Guide to Unicode Greek

by Joel Kalvesmaki, 2012

Introduction

All authors publishing with Dumbarton Oaks must submit text that is Unicode compliant. This policy applies to the submission of text in any language, not simply polytonic Greek. This introductory guide explains, with particular reference to the Greek language, the development and architecture of Unicode, methods for working with Unicode, and related issues. It is aimed at anyone new to Unicode or anyone working with non-Unicode fonts (Greek or not). Suggestions for further reading can be found at the end of this guide.

What is Unicode?

To fully appreciate Unicode it helps to start with the telegraph. The earliest versions transmitted and received electronic pulses, which were transcribed onto long paper strips. As the voltage changed, either a pen wiggled across the paper, or a set of needles or stamps punctured or indented it, creating a visual pattern representing the message. (The audio component—the clicks and clacks we associate with telegraphs—came later.) To interpret the message it was necessary that both sender and recipient use the same code, so the pulses could be converted to letters or numbers. These code tables were the prototypes of the code tables used in computers.

The initial codes consisted of a limited set of characters. International Morse Code, for instance, has only fifty-one characters: the twenty-six letters of the English alphabet (assumed to be uppercase), the ten digits, and fifteen signs of punctuation. All the earliest telegraph codes possessed a limited number of characters. Small character sets were cost effective, reducing the number of dots and dashes, and therefore the time, needed to encode, transmit, and decode a message.

As technology developed, new communication devices introduced more expansive code tables, ones that included lowercase letters, punctuation, and commands (such as “start new line” or “end of transmission”), but were frequently inconsistent with the code tables of other devices. The proliferation of different coding systems compelled the International Organization of Standards (ISO) to develop a single standard for coding telecommunications. Thus was born in the 1960s the American Standard Code for Information Interchange (ASCII), a 128-character code that included upper and lowercase letters, the digits, standard punctuation, and commonly used command codes. The size was chosen for its efficiency and its compatibility with binary systems ($128 = 2^7$).

IBM and Apple made ASCII the basis for the character sets of their new computers. But these systems transmitted information in packets of 8 bits, so 256 ($= 2^8$) was the natural size for the code table. Both IBM and Apple developed a 256-character set, populating the extra places with characters absent in ASCII. The two systems were developed independently, so texts using the upper 128 characters on one system could not be read on the other. Language was an issue too. The character sets computers used through the mid-1990s catered almost exclusively to the Latin alphabet. Both PC and Macintosh upper character sets (spaces 128–255) assigned some slots to Greek letters, but these were intended to serve mathematicians, not users of modern or classical Greek. Thus, in the 1980s and 1990s anyone who wanted to use computers to work with Greek or other alphabets had to invent creative ways of getting around the Latin alphabet.

Most often, the way to type non-Latin alphabets was simply to work with a customized font. By all appearances, this seemed to imitate what had been done in the past, where printers would swap out and mix sets of lead blocks. With these customized fonts, when a Latin letter was typed, what appeared was a Greek letter assigned its place. For instance, by pressing the l key, a λ would appear. The underlying data was still a Latin

letter, but it looked Greek. This technique made the most of the 256-character set, but it led to other hassles and problems:

- To share a Greek text either the author had also to send (usually illegally) the font or the recipient had to buy a copy.
- Greek letters were assigned arbitrarily. Some fonts put the χ on the x, others, the c; a rough breathing was a left parenthesis on one font and a J on others. Some fonts assigned precomposed combinations of vowels and their diacritical marks to arbitrary, hard-to-remember places in the upper character set. Other fonts split vowels and their individual accents. Every font required the user to learn a new keyboard configuration. And it made accurate text searches across texts with different fonts nearly impossible.
- A text set in a font using the upper 128 characters was not legible in a different operating system, compounded by incompatible font technologies used on each platform (introducing Mac-versus-PC compatibility spats, common especially in the 1990s).
- Even the most beautifully designed Greek font produced shoddy typography. For instance, a rough breathing correctly centered over a lowercase eta (ἡ) was off-center with an iota (᾽ι). An iota subscript properly centered under an omega was not correctly positioned under an eta, which takes the iota subscript under its left leg. The practice was a setback in fine typography.

These are very serious problems. And for every challenge Greek poses, there are dozens more in other alphabets and languages, such as Chinese, Arabic, and Tibetan. In the late 1980s Xerox and Apple attempted to standardize Han Chinese. Other corporations (e.g., Sun, Adobe, Microsoft, and IBM) joined the effort, newly broadened to address all human languages, and in 1991 led to the establishment of the Unicode Consortium, a nonprofit organization that supports and develops the Unicode Standard. Their intent was simple, to develop a character code that addresses every need in every language, a code that is universal, uniform, and unique (hence the name).

Each and every character in the world's writing systems, living and dead, is assigned a unique, unambiguous code point. The uniqueness means that the method of entering Greek text is no longer an issue. An ἄ, no matter how it is typed, will be the same character on everyone's system, independent of fonts. And texts can be shared, without worry about recipients having the same font used for its composition.

The Unicode character map has 1,114,112 (2^{32}; i.e., 32-bit code) points, to accommodate all possible scripts, living and historical. It works on a hexadecimal system (0-9 and A-F), not the decimal system, starting from zero and going through 10FFFF (hexadecimal for 1,114,112). The Latin k is assigned the Unicode value 6B (= decimal 107) and the Greek ϰ is Unicode value 3BA (= decimal 954). The standard way to refer to a Unicode character is with U+ and a four-digit hexadecimal number. Thus, k and ϰ are U+006B and U+03BA respectively.

It is helpful to think of the Unicode character map as a very long ribbon, sixteen characters wide. That ribbon is divided into blocks (also called code tables), of varying lengths but always in multiples of sixteen. Blocks are devoted to specific scripts, or types of scripts (e.g., Basic Latin, Latin-1 Supplement, Greek and Coptic, Armenian, etc.). Some scripts, such as Latin and Greek, fall into more than one block, and are not placed next to each other (usually reflecting the sequence of the development of the Unicode standard).

Classical and Byzantine Greek falls principally in two blocks: Greek and Coptic (assigned U+0370..U+03FF; see official code chart here) and Greek extended (U+1F00..U+1FFF; official code chart). Specialized Greek characters fall in other code tables:

- Combining characters such as underdots (to express doubt about a letter) and overbars (to show suspension marks) appear in Combining Diacritical Marks (U+0300..U+036F; official code chart).
- Some manuscript punctuation marks are found in General Punctuation (U+2000..U+206F; official code chart).

- Greek characters relevant to papyrology and inscriptions (especially relevant to Byzantine weights) are found in Ancient Greek Numbers (U+10140..U+1018F; official code chart).
- Byzantine music is represented in Byzantine Musical Symbols (U+1D000..U+1D0FF; official code chart).

In the future Unicode will expand (currently it is at version 6.1), but it will preserve every previous standard. Byzantinists should watch these developments, since occasionally new symbols are introduced for inclusion in Unicode, in Greek and in other languages.

Working with Unicode Greek

Unicode is a central part of all modern computers, so you are likely already running it on one level or another. To work with Unicode Greek, it helps to check four areas that must cooperate: the operating system, the fonts, the keyboard driver, and the word processor.

Operating system. Macintosh users running OS X are fine. Previous versions (OS 9 and below) will not work. PC users must run Linux or Windows CE, NT 4.0, 2000, XP and any other more recent operating system. Many older operating systems, such as Windows 98, do not work. If you have an old computer that cannot run the latest software, then it is time for you to upgrade to a new system.

Fonts. Fonts normally support only a small subset of Unicode characters. So you must work with fonts that include polytonic Greek. If your operating system is Unicode compliant (see above), you already have useful fonts for polytonic Greek: Lucida Sans, Palatino Linotype, Tahoma, and Arial Unicode MS (all PC) as well as Lucida Grande, Helvetica, and Times (Mac). To find other fonts on your system that support polytonic Greek, use a character browser. On the PC, try the free program BabelMap. On the Mac, use the character map (first accessed through System Preferences > Language & Text > Input Sources; thereafter accessed in the top right corner of your screen). Scroll down to the Extended Greek block, pick one of the characters, and a feature in the program should tell you what installed fonts support that character.

There are many other Unicode-compliant Greek fonts available, many for free. Many Dumbarton Oaks publications use Adobe Garamond Premiere Pro and Minion Pro. An internet search for other Greek fonts will give you more options. Make sure, however, that the font you install supports the Greek Extended code page (see above).

In the past, Macintosh and PC fonts could not be interchanged, but that is now a nearly obsolete problem. Any font that works on a Mac (OS X) works on a PC (Windows XP and higher), and vice versa, thanks to the font standard called OpenType, which supersedes and combines the formerly incompatible TrueType and PostScript font standards. If you want to find out if a font you are using is Unicode compliant try this simple test. In your word processor select some Greek text set in the font. Change the font to another that you know has Unicode Greek. If the result is legible, then the original font is indeed Unicode compatible. If the result is a series of boxes or gobbledegook, the font is most likely not Unicode compliant. You should probably discontinue use of that font.

Keyboard driver. Let's say you want to type in Greek. Although Unicode has expanded its character set to over one million characters, our keyboards still have little more than a hundred keys. In the days of the 256-character set every character was only one or two keystrokes away. To use different blocks of Unicode, you need to work with different keyboard drivers, which interpret the keys being typed. Users may install multiple keyboard drivers, and switch languages as they type.

Macintosh OS X has a built-in polytonic Greek option (System Preferences > Language & Text > Input Sources). If you don't like the layout there, try SophoKeys, a free keyboard driver.

Windows XP and later have a polytonic Greek keyboard driver, which you must activate yourself (Control Panel > Regional and Language Options > Languages). You may find alternative keyboard layouts here. One other alternative is MultiKey. (Bear in mind, MultiKey is not a true universal keyboard driver, since it works only for Microsoft Word and Classical Text Editor.)

Most of these keyboards are context sensitive. That is, the drivers are able to convert key combinations into single glyphs on the fly. For example, an alpha will change its shape as you continue to add diacriticals to it, but still remain only one character. Or if you type sigma then press the space bar the closed sigma will automatically turn into a final sigma. Bear in mind that the keyboard driver is a means to an end, and as long as the end result is Unicode-compliant polytonic Greek, it does not matter what method you use. Experiment with different keyboard drivers and use the one you find most comfortable.

Unusual characters. Let's say that you have certain Greek characters you want to drop into your word processor or email, but you can't easily type them (e.g., the Greek numeral 6, ☐).

On the PC, try the free program BabelMap, which gives you access to all the Unicode characters, and lets you search for a particular character by name. You can pick characters and copy them from the text buffer, to be pasted in any other context.

On the Mac, use the character map or Emoji & Symbols tool (first accessed through System Preferences > Language & Text > Input Sources; thereafter accessed in the top right corner of your screen, via the icon that indicates what language you're typing in). When you get there, make sure you click the cog in the top left corner and choose Customize List... Once you're there, pick Greek and, even better, Unicode. Double-clicking on any character will insert it into whatever application you're typing in at the time.

Word processors. If your word processor is less than ten years old, it's probably already Unicode compliant, e.g., Microsoft Word, Open Office, Google Documents, or one of the many other etc. (Some of these programs may not support other parts of Unicode, such as right-to-left languages such as Syriac or Arabic.) WordPerfect is not at all compatible with Unicode Greek, and we strongly discourage its use.

Frequently asked questions

Will Unicode become obsolete any time soon?

No. It may seem that one standard has been abandoned for another. But this is not the case. Unicode has not rendered the older system, ASCII, obsolete. ASCII is still an industry standard, but now as a proper subset of Unicode. What is being rendered obsolete is the incorrect use of ASCII. Scholars have been trying to use ASCII to do things for which it was never designed. Unicode has been designed specifically to address the needs of those who have had to make ASCII do what it was not intended to do.

I have already committed large amounts of Greek to a font that is not Unicode compliant. Does this Greek need to be retyped?

No. There are resources that allow you to convert your preexisting Greek into a Unicode-compliant format. You may be able to make this conversion on your own. There is a website that provides conversion of a few fonts. GreekKeys Converter facilitates other kinds of conversion for Mac OS (see helpful guide here). Other utilities for the PC are listed here.

If none of these resources work, do not despair. There probably is a way to make the conversion. Contact the publications office, present a sample of your text, and we will suggest a way to handle the material.

I need to type Byzantine Greek letterforms and symbols that don't seem to be in Unicode, such as the OY ligature and an uncial omega that is shaped like the minuscule one (ω), not a Ω. What do I do?

Unicode sharply distinguishes between characters and glyphs. Characters pertain to semantics and meaning; glyphs are specific instantiations of characters. So Unicode provides a place for the uncial omega, but it does not concern itself with how a font might shape the letter. That is a good thing, because whether or not a character looks like a Ω or a ω, it means the same thing, and you don't want two different code points for the same meaning. Unicode also does not encode ligatures (grandfathered exceptions notwithstanding). That too falls under the purview of font design.

Dumbarton Oaks has a special inscription font, Athena Ruby, a Unicode-compatible font that supports variant letterforms. Other custom fonts, such as Cardo, IFAOGrec Unicode, Junicode, and New Athena Unicode, may have the non-Unicode glyphs you want. Note, however, that these glyphs are usually placed in the Private Use Area (PUA) block of Unicode. The PUA is provided for arbitrary glyph placement, but the use is not encouraged. Information given to the PUA of a font will not be readable outside that font.

For further reading

Introduction to Unicode Greek, by Rodney J. Decker. Although written for Biblical scholars, this site is useful to classicists and Byzantinists.

Official site for Greekkeys, American Philological Association. This site distributes various Unicode fonts such as KadmosU, BosporosU, and AtticaU. These fonts are especially helpful for Copticists, since they include the Coptic code page (separate from the Greek and Coptic block).

List of Unicode Greek fonts available, courtesy Russell Cottrell.

Very good explanation of Greek Unicode; the free Unicode font distributed here (Cardo) incorporates Greek, Hebrew, and Latin.

A long page from the TLG listing available Unicode Greek fonts. See the subpage on how to configure your web browser to view Unicode Greek on the internet. See the extensive discussion about Unicode and Greek, addressing specific points of Greek philology, such as the two types of qoppa.

The official website of the Unicode Consortium, with a plethora of information about the Unicode standard. Of interest to Byzantinists will be their section on Greek.