "encoding problems": letters or punctuation marks outside of the very basic ones used in English, which show up on screen or in print as something other than what was originally intended. While in some cases, such as jumbled punctuation marks, the reader can "read around" these characters, when using non-Latin writing systems it becomes impossible to use a document without decoding it.

While a scrambled document can sometimes be decoded simply by using a different *character encoding scheme* (*character encoding* or *character set*) or a different font, the solution is often not so simple. The problem might involve settings in software used to create, copy, view, or deliver the file, or a combination of these. Information providers, including librarians, have the responsibility to use settings in their own software properly and make choices that will guarantee wider access to their materials. Furthermore, while settings in the user's software are out of the control of information providers, steps can be taken to minimize the likelihood that these settings will interfere with properly reading the document.

We have so many different standards, both *de jure* (official) and *de facto* (unofficial), because multilingual computing, like most technologies, caught on faster than a standard could be agreed upon, especially since dominant commercial and political interests tend to oppose standardization initially in order to preserve market share, sphere of influence, or ideology. The proliferation of conflicting, inferior standards is the result of a history we have not yet succeeded in fully escaping, but all stakeholders, even dominant ones, now agree that it is in everyone's interest to escape this history.

It is so complicated to store and enter characters used in the writing systems of natural languages because there are many software layers and levels of abstraction involved, all of which are easily confused with one another. Terminology is inconsistent, with some terms having common usage but a different technical meaning.

I will address common points of confusion in multilingual computing, using Cyrillic as an example. However, everything that I say ap-

plies to those using non-U.S. computers, those writing in languages in the Latin alphabet other than English, and those using other *scripts* (*writing systems*)[1] on the computer besides Cyrillic. I gloss over distinctions when I believe that they are insignificant to users of contemporary operating systems and software: my loose use of terminology, guaranteed to offend experts in text processing, is designed to make this very complex topic accessible to the lay audience. I hope that my conversational style and use of the frequently-asked-questions (FAQ) format, while unorthodox for a scholarly journal, will make this very difficult topic easier to understand. Following the FAQ is a short bibliography of resources on multilingual computing.

IMPORTANT POINTS OF CLARIFICATION

There is one terminological and ontological distinction that must be explained up front: the difference between a *character* and a *glyph.* Characters are basic, abstract units in a writing system and are not to be confused with their particular appearance in any font (called a glyph).[2] The Latin character *B* is not the same as the Cyrillic character *V*, despite appearances, and the Latin character *M* is not the same as the Cyrillic character *M* even though the latter pair are used for essentially the same sound. A character is still the same character no matter what glyph you use: whether you write it in a serif font or a sans serif font, or whether you write it in italics or boldface or large or small. Storing text as characters rather than glyphs enables searching within and across files, where the particulars of appearance are rarely significant.[3] Characters can be correlated when helpful in searching–for example, capital and lowercase letters are correlated at the operating system level (and in many programs), whereas it is not helpful to correlate Latin *M* and Cyrillic *M*. Which languages are considered for the purposes of character sets to use the same writing system has the potential to be a contentious issue, but such decisions are increasingly made by experts who strive to balance linguistic and pragmatic needs. French *B* and English *B* are considered the same character, as are Russian *G* and Ukrainian *H*. That is, the notion of character is writing-system dependent, not language-dependent.

Another common point of confusion among computer users is over whether it is possible in creating a digital document of any sort to guarantee that the reader will see it exactly as the creator does. While the exact appearance of a document can be preserved in PDF format,[4] it is

impossible to preserve it in HTML and even word processing formats since many of these details change as you view the file. There are non-standard and deprecated practices in HTML for specifying and even embedding fonts, but there is no way to be sure that the document will show up exactly as you see it when viewed by another person, who may have a different operating system, collection of fonts, web browser, and browser settings. The appearance of word processor documents is subject to the user's settings in the word processor, the fonts available, and the printer driver. So you should never create a document whose proper appearance depends on text appearing at a certain location on the page or appearing in a certain font or size.

FREQUENTLY ASKED QUESTIONS

1. I want to read text written in Cyrillic. Do I need to download a special font? What if I know I can view text in Cyrillic already but the website claims I need a special font? Do I really?

Until the mid-1990s special fonts were needed to work in "exotic" languages on major operating systems distributed in the U.S. Since then all major operating systems come with some built-in or freely available capacity for displaying documents in Cyrillic. At first this involved adding optional components–especially fonts and keyboard layouts–to work in exotic languages. These fonts often had "Cyrillic" or "Cyr" appended to the name but otherwise matched pre-installed fonts in name and appearance. Since then, due to standardization on Unicode,[5] any given font usually has Cyrillic "built in" to it, so separate fonts for different scripts are no longer needed and one web page can now contain characters from more than one non-Latin script. In today's operating systems the fonts and keyboard layouts are installed by default for many languages and need only be activated, or at least they are included on the installation disks.

However, sometimes websites providing textual resources in Cyrillic will claim that a certain font is necessary to view the text properly. While this is never strictly true, there are three reasons why it can be practically true. First, if the resource was created using a custom font with a non-standard *font encoding*,[6] such as those that allow you to type in Cyrillic by changing fonts but not changing keyboard layouts, you will need to use some font with the same font encoding, and chances are

this font is the only one out there. (See question #6 for reasons to avoid such custom fonts.) Second, it is possible that the resource uses a standard encoding but includes at least one character whose glyph is simply missing from common pre-installed fonts; in this case, you will need to download this font or use another one that includes all the glyphs required. Third, the author might have attempted to guarantee the *exact* appearance of the document. While using the specified font can help, there is no guarantee that this will give the exact desired appearance, as explained above.

Therefore, to read documents including Cyrillic text, no special fonts should be needed if the document was created using a standard font (with a standard font encoding). Word processing documents with a standard font generally open without problem, whereas the proper viewing of HTML files is subject to browser and server settings, either of which can cause the wrong character encoding scheme to be chosen. If this is the case, see question #2.

2. I have trouble viewing certain websites. Why?

As explained in question #1, you generally do not need any special fonts to view text in Cyrillic; therefore, problems with garbled Cyrillic text are most likely caused by improper settings on the server where the file is located or on your browser, and possibly both. The following explains how to decipher pages you come across; see questions #9, #10, and #11 for information on how to minimize problems for viewers of your own websites.

Regardless of what is causing the garbled text, web pages are best deciphered by enabling the character encoding "auto-detect" feature of your browser since this feature usually selects the character encoding you need. In some browsers, there are different auto-detect features for different languages. If the feature ever selects the wrong encoding when viewing a web page, you can override it by selecting an encoding from the list, usually without disabling the auto-detect feature. Auto-detect is usually not enabled by default in browsers for some strange reason, and it can easily be disabled by accident. There is no particular disadvantage to leaving this feature on unless you are testing your own web pages for encoding compatibility. Note that auto-detect features poorly judge web pages with HTML frames, so you will most likely need to manually choose an encoding when viewing such pages.

3. I want to type in Cyrillic. Do I need to download a special keyboard layout (driver)? Is there any way to type without adding a keyboard layout?

First we need some background on keyboards. When you buy a computer, it almost always comes with a keyboard made for users in the country in which it was purchased, with keys labeled for the most commonly used characters. Some characters do not have their own keys but instead are sent to the computer by typing a combination of keys simultaneously, for example, by holding down Ctrl, Shift, Alt, or AltGr (the right Shift key on American keyboards). While American keyboards only use the Shift key to enter alternative characters, in other keyboard layouts it is common to have access to more than one script through the keyboard by using key combinations. A similar functionality might be provided by your operating system, word processor, or both, but these are in fact layered on top of what is provided by the physical keyboard.

Thankfully for users, keyboard layouts are quite standard in all countries: the QWERTY layout, with slight national variants, is nearly universal for languages using the Latin script, and languages using Cyrillic have just a few basic layouts.

Your operating system can be set to work with any keyboard layout you specify, and today's operating systems come with all the major international layouts by default, just as they contain all fonts for non-Latin scripts. Choosing a keyboard layout is one of the first steps in an installation process. Since keyboards all over the world have almost the same number and arrangement of keys, just with different arrangements of characters, it is generally possible to use any physical keyboard with any keyboard layout. Furthermore, operating systems allow you to change the default keyboard layout or to install more than one keyboard layout with one or more convenient ways to switch between them (such as by clicking in a special toolbar for language settings) or by using a key sequence like Alt + Shift.

The problem, then, after changing your keyboard layout or adding an additional one, is knowing where to find the characters you need if your physical keyboard is labeled with the characters of the country where you purchased it. Keyboards almost all use the same type of cable to connect to a computer,[7] so one often overlooked solution is to buy a keyboard in another country while traveling there. Keyboards sold in their target market, even in the West, are quite inexpensive. Foreign keyboards can be purchased online from domestic websites but cost

much more this way. You can also buy clear stickers online for your domestic keyboard, thereby allowing you to fake a keyboard that you might buy in another country. A cheaper solution is to make your own stickers. Transparent (scotch) tape works well but is difficult to remove after being constantly pressed by warm fingers. Masking tape is more easily removed but tends to slide under the heat of your fingers and is annoying to the touch for some users. If you type in another language frequently enough, you might instead print a picture of the keyboard layout you are using and learn to touch-type by looking at it. Learning the layout doesn't take as much effort as you might think.

I highly recommend one of the above methods of learning a standard keyboard layout so that when traveling to Slavic, East European, and Eurasian countries you will be able to adapt to the commonly used keyboard layout immediately. Some Western users, however, prefer a *homophonic* ("transliterated" or "phonetic") keyboard layout, in which characters are arranged similar to those in one's native layout. This layout is easier to learn for the non-native user and does not require stickers or a new keyboard. Numerous web pages explain how to download and install such keyboard layouts (there are various versions but no one accepted standard) on your computer (see Further Reading section).

Some text editors have a built-in "de-transliteration" function, by which you type in transliteration and the editor converts to Cyrillic characters. Some also have built-in phonetic keyboard layouts, allowing you to use a phonetic keyboard just in that program without changing your operating system keyboard layout.

For especially small input needs, there are three options, none of which requires changing the operating system keyboard layout. Websites such as *Translit.ru* allow you to type in transliteration and the site will convert your text to Cyrillic on the fly. Some text editors and websites have virtual keyboards in small windows that allow you to choose Cyrillic characters. These applications can be used to copy and paste text into other programs or websites. Another option is to copy and paste letters from other programs. This usually works smoothly, especially with Unicode-conformant software. A third option is to use a *character palette* (or map) to insert characters. A palette may be found within a word processor or as a standalone program in the operating system. If using this, it is best to select from the default font; otherwise, you risk selecting characters from a font with a non-standard font encoding (see questions #1 and #6).

4. When I copy and paste between programs, my characters get jumbled. How do I fix this?

More often than not, this is a sign that one of the programs you are using is not fully Unicode-conformant.[8] Try using other combinations of programs.

5. Say I need to type just one Russian word in the middle of a document in English and it uses letters also found in the Latin alphabet. Can I just type these letters using my default American keyboard layout?

Please do not. The three characters in the Russian word *mat* are not the same as the three characters in the English word *mat*. Storing them as different characters prevents the Latin lowercase *T* from ever being displayed like the Cyrillic lowercase *T* (which in most typefaces are not identical), and it prevents a search for the English word from finding the Russian word or vice versa.

6. What font should I use to type in Cyrillic?

When using a contemporary word processor to compose a document involving characters from more than one script, choose characters from a Unicode-conformant font (one that uses Unicode font encoding)[9] rather than a font like Symbol, Zapf Dingbats, or one of the pre-Unicode custom fonts for typing in a non-Latin script. These fonts use only the first 255 *code points*,[10] replacing the Latin glyphs and basic punctuation that should be in these code points with Greek glyphs, printer's dingbats, or glyphs from another non-Latin script (respectively). While these fonts can be used with old programs that recognize only fonts with 255 code points and were especially useful before special keyboard layouts became common, the documents created with them are entirely dependent on the arrangement of glyphs in this font–an arrangement that does not conform to any particular standard and therefore will likely require conversion in case the font is no longer available.[11] For example, while the Symbol font might continue to be shipped with operating systems for the foreseeable future, Greek texts using it are not interoperable with Greek texts in Unicode, so cross-searching is not possible unless the document in Symbol is converted. In such a case you will need to find a utility that will do this conversion, and unless you write one yourself, the chance that you find one is related to the size and

technical expertise of the user community for that font with a non-standard encoding.

Using Unicode rather than an older standard allows you to capture almost every character likely to be encountered in a printed text[12] and store the electronic text in a standard broadly accepted by industry and the international standards community. If you have scholarly or practical problems whereby the 96,447[13] characters of Unicode version 4.0.1 are inadequate for the kind of transcription you would like to do, you can use Unicode's Private Use Areas, which contains unassigned code points.

In short, I recommend that you never switch fonts in a word processor just to type in another language. Stick to one font, using a second keyboard layout or other input method as described in question #3 to insert glyphs for the character you need.

7. I want to be able to use diacritical marks with certain characters in Cyrillic, but I don't see the glyphs I need in Unicode. Can I still use Unicode? Do I need a special font?

Unicode aims to include all characters in all scripts, so one font could have all the glyphs for every character needed in any document. However, the designers of Unicode realized that many diacritical marks can be used on nearly any character, and therefore it makes sense to define *base characters* and *combining characters* separately.[14] Since Unicode includes all the diacritical marks you'll ever need, I recommend against using another font just because it has particular *precomposed characters* that you need. A document relying on such a font would suffer the same compatibility problems as affect other pre-Unicode fonts (see question #6). Instead, I recommend using Unicode base characters and combining characters.

However, many programs that can read, write, or print files using a Unicode character encoding scheme have trouble displaying *base characters* and *combining characters* together properly, especially if there is more than one combining character applied to the base character. This can even be true when going from the same program running on different operating systems. If you are unhappy with the way your program renders *composite characters* (decomposable characters, including base characters with combining characters applied to them), you might try using other Unicode-conformant fonts. If you cannot find a font that works acceptably with the editor or word processor you are using, it is best to find another program that works better: the strong disadvantages

of using a font with a non-standard encoding, as explained in question #6, outweigh the convenience of using a program you are most familiar with. Support for Unicode character sequences involving combining characters can only improve in the future.

8. I see that Unicode contains a number of precomposed characters to guarantee round-trip compatibility with earlier standards, but "the same" character can be encoded by finding the appropriate base character and combining character and using them together. Which should I use?

The World Wide Web Consortium,[15] stewards of HTML and other web standards, recommends using precomposed characters (using Unicode's Normalization Form C) for web documents.[16] Even if your documents are not intended for the web, you will find it easier to work with precomposed characters since many programs handle composite characters poorly, as explained in question #7. There are utilities for automatically converting between normalization forms in Unicode (between using precomposed and composite characters), so you can always decompose later. Note that the Unicode Consortium refuses to admit any new precomposed characters to the standard, and a major design goal of Unicode is to decompose all characters as far as possible (to the atomic level).

9. What character encoding (character set) should I use for saving my web pages or other documents?

Word processors often do not present the user with a choice of character set when saving in their native formats; instead, they translate to a standard character set when saving in a text format or in HTML.

For text and HTML including any number of languages but no more than small amounts of CJK (Chinese, Japanese, or Korean) text, I personally recommend using UTF-8, one of the seven Unicode character encoding schemes.[17] UTF-8 has the advantage over the other Unicode encodings of working fairly well with most pre-Unicode software for the first 128 characters (basic Latin characters and punctuation marks). Since Unicode encodings allow you to encode practically any known character, it is superior to earlier script-specific encodings, such as ISO 8859 encodings, Windows codepages, and national standards such as the KOI encodings. Support for Unicode is now well-established in Internet standards, major operating systems, and office software, and

the proportion of computer users using pre-Unicode software will continue to shrink as software and hardware are upgraded.

While scholars and librarians are right to be concerned about accessibility overseas, where older hardware and software are the norm, the value of international standards and the need for long-term accessibility must be borne in mind as well. There are many utilities for converting files between character sets, as well as web forms that let you cut and paste chunks of text for conversion or transliteration, so documents in a non-Unicode character encoding scheme can be automatically converted to UTF-8 or another Unicode character encoding scheme. When creating new files, it is best to start in UTF-8 rather than convert to it later because you will never be limited by the set of characters found in any particular pre-Unicode character set; however, converting to UTF-8 at a later stage is effortless. Note that this conversion is not the same as converting from a font with a non-standard encoding (as explained in question #6), for which there are few utilities available.

I also recommend using UTF-8 with documents entirely in English to be consistent and to encourage the adoption of this superior standard.

For character set recommendations for e-mail, see question #13.

10. How do I specify the character encoding (character set) for my web pages?

There are a few levels at which encoding can be specified for an HTML web page:

a. If composing HTML in a text editor, you need to choose an encoding in which to save the file. An option may be presented to you when saving, or it might be set by default.
b. The HTML file itself can contain a declaration of the encoding being used (explained below). In HTML editors this is tied to (a).
c. The server on which the file is stored can be configured to declare an encoding when serving files to users, directing their browsers to use this encoding when displaying the file.

Step (a) must happen, even if you're not aware of it because your program for generating HTML (an HTML editor or word processor) takes care of this for you. Neither (b) nor (c) is required, but if neither is used, you rely on the user's auto-detect feature being enabled and your text

being sufficiently predictable to have its encoding guessed. There is no need to risk this when the encoding can be specified otherwise.

Popular web page editors and text editors do not default to UTF-8 for (a) and (b) because most users do not require multilingual capabilities and want their web pages to be accessible to as many users as possible, even those using outdated web browsers and operating systems. However, current editors can be customized to work entirely in UTF-8, and contemporary browsers have no problem reading files in UTF-8 encoding.

Even if you save your web documents (HTML or otherwise) in UTF-8, there are other settings which could interfere with their proper delivery online. There are four different sources of information a web browser uses to determine which character set to use to view a page, given here in order of precedence (the browser stops as soon as it finds a suitable setting):[18]

1. The value of the *Content-Type* field in the HTTP header sent by the server
2. The XML declaration (applies only to XHTML and other XML documents): <?xml version="1.0" encoding="utf-8"?>
3. This line (or an equivalent for using another character encoding) in the HTML or XHTML head element: <meta http-equiv="Content-Type" CONTENT="text/html;charset=utf-8" /> *(Exact syntax varies depending on the version of HTML or XHTML. This is an alternative to reconfiguring the HTTP header but is considered an unofficial fix.)*
4. The most recent setting in the user's browser for which character set to use (if auto-detect is not used), or the auto-detect feature of the user's browser

While the last is completely out of your control as the creator of a web page, the first can be properly configured with the help of your server administrator, and the second and third can be included in your HTML code.

11. Should I offer more than one encoding of all my documents?

Some websites offer more than one encoding of their pages because there was a time when browsers could only view certain encodings and because you can automatically generate files converted into another encoding–or even transliterated–quite easily. Today, however, in my opin-

ion there is little reason to provide your documents in more than one encoding. For multilingual documents, the only real choice right now is between UTF-8 and any of the earlier script-specific or national character encoding schemes.

12. How can I be sure others will be able to view my web page or other documents properly?

PDF is the only major format preserving the exact appearance of a document. It's a good option for providing documents meant to be printed, for guaranteeing exact appearance, and for archival preservation, but it is a bad choice for building a website.

If not using PDF, you cannot guarantee that a document provided in another format will appear for the user exactly as you intended: there are simply too many factors outside your control. However, by following the directions in questions #9, #10, and #11, you will minimize the chance that a user will not be able to view your pages. It would not hurt to try viewing your pages on a variety of operating systems using a variety of browsers to make sure there are not any problems. Doing so will help you spot not only character encoding problems but also design problems caused by browsers–even major ones–that are not fully standards-compliant.

13. What character encoding (character set) should I use in e-mail messages?

Character encoding in e-mail is even more complicated than in HTML and word-processing documents. Unicode support in e-mail software lags behind operating systems, word processors, and web browsers, so a Unicode encoding should be used only if you know all recipients of your message can view it. It is better to use a pre-Unicode encoding standard (as long as you can limit your conversation to Latin plus one additional script) or transliterate[19] everything if you must.[20]

Note that some e-mail clients and servers that can handle "exotic" character sets in the body are not able to display or even process these character sets when found in the header of the message, which includes the *to*, *from*, and *subject* fields, among others. So you may find it best to use only basic Latin characters in your subject lines and even your name as it displays in the *from* field.

14. My colleague can't read my e-mail message written in Cyrillic. Have I done something wrong, or has my colleague?

There are three types of *e-mail clients* you can use to read e-mail: a program that runs on your e-mail server (such as Pine), a *stand-alone* program that runs on your computer (such as Microsoft Outlook, PC Pine, Netscape Messenger, or Eudora), or a web interface hosted by whatever service you use for e-mail (such as Hotmail, Gmail, or a university's "webmail" service). Most, though not all, e-mail clients allow you to set the character encoding of an outgoing message. Either this setting is program-wide for all messages, or it is a default setting that can be overridden for an individual message. In the former case, if you find yourself needing to compose in a different encoding than usual, you should change the setting before you begin composing a message. Clients of all types work similarly to browsers when deciphering a received message: most have auto-detect features and allow you to override the current setting.

When using a program on an e-mail server, both sender and receiver need to have a client capable of understanding the character encoding used in the message, and their software for connecting to the server (such as SSH or Telnet) must be able to understand and display the character encoding using a font on the user's computer.

When using a standalone client, both sender and receive need to make sure their clients can understand and display the character encoding used in the message using a font on their computer.

When using a web interface, both sender and receiver need to make sure their clients can understand the character encoding and that their browsers are configured to display this encoding, as explained in question #2.

15. I can read messages from individual colleagues without any problems, but messages I receive from e-mail lists consistently give me problems. How can I fix this?

Electronic distribution lists (*electronic mailing lists* or simply *e-mail lists*) present additional problems beyond the pragmatic solutions to today's e-mail situation explained in #13. As with web pages, you can never really guarantee how your message will appear to recipients, and this is especially true for messages sent to mailing lists because the software running them intervenes while relaying your message. Users who subscribe to a list in a non-MIME digest mode receive mes-

sages forced into one character encoding, and some list software (such as ListProcessor, which runs SLAVLIBS[21]) strips the encoding declaration out of all messages. In order to guarantee that you will reach the maximum number of people, you will need to stick to Latin characters (and avoid sending HTML or "rich text" mail). Using the lowest common denominator of e-mail standards is also important for e-mail distribution lists since many are archived, and archive interfaces are not always friendly to multiple character sets.

16. Do search engines reach all documents no matter what their character encoding (character set), or do they only pick up certain encodings?

Any major search engine is bound to understand more than one encoding and allow searching across documents.

17. Now that I can type in Cyrillic on my computer, can I search online library catalogs (OPACs) in Cyrillic?

This depends on the capabilities of the software behind the OPAC, as well as on whether the records have Cyrillic text in the 880 field (in addition to the usual transliterated fields). Generally speaking, records must have Cyrillic text and the OPAC software must be Unicode-conformant. Because of the way shared cataloging works in Western libraries, it might be the case that some records have Cyrillic text while others have only transliterated text. In this case a search in Cyrillic would yield only records with Cyrillic text while a search using proper transliteration will yield all records. Therefore, users should continue searching in transliteration for the near future for languages written in Cyrillic, for which use of the 880 field among catalogers has been sporadic.

NOTES

1. I give alternative terms in parentheses following the form I prefer, which is either more closely aligned with computer, as opposed to linguistic, terminology or more precise than another computer term.

2. There are many attempts to elucidate the character-glyph distinction, as well as some to discredit it. For an analysis of medieval Slavic arguing that a language-independent distinction is ultimately unsustainable, see David J. Birnbaum, "Standardizing

Characters, Glyphs, and SGML Entities for Encoding Early Cyrillic Writing," *Computer Standards and Interfaces* 18 (1996): 201.

3. For times when appearance is significant and could be helpful in searching, texts should be encoded using *descriptive markup*, such as that prescribed by the Text Encoding Initiative <http://www.tei-c.org/>, to enable such searching.

4. Software and printer drivers for generating PDFs try to replicate the exact appearance of a printed document; however, such software is not without flaws. *Combining characters* (defined in question #6) in Unicode are known not to work in some PDF-generating software and printer drivers. Furthermore, the PDF standard allows for only the most common fonts to be used without embedding them in the PDF file, so if a PDF file uses a rare font that is not embedded, it will display as intended. See Adobe Systems, *PDF Reference, Fifth Edition: Adobe® Portable Document Format Version 1.6* (San Jose, CA: Adobe Systems, 2004), <http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf> (accessed December 22, 2004), 15-16.

5. See the website of Unicode, Inc., <http://www.unicode.org/>. Unicode is actually the common name for two separate standards synchronized since 1991: the Universal Multiple-Octet Coded Character Set (UCS), defined by ISO/IEC 10646 (JTC 1/SC 2/WG 2), and the Unicode Standard, defined by the Unicode Consortium, an industry group that allows for limited outside participation.

6. Think of the difference between *font encoding* and *character set encoding* like this: *font encoding* concerns the arrangements of glyphs within the font itself, reflected, for example, in how the characters are arranged in a character map program, whereas *character set encoding* concerns how the data (numbers) that make up the file are matched to characters.

7. Macintoshes here are the most significant exception: recent models would mostly but not entirely work with PC keyboards, but the latest models do not work with PC keyboards or even earlier Macintosh keyboards. See MacWindows Solutions, "Sharing Keyboards and Monitors Between Macs and PCs: Cross-Platform KVM Solutions," <http://www.macwindows.com/keyboard.html> (accessed September 12, 2004).

8. It is possible for both programs not to be Unicode-conformant but both to use the same character encoding in communicating with the clipboard, but this is decreasingly likely when using recent software.

9. At least one Unicode-conformant font comes with all major operating systems today. See question #1.

10. A *code point* is the unique number assigned to each character for data storage.

11. One such conversion effort, involving various Cyrillic fonts from the 1980s and early 1990s, is the Cyrillic Font Project: <http://www.stg.brown.edu/projects/indexcard/displaycard.php3?card=9> (accessed September 12, 2004). Similar work has been done by the staff of the Slavic and East European Language Resource Center <http://www.seelrc.org/>.

12. For the Slavicist, note that Unicode includes Glagolitic and all the Cyrillic characters needed for any modern language written in Cyrillic, but some historical Cyrillic characters are still missing, according to Deborah W. Anderson's presentation "Unicode in Multilingual Text Projects: A Status Report from the Script Encoding Initiative, UC Berkeley" delivered at the Joint International Conference of the Association for Literary and Linguistic Computing and the Association for Computers and the Humanities, Göteborg, Sweden, June 11–16, 2004.

13. Unicode Consortium, "*The Unicode® Standard*: A Technical Introduction"; "Unicode 4.0.0," <http://www.unicode.org/versions/Unicode4.0.0/> (accessed August

21, 2004). This number was calculated by summing figures given in these two documents.

14. This is the same principle used in MARC (where it originated), though Unicode combining characters follow base characters rather than precede them in the file's sequence of characters, as was originally the case in MARC.

15. See <http://www.w3.org/>.

16. Patrick Rourke, "Unicode Normalization Forms," *Unicode Polytonic Greek for the World Wide Web (version 0.9.7),* <http://www.stoa.org/unicode/normalization.html> (accessed September 23, 2004).

17. *Wikipedia: The Free Encyclopedia*, s.v. "UTF-8," <http://en.wikipedia.org/wiki/UTF-8> (accessed September 23, 2004). CJK ideographs are more efficiently stored in UTF-16.

18. Web Standards Project, "Specifying Character Encoding," <http://www.webstandards.org/learn/askw3c/dec2002.html> (accessed September 13, 2004); Alan J. Flavell, "Notes on Internationalization," <http://ppewww.ph.gla.ac.uk/%7eflavell/charset/internat.html> (accessed September 15, 2004); Richard Ishida, "Tutorial: Character Sets & Encodings in XHTML, HTML, and CSS (DRAFT)," <http://www.w3.org/International/tutorials/tutorial-char-enc/> (accessed September 15, 2004).

19. What transliteration scheme you use is up to you. Ideally you will use a *reversible* transliteration scheme–one that can be de-transliterated automatically. Library of Congress (LC) Romanization schemes are reversible if combining characters are used (difficult to represent outside of MARC software), as are proposed standards like Russkaja Latinica <http://www.kulichki.com/centrolit/rl/latinic1.html>. Note that multiple systems of transliteration, like multiple character encodings, are a barrier to cross-searching and future reuse of material. (See question #6.)

20. Use of Unicode in e-mail is less important than in other electronic documents for which long-term accessibility and preservation are of greater concern. While social historians may one day be interested in your correspondence and having some trouble deciphering electronic records of it, they will have to decipher everyone else's as well and will be quite good at it.

21. For more information on SLAVLIBS, see "Internet Links 'As Seen on SLAVLIBS,'" <http://www.columbia.edu/~jsi19/slavlink.html>.

SELECTED BIBLIOGRAPHY

Czyborra, Roman. *Czyborra.com.* <http://czyborra.com/> (accessed September 12, 2004).

This personal web page contains links to many comprehensible sub-pages on multilingual computing. While some are aimed at computer experts and focus on Linux and the pages are generally a bit outdated, the historical information is quite well-digested. Unfortunately, the site is frequently unavailable.

Drozd, Andrew M. "Slavic Fonts and Keyboard Drivers." *American Council of Teachers of Slavic and Eastern European Languages*. <http://www.aatseel.org/keyboards.html> (accessed September 24, 2004).

This section of the AATSEEL website contains links for fonts and keyboards (including homophonic keyboards), but as explained, there is very little you need to download if you are using recent software.

Gorodyansky, Paul. *Cyrillic (Russian): Instructions for Windows and Internet*. <http://www.ruswin.net/> (accessed September 24, 2004).

This fully bilingual (English and Russian) site contains exhaustive, accessible information about "russifying" many versions of Windows, including installing a homophonic keyboard created by the author.

Il'in, Igor'. *Translit.ru*. <http://www.translit.ru/> (accessed September 24, 2004).

This online tool allows you to type in Latin characters and have them converted to Cyrillic based on the transliteration scheme built in. You can copy and paste in text for transliteration or de-transliteration.

Korpela, Jukka "Yucca." "Characters and Encodings." *IT and Communication*. <http://www.cs.tut.fi/~jkorpela/chars/index.html> (accessed September 24, 2004).

The subpages from this site contain a huge amount of highly accessible information, including historical perspective. Especially useful is "A Tutorial on Character Code Issues."

Main, Linda. *Building Websites for a Multinational Audience* (Lanham, Md.: Scarecrow Press, 2002).

This book explains basic terminology in depth and explains how to use current and emerging web standards. Particularly useful to librarians is Appendix A: Library Automation Vendors and Unicode Compliance.

Mashkevich, Stefan. *Automatic Cyrillic Converter*. <http://www.mashke.org/Conv/> (accessed September 24, 2004).

This online tool allows you to type or paste characters in any Cyrillic encoding or in one transliteration scheme and convert them to another encoding (or to transliteration). You can also upload whole files for conversion, or convert the page at a given URL.

Microsoft Keyboard Layout Creator (MSKLC) Version 1.3.4073. <http://www.microsoft.com/downloads/details.aspx?FamilyID=fb7b3dcd-d4c1-4943-9c74-d8df57ef19d7&displaylang=en> (accessed September 24, 2004).

This program, available from Microsoft for Windows 2000, Windows XP, and Windows Server 2003, allows you to make your own keyboard layouts.

Palchuk, Matvey B. *Russification of Macintosh*, <http://www.friends-partners.org/partners/rusmac/> (accessed October 15, 2004).

This site in English gives full russification instructions for many versions of Mac OS and common software. As with Windows, many of these steps are no longer required when using recent operating systems and programs.

Rourke, Patrick. "Unicode Polytonic Greek for the World Wide Web: Version 0.97: Draft." *Stoa Consortium*. <http://www.stoa.org/unicode/index.html> (accessed September 24, 2004).

Though this resource has not been updated since 2002, its clear description of Unicode for the non-specialist is still highly useful. It is useful to those using Unicode for languages other than Ancient Greek.

Sidorenkov, Igor. *Otpad 2.3–Russian Text Editor*. <http://www.ingenit.com/home/programs/otpad/index> (accessed September 24, 2004).

This free Windows text editor that has built-in transliteration and encoding conversion capabilities.

Wood, Alan. "Unicode and Multilingual Support in HTML, Fonts, Web Browsers and Other Applications." *Alan Wood's Unicode Resources*. <http://www.alanwood.net/unicode/> (accessed September 24, 2004).

This site is updated regularly and very easy to use. There are many pages to test encodings, as well as the best listing of freely available Unicode fonts on the web. (If using recent software, you do not need to download a special font. See question #1.)