

Before and After Unicode: Working with Polytonic Greek¹

Donald Mastrorarde, U. of California, Berkeley

After the age of the punch card, when input and output for computing turned to the human-readable form of numbers and letter, these processes were very much English-centered. Computer engineers and programmers had little interest in the needs of multilingual and multiscript texts. When character sets other than plain-vanilla US English did become available, each set was limited in size to 256 items and the real limit is more like 220. For many purposes, only 128 items could be handled more or less gracefully, while the items in the upper half of a larger set might be ignored or misinterpreted by some programs. When the TLG originally began digitizing polytonic Greek texts, the betacode transcription and an elaborate collection of beta escapes were needed to cover the multitude of characters and symbols in specialized texts.

The introduction of the Macintosh in 1984 inaugurated a new era for those who wanted to use fonts for specialized scripts: users could create and edit bitmap fonts and print them without difficulty on the ImageWriter, Apple's dot-matrix printer, and they could hack the Mac's system resources to adapt keyboard input for the customized font. The late George Walsh recognized the potential benefit to Hellenists and released SMK GreekKeys soon after the appearance of the first Mac. The same capability attracted users of other non-Roman scripts.

¹ This is a slight revision (Feb. 2008) of the presentation made at the panel "Fonts, Encodings, Word-Processing and Publication: a tutorial for classicists on fonts and Unicode" at the Annual Meeting of the American Philological Association in Montreal, January 2006. In this PDF version the illustrations from the PowerPoint slides are interspersed with the text.

Each font contains within it an encoding, that is, a scheme for identifying each character numerically. The user strikes a key on the keyboard and thus generates a numeric code, and this code tells the operating system to display the corresponding item in the current font. In the many special fonts, including those developed by George Walsh for GreekKeys, the encoding was *ad hoc*, invented by the end-user: such fonts pretended to be organized as a standard Macintosh Roman font, but in reality placed other characters in some or all of the 220 or so positions available in fonts at that time.

This explains why in such encodings you can change the font to a normal Roman font like Times and see ordinary Roman characters mixed with odd symbols or Roman letters with diacritics. But if you turn one Greek font into another Greek font that has a different encoding, the result is close to gibberish. The incompatibilities between the different encodings are obvious, and even more trouble may arise when a file is transferred from one platform to another.

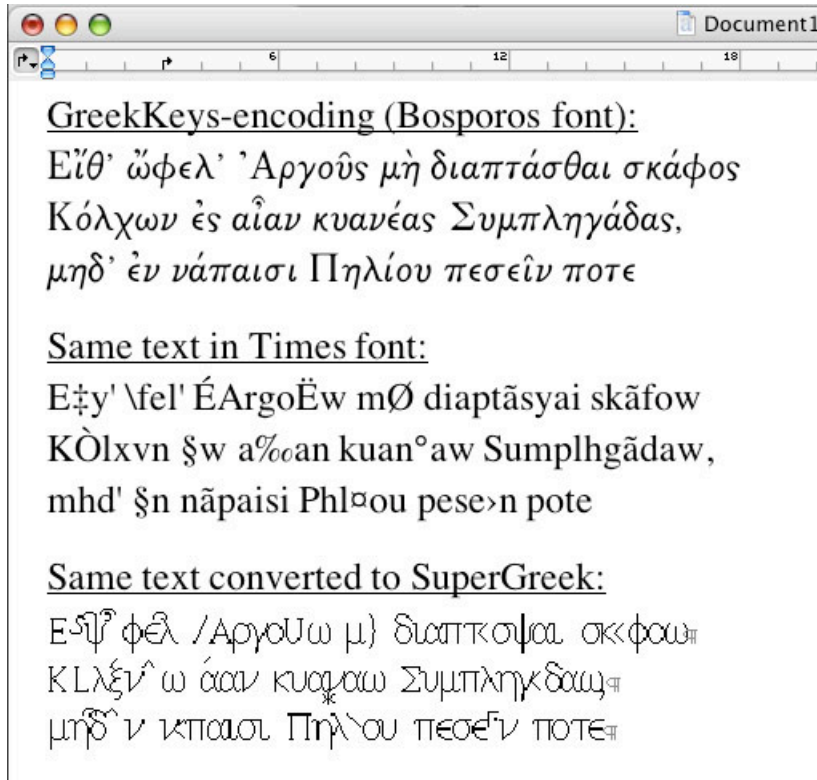


Fig. 1: GreekKeys-encoding with and without a GreekKeys font

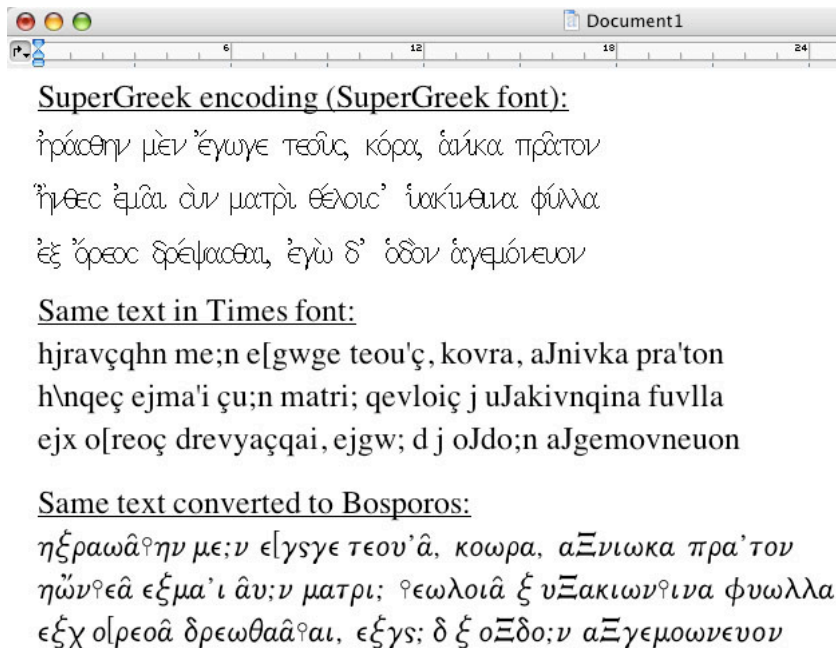


Fig. 2: SuperGreek encoding with and without a SuperGreek font

Look at the list of encodings that the web-based TLG or Perseus offers to cope with or that GreekKeys Converter offers to convert.

Welcome to the Online TLG®

Greek, a polytonic Greek font must be installed on your computer.

[DOWNLOAD GREEK FONTS](#)

Input and Greek Display Options

display and input,

OR

B. Input in Beta or Transl./Greek Display

If your browser does not support Unicode input, choose your display Greek font below. Your input may be in Beta or Transliteration. The font you select will be the default font for your session. [\(HELP\)](#)

Choose your display Greek font

- Beta Code
- Translit (no acc)
- Translit (w/ acc)
- Unicode
- GreekKeys
- WinGreek
- SGreek
- Ismini
- SPlonic
- SuperGreek
- LaserGreek

preferences to be automatically pre-selected in all sessions, you can [establish a User Profile](#).

[LOGIN OR ESTABLISH A NEW USER PROFILE](#)

Fig. 3: TLG font choices

Perseus Display Configuration

Use this page to set defaults for various features of the Perseus text system at Perseus. When you make your selections, click on "Set configuration". Your browser may ask if you want to accept a "cookie"; you can keep your settings from one session to another.

Word Study Links

Do you want words linked to the Word Study Tool? This tool gives a morphological analysis, information, and a short dictionary definition for Greek and Latin words. For more information, see our [page](#).

Yes
 No

Greek display

What encoding do you want for Greek text? For more information, see our [font help page](#).

Latin transliteration SuperGreek
 Greek transliteration Beta code
 GreekKeys Unicode (UTF-8)
 Sgreek for Windows Unicode (UTF-8) with pre-combined accents
 SP Ionic

Fig. 4: Perseus Greek display choices

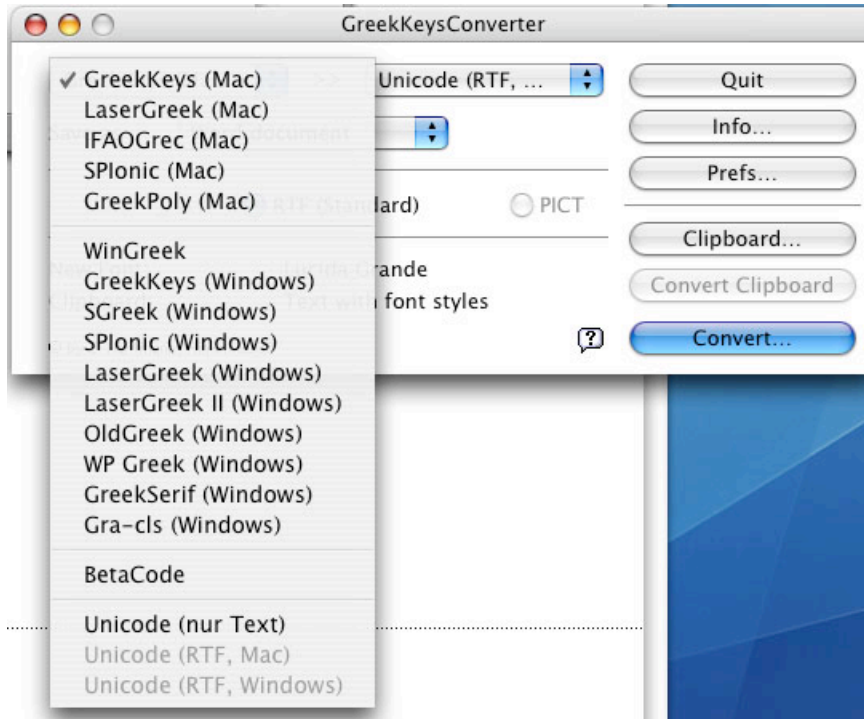


Fig. 5: GreekKeysConverter encoding choices

A secondary consideration for those who want to use such non-standard fonts is keyboard input. In terms of ideal human interface design, the input scheme should be a service of the operating system, so that it is available in any application in which the user may wish to enter the specialized characters. The Macintosh OS has always had such capability. If you install the traditional GreekKeys Universal keyboard resource or the GreekKeys Unicode Input, then in principle such inputs are available to all applications that comply with protocols of the operating system. Windows was slower to provide such each of use, but now a Polytonic Greek keyboard using the old manual-typewriter scheme is supplied in Windows 2000/XP, and the tool Microsoft Keyboard Layout Creator allows the creation of customized keyboards, although its current version leaves something to be desired in terms of versatility and its products, even if designed with care,

may be hampered by incompatibilities in Word for Windows. With the help of version 1.4 for Microsoft Keyboard Layout Creator, GreekKeys Unicode keyboards have been produced as part of GreekKeys 2008 that are very close in functionality to the Mac Unicode keyboards.

Keyboard layouts feature two styles of dealing with diacritics. [slide 7] From the beginning, GreekKeys adopted the deadkey protocol: this is the same mechanism by which on some manual typewriters an umlaut or acute or grave or tilde could be added to a standard Roman character by entering it with a deadkey or non-advancing key before the regular character. An alternative choice is to use a ligature system or zero-width diacritic characters entered after the regular character. This is the technique used by SuperGreek. While this avoids the need for a separate keyboard resource, this method often produces less than ideal placement of the diacritics, and the encoding of course remains non-standard and therefore should be discouraged now that Unicode is robust enough and widely supported enough to be used for all polytonic Greek purposes.

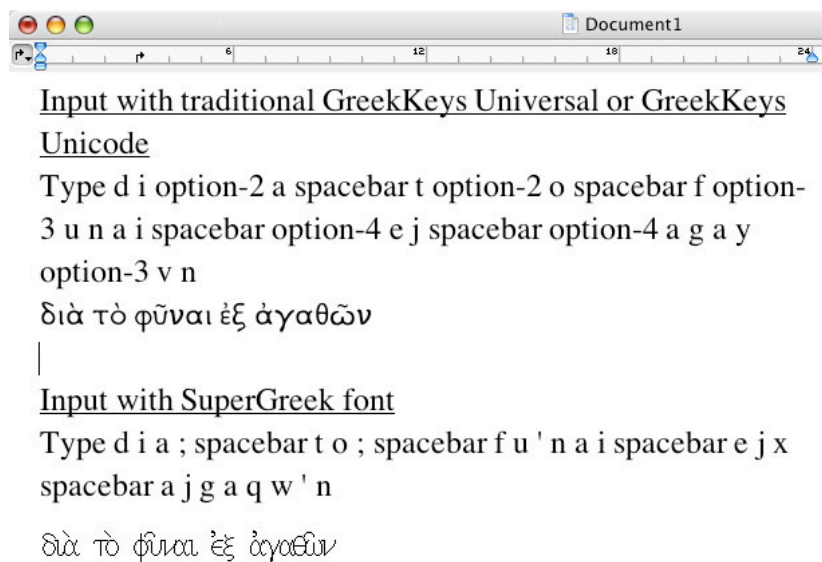


Fig. 6: Options for entry of characters with diacritics

Now let me turn to Unicode. [slide 8] Unicode is an international standard that aims to provide a unique digital identification or code point for the characters and symbols used in most of the world's writing systems and publishing traditions. [slide 9]

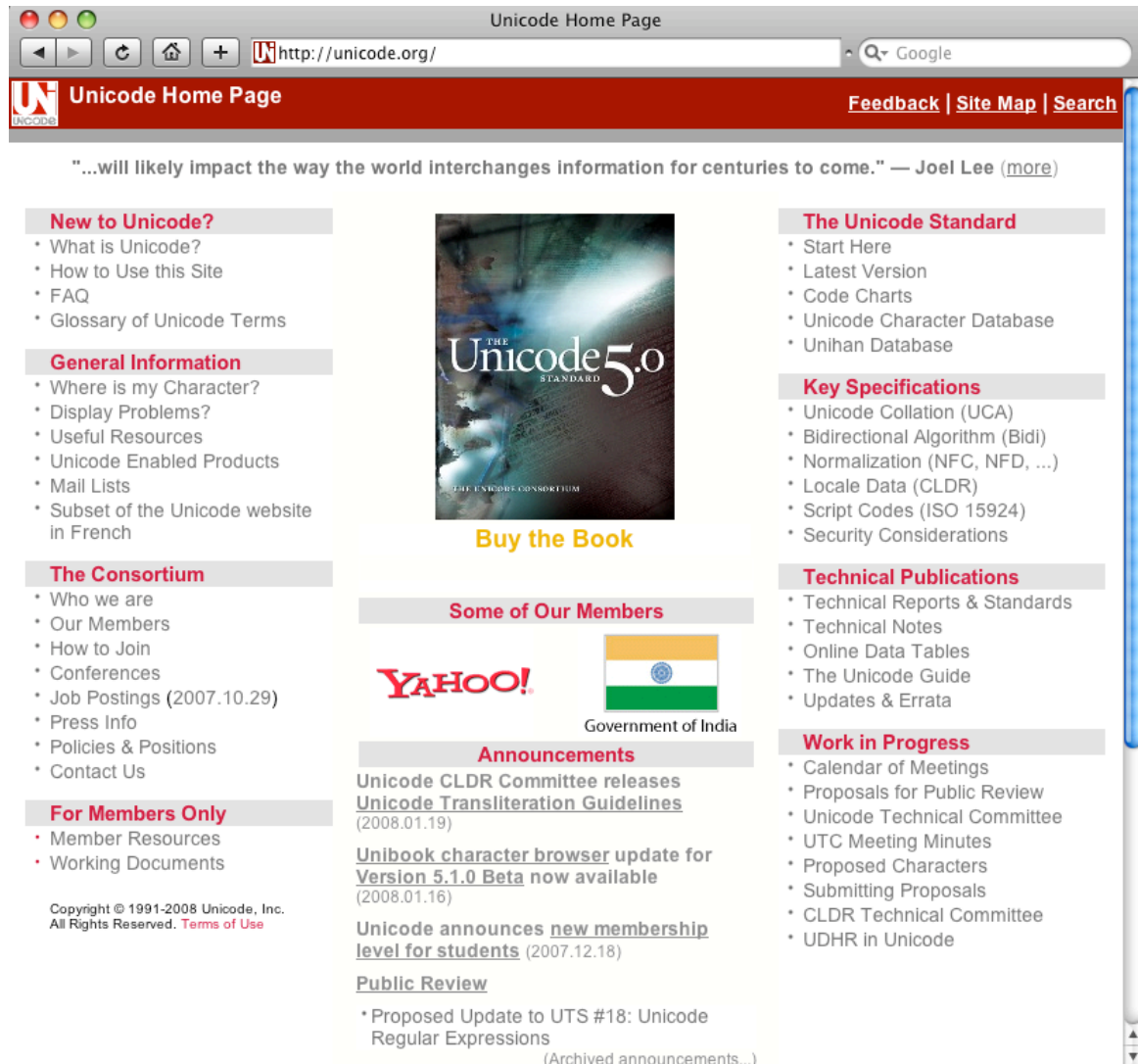


Fig. 7: Unicode.org home page

The code points are assigned as hexadecimal numbers of four or five digits. Hexadecimal notation uses the arabic numerals 0 through 9 and the Latin letters A through F. The four-digit code points belong to the Basic Multilingual Plane or BMP or Plane 0, which was originally to contain everything in the standard. [slide 10] But as time has passed, it has become

apparent that many more code points are needed for a truly global treatment of character sets, and five-digit code points have been introduced. Unicode is also divided into blocks, sets of characters and symbols that are related or that were proposed as a group.

Unicode Code Points

Character Name	Sample	Unicode hexadecimal code point
Latin capital letter a	A	U+0041
Latin small letter a	a	U+0061
Greek capital letter alpha	Α	U+0391
Greek small letter alpha	α	U+03B1
Latin capital letter d	D	U+0044
Latin small letter d	d	U+0064
Cyrillic capital letter de	Д	U+0414
Cyrillic small letter de	д	U+0434

Unicode Planes

Basic Multilingual Plane	BMP = Plane 0	U+0000 to U+FFFF
Supplementary Multilingual Plane	SMP = Plane 1	U+10000 to U+1FFFF
Supplementary Ideographic Plane	SIP = Plane 2	U+20000 to U+2FFFF

Unicode Blocks of Interest to Classicists

Name of Block	Code points start at (hexadecimal)
Basic Latin	0000
<i>basic roman characters and simple punctuation (type English and Latin)</i>	
Latin-1 Supplement	0080
<i>roman letters with acute, grave, circumflex, umlaut, and additional symbols (type French, German, etc.)</i>	
Latin Extended-A	0100
<i>roman vowels with macron or breve, letters with diacritics used in Central European languages (type Latin with macron or breve)</i>	
Combining Diacritical Marks	0300
<i>accents, macron, breve, etc.</i>	
Greek and Coptic	0370
<i>Greek for monotonic Greek, plus a few math/science symbol versions of Greek and a few distinctive Coptic letters</i>	
Latin Extended Additional	1E00
<i>roman letters with underdots, underlines, and the like (Demotic Egyptian transliteration)</i>	
Greek Extended	1F00
<i>precomposed characters for polytonic Greek, and some archaic letters</i>	
General Punctuation	2000
<i>additional symbols, such as obelus, curved quotes</i>	
Miscellaneous Technical Symbols	2300
<i>metrical symbols at 23D0 ff.</i>	
Private Use Area	E000
<i>used in GreekKeys Unicode fonts and some other scholarly fonts for precomposed characters like epsilon with circumflex, alpha with breve and smooth and acute</i>	
Linear B Syllabary	10000
Linear B Ideograms	10080
Aegean Numbers	10100
Ancient Greek Numbers	10140
Old Italic	10300
Cypriot Syllabary	10800
Byzantine Musical Symbols	1D000
Ancient Greek Musical Notation	1D200

The early history of Unicode was marked by some poor choices made for reasons of politics, economy, ignorance, or backwards compatibility. Thus Greek was initially served by a block called Greek and Coptic, starting at U+0370, and this block served only the needs of monotonic Greek (and was also considered an inadequate provision for Coptic by the Coptic community). For scholarly purposes, an additional block called Greek Extended was later added at U+1F00, and this serves many, but not all, of the needs of those who deal in polytonic and ancient texts. Additional needed characters occur in other blocks in the 4.1 and 5.0 version of the standard, and Coptic has now received its own block. Some of the special added symbols have been encoded in the Supplementary Multilingual Plane (SMP or Plane 1), which can make their use problematic in some applications that have been written to expect and process only four-digit code points from the Basic Multilingual Plane.

What difference does the maturing of Unicode make to the situation for scholars using polytonic Greek? First, there is a real hope of better and simpler communication, in documents, email, and in browsers. A user with a modern OS probably has a default system font containing both Greek and Greek Extended, and modern programs are ready to display Unicode Greek without additional installations or fussy configuration. Second, there is also hope for some sort of long-term survivability (or the best that we can do in this regard in the ever-shifting world of information technology and telecommunications): what we create today using Unicode should remain compatible and readable for a long time. And third, it gives us an alternative to systems that are breaking down. With Unicode, the incompatibilities shown on the slide are gone, and important features of the software are restored to usefulness.

Incompatibilities of Traditional GreekKeys Encoding

- compulsory display of white space for the encoding that George Walsh had selected for omega with smooth and acute (in modern applications)
- conflict with autotext features of MS Word (such as initial capitalization)
- conflict with MS Word automated smart quotes feature
- MS Word from version 6 onwards unable to search correctly for many Greek letters with diacritics
- MS Word unable to interpret where Greek words begin and end

In most of the remainder of this presentation, I want to talk about the practicalities for those who are making the transition to Unicode Greek. If you have typed whole books of alternating English and Greek, as I have, or just typed articles and handouts for classes with that same alternation, then you perhaps arrived at a habitual practice similar to the following. With traditional GreekKeys, I assign a keyboard command in Word to the Roman font in which I want to type English, Latin, etc. and a different command in Word to the Greek font that I want to use. I set the input to GreekKeys Universal and can leave it there almost always, since most of the optional features of the Roman font, such as umlauts and accents on vowels, work normally with GreekKeys Universal. I am compelled to change to the US keyboard only in the event that I need an option character on the top row, such as the section symbol (option-6) or the en-dash (option-hyphen). Here is an example, showing the places where I need to issue a single keyboard command to toggle fonts at each transition between typing English and typing Greek.

Typing with Traditional GreekKeys

For the glance of the bull as emblematic of ferocious anger, see 188 below and Ar. *Frogs* 804 (of angry Aeschylus) **#ἐβλεψε γοῦν ταυρηδὸν ἐγκύψας κάτω#**; also related are the rolling eyes and askance gaze of the agitated, resisting bull in *Hel.* 1557-8 or of the maddened Heracles likened to a bull in *Her.* 868-9, and the playful use in Plato *Phaedo* 117b (Socrates) **#ὥσπερ εἰώθει ταυρηδὸν ὑποβλέψας#**.

indicates change of font; GreekKeys Universal keyboard is left active throughout the typing.

We saw earlier that traditional GreekKeys and SuperGreek masquerade as Roman fonts, so that the Greek can be shifted to Roman characters. When you use Unicode Greek, however, it is no longer the case that changing the Greek to another font will produce Roman characters.

Font Change with Unicode

KadmosU font

superlative with ὥς (δύναται or the like understood)

converted to Times (OS 10.4 only)

superlative with ὥς (δύναται or the like understood)

converted to Lucida Grande

superlative with ὥς (δύναται or the like understood)

Either the new font will also have the same Greek Unicode characters, or it will have only some, as in the case of a font containing a few letters used in math or science or a font containing only the original Greek block of

Unicode and not Greek Extended. In some circumstances, you will see boxes displayed instead of characters, indicating that the selected font does not contain the needed code points.

converted to Helvetica (OS 10.4 version)
superlative with ὤς (δύναται or the like understood)

converted to Helvetica (OS 10.3 version)
superlative with □□ (□□□□□□□□ or the like understood)

converted to Times New Roman (in OS 10.3)
superlative with □ς (δύναται or the like understood)

Fig. 8: Font Change with Unicode

Sometimes, however, the operating system or application will display the correct letters but in a different font, a default system font that is used as the last resort to display rare code points. In this case, one often cannot change the font of the characters at all.

Times in PowerPoint 2004, Lucida Grande mixed in
ἀγαθός εἶμι

Fig. 9: Mixed Fonts with Unicode

The uniqueness of the digital identification of each character gives a new ease to moving textual material across applications and across platforms, although there are still a few glitches because of faults in the OSes or in the applications. What becomes a little more cumbersome, on the other hand, is the process of input when you are frequently changing from English to Greek and back. Except for brief documents and short handouts, I strongly recommend that you enter your English text and your Greek text in

separate fonts, as this can save a lot of work later in the process of editing and publishing. In this case, in order to type with GreekKeys Unicode or any other similar input, you must make a double command at each transition in your writing. Thus, if I were to type the same sentences shown earlier with Unicode Greek, I would have to change my input and change my font at each transition, using two separate commands.

Typing with GreekKeys Unicode

For the glance of the bull as emblematic of ferocious anger, see 188 below and Ar. *Frogs* 804 (of angry Aeschylus) @#ἔβλεψε γοῦν ταυρηδὸν ἐγκύψας κάτω@#; also related are the rolling eyes and askance gaze of the agitated, resisting bull in *Hel.* 1557-8 or of the maddened Heracles likened to a bull in *Her.* 868-9, and the playful use in Plato *Phaedo* 117b (Socrates) @#ὥσπερ εἰώθει ταυρηδὸν ὑποβλέψας@#.

@ indicates change of input; # indicates change of font

This represents a distinct change of habit from traditional GreekKeys and requires some getting used to. If you are fairly sure your document will not need to have its Greek converted separately to a different font, then you may of course limit yourself to only one command at each change, for the input. Unicode Greek fonts will normally include Roman characters as well, so you simply select a font, such as Apple's Times (10.4 version) or KadmosU or Cardo, and alternate between US and Greek inputs as needed.

Where do fonts and inputs get installed? In Mac OS X, the relevant folders are in two locations, the top-level Library folder and the Library folder in your home directory. What you install in the former is available to all users of the computer, while what you install in the latter is available to

the particular user and not to other users. Please be aware that there is a third Library folder in OS X, inside the folder named System, and you should never change, remove, or add items in this folder.

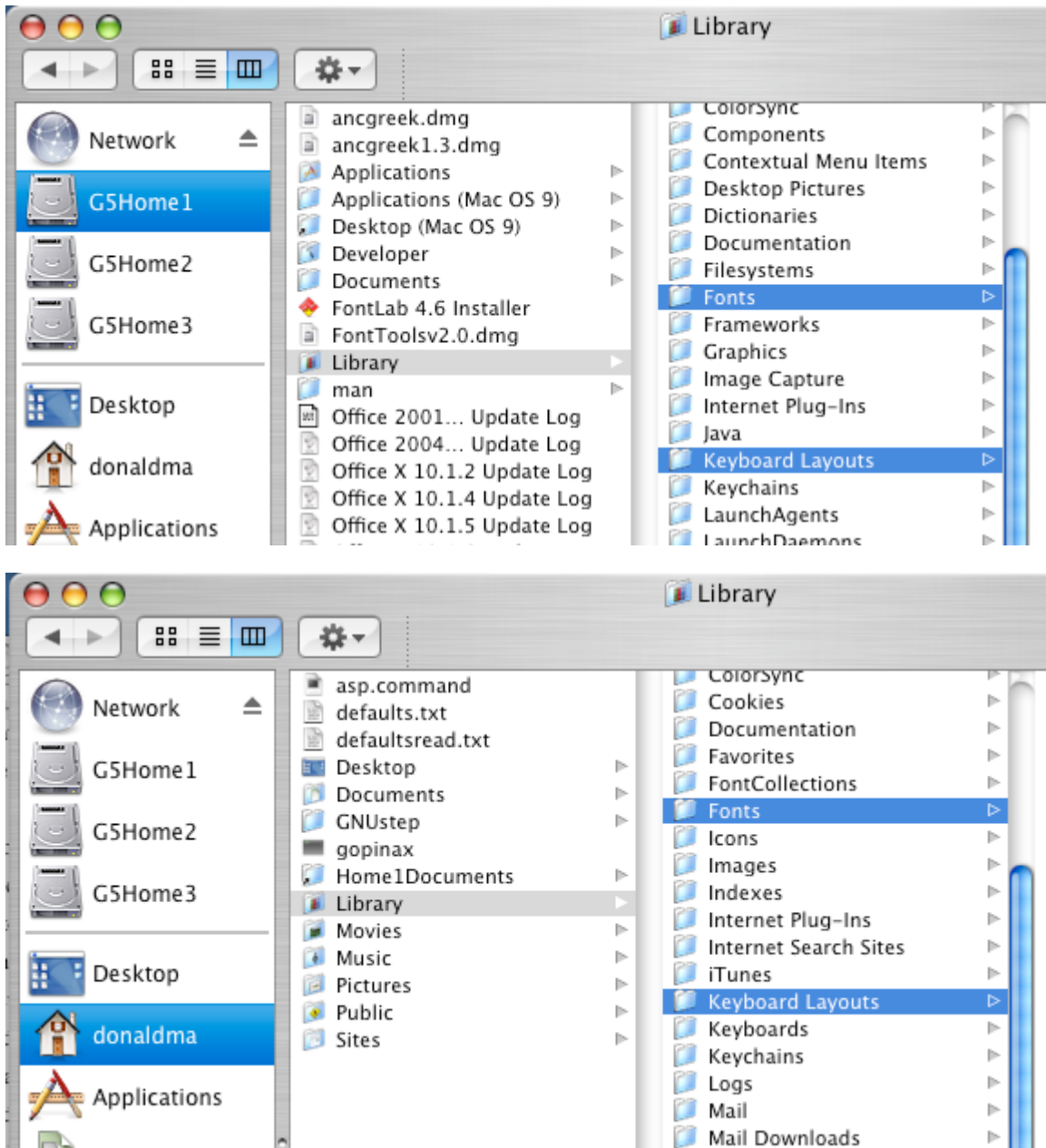


Fig. 10: User-accessible Library Folders in Mac OS X: top-level above; user's level below

Inputs are placed in the folder Keyboard Layouts. Old-fashioned Roman input resources, like GreekKeys Universal, are small files that have the suffix .rsrc. The newer Unicode inputs are XML files with the suffix .keylayout, but are usually enclosed with related files in a special type of folder that has the suffix .bundle. Fonts go in one of the two user-accessible Fonts folders, and in OS X they appear in various forms with various icons and suffixes. In recent versions of OS X, rather than installing fonts manually to the Fonts folders, you should use FontBook.app to install new fonts and to remove duplications by verifying which version you want to be activated. Use the Preview Menu of FontBook to select Show Font Info, then examine the info to see which version of a duplicate font is more recent. If the most recent version has a black dot next to it, it is being ignored: you should select that font and then select the command Resolve Duplicates under the Edit menu.

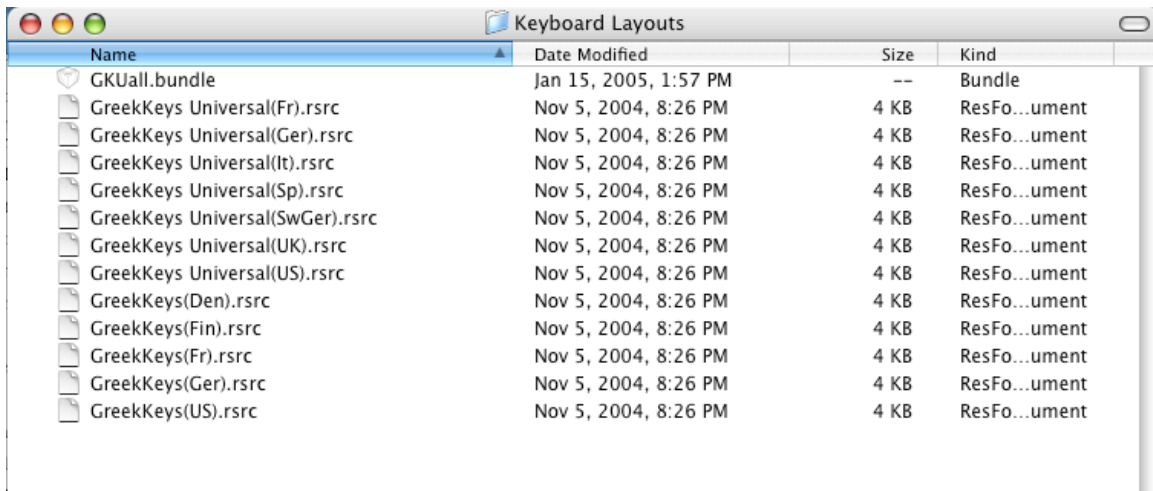


Fig. 11: Added Keyboard Layouts



Fig. 12 : Font types and icons (Mac OS X)

When a new font is installed, it is available to an application the next time that application is launched. When a keyboard layout is installed, the user must either log out and log back in or restart, and then use the System Preference entitled International.



Fig. 13: International System Preference

When International is open, click on the Input menu tab. At the top of the list are useful palettes. I recommend turning on the Character Palette and the Keyboard Viewer.

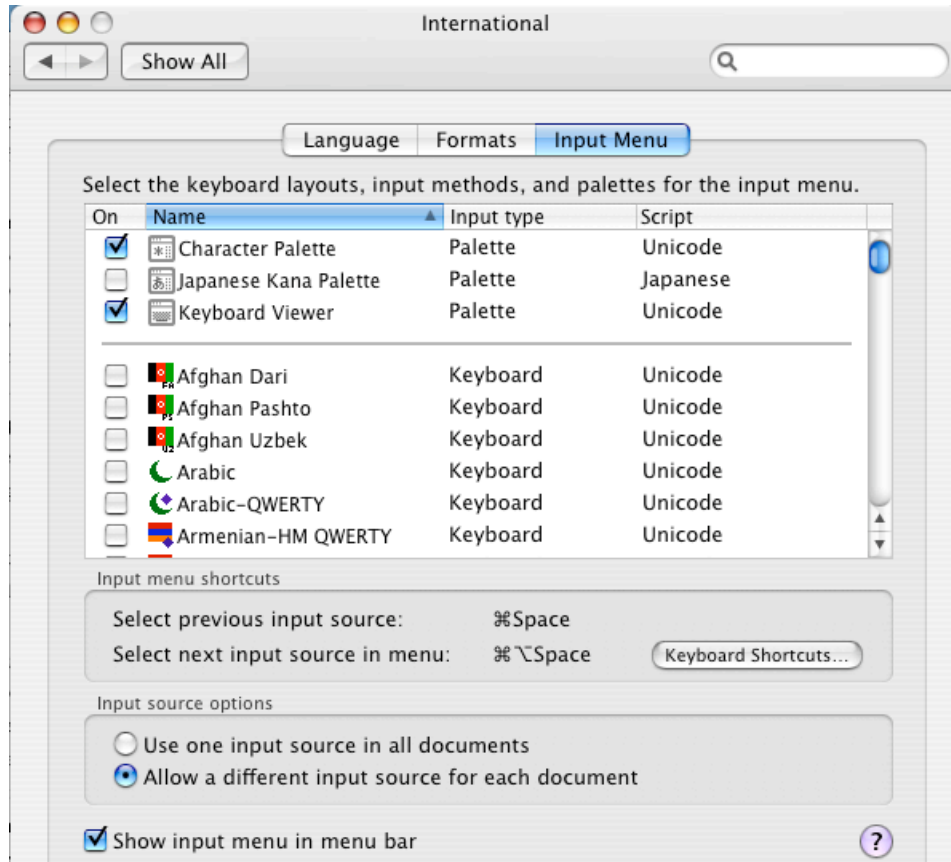
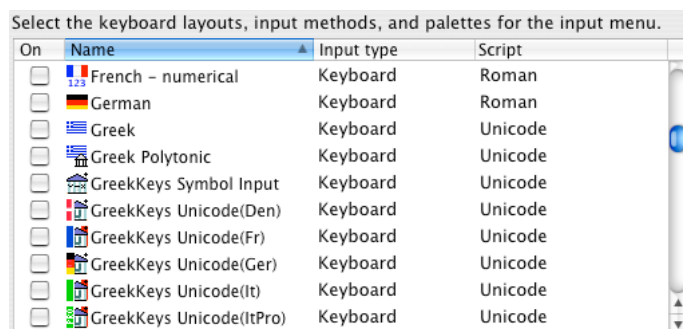


Fig. 14: Input Pane of International System Preference

The rest of the scrolling list contains dozens of localized inputs.



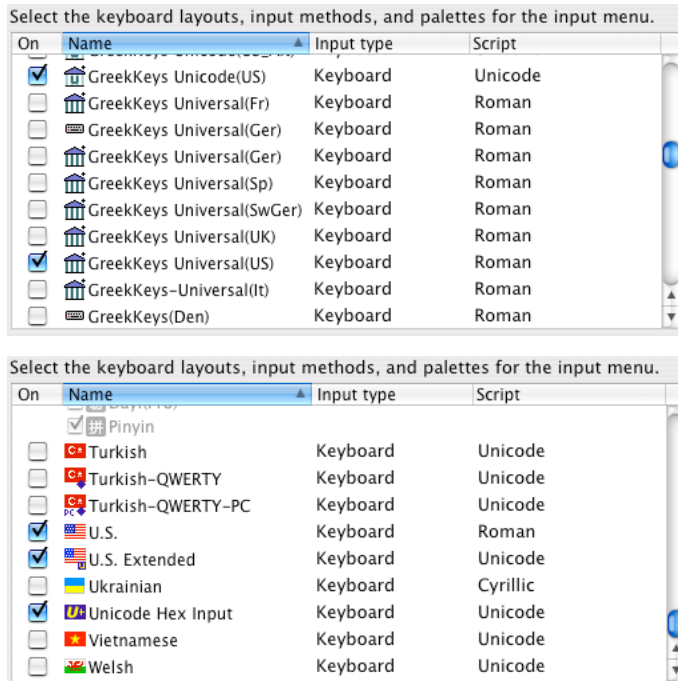
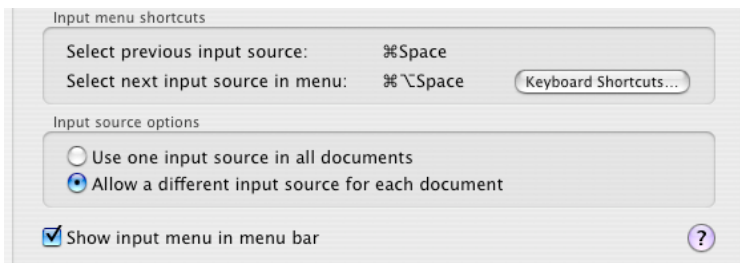


Fig. 15: Three views of list of inputs in Input Pane

Below the scrolling list are important settings. The checkbox at the bottom makes the input menu icon appear in the menu bar.



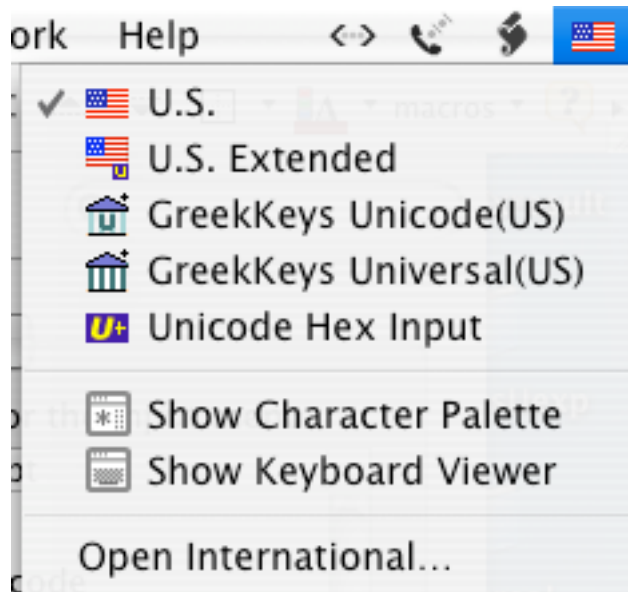


Fig. 16: Settings in Input Pane and the Input Menu

The other settings are for keyboard shortcuts to switch inputs without using the mouse. The default settings up to Tiger are that command-option-spacebar will move one item down in the list of activated inputs, and command-spacebar will toggle between the two types of input (Roman vs. Unicode); more conveniently in Tiger, command-spacebar will toggle between the two most recently used items. In Tiger and later, however, Apple has interfered with this longstanding convention by trying to give the same keyboard shortcuts to Spotlight, and a user may need to reassign the shortcuts (which is easy enough to do: click on the Keyboard Shortcuts... button as seen in Fig. 16, or use the Keyboard & Mouse System Preference).

In Windows XP user-added fonts are installed by opening the Fonts Control Panel and using the Install new font... command in the File menu of the Fonts window. In Vista the same command is available by right-clicking in the Fonts windows.

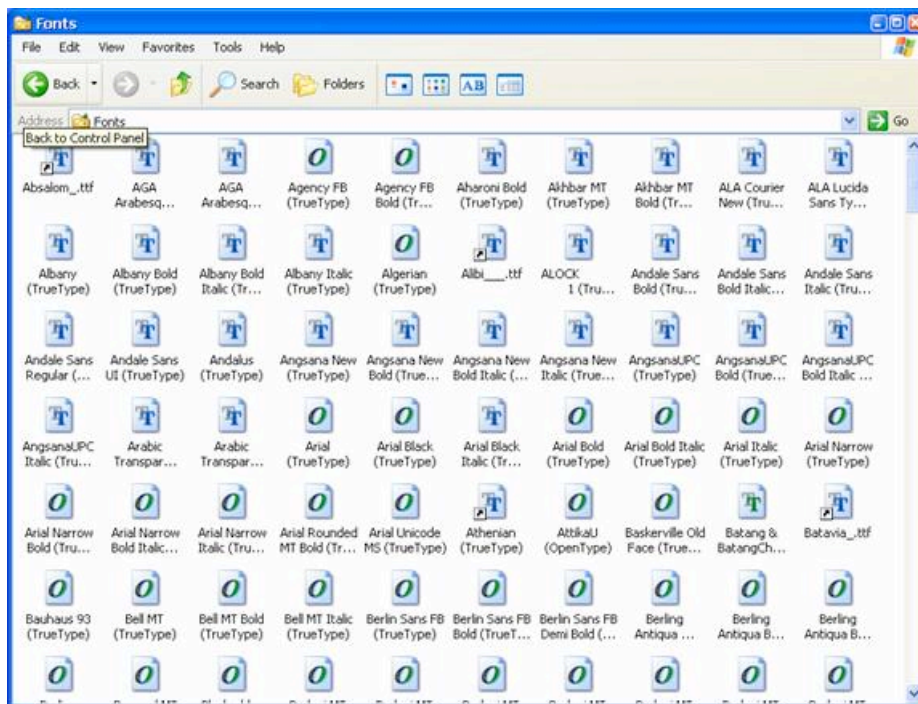
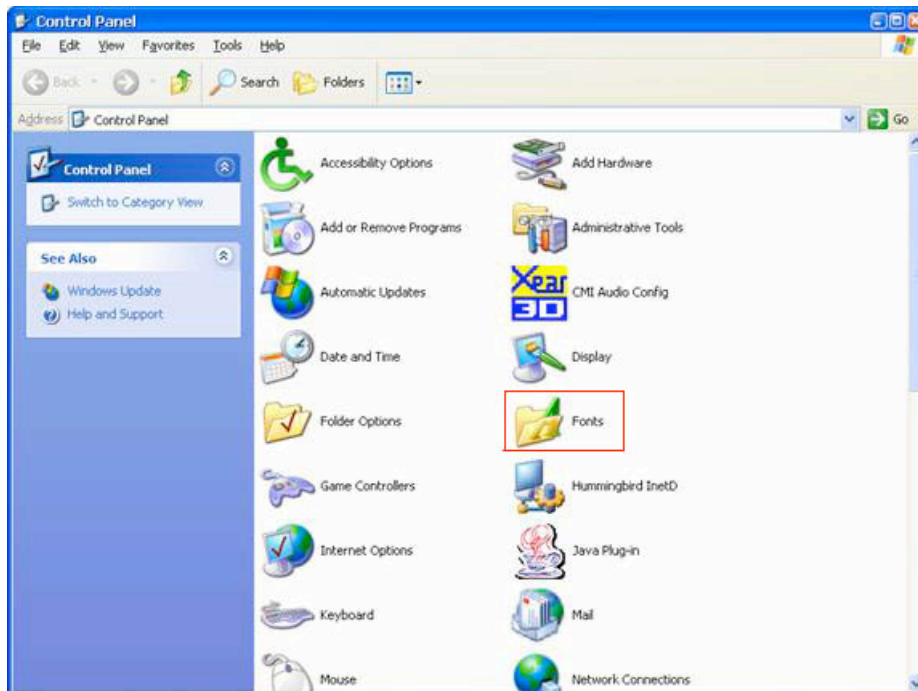


Fig. 17: Windows XP Control Panels: Fonts, and Fonts window

The keyboards available in Windows are activated by a rather complicated sequence of steps. With GreekKeys 2008, however, the installer

itself activates the newly-installed keyboard in Windows XP or Vista (it is best to restart after the installation so that the keyboard choices in the language menu or language bar are accurately updated). Manual activation will be needed for older versions of Windows or if another user on the computer wants to have the new keyboard available as well.²

There are a number of ways to enter Unicode characters in OS X. First, there are specific inputs for many languages, including Apple's own Greek polytonic input (in 10.4 and higher) and of course GreekKeys Unicode. If you want to try out Apple's input, you can activate it and then open the Keyboard Viewer to see how it is set up.

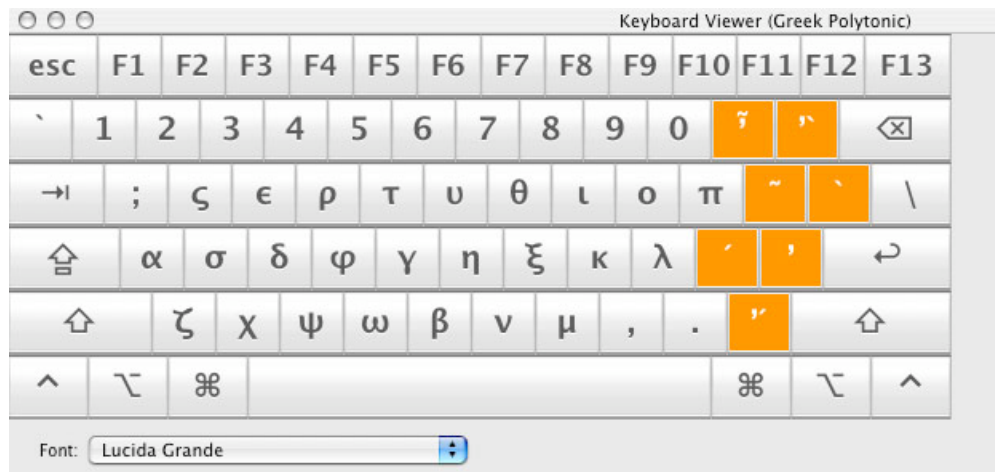


Fig. 18: Keyboard Viewer showing Apple's Greek Polytonic

As you can see, this input places upsilon on the y key and theta on the u key, but the basic letters are otherwise placed just as in GreekKeys. The orange keys in the Viewer indicate the deadkeys. Holding down the shift and/or option key while the Viewer is open will show capitals and option characters and additional deadkeys.

² The steps to take are explained at the GreekKeys support site: <https://webfiles.berkeley.edu/~pinax/greekkeys/installActivate.html#winactive>

When you know the proper code point but do not know whether there is an input that would make entry of this code point easy, the solution is to use the input called Unicode Hex Input.

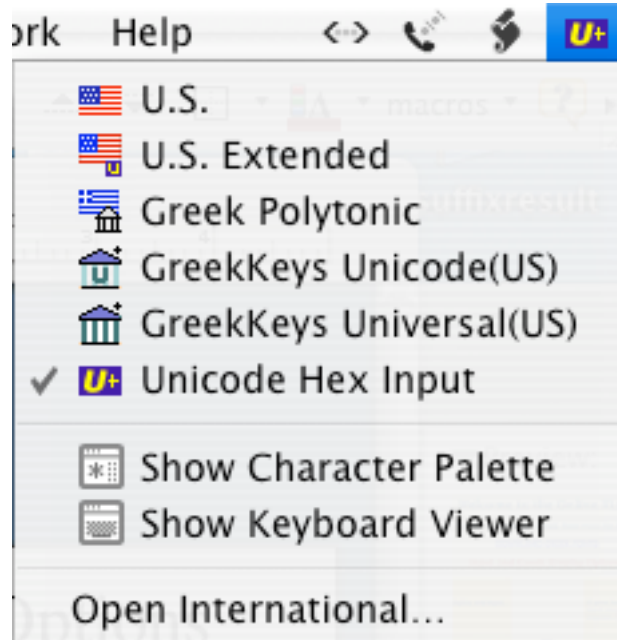


Fig. 19: Selecting Unicode Hex input

As long as the code point is four digits, that is, within the Basic Multilingual Plane, you can activate Unicode Hex, then hold down the option key continuously while you type the four hexadecimal digits. For instance, typing 03e2 will get you Coptic Capital Letter Shei if you have a scholarly font that contains this code point (which is not in any of the OS X system fonts; it is in some of the fonts installed with MS Office 2008). There is a somewhat similar utility on the Windows side, but is internal to MS Word for Windows and does not apply more broadly: in this case, you would type 03e2 into the document and they would appear on the screen, and then before entering anything else after the final 2, press (the left-hand) ALT-x: the 03e2 in your document will be transformed into the capital shei (again, if you have a font containing it). The ALT-x key device for Windows is also

capable of handling five-digit code points in Plane 1. Unicode Hex input does not handle SMP.

Another way to find and insert unusual Unicode characters is to use the OS X Character Palette or the Windows Character Map. The Character Palette can be set to see characters in various sets. The most useful settings are perhaps those for Code Tables and for Glyph. With the former you can see all the blocks of Unicode arranged in order; with the latter you can see the inventory of all the glyphs in a given font.

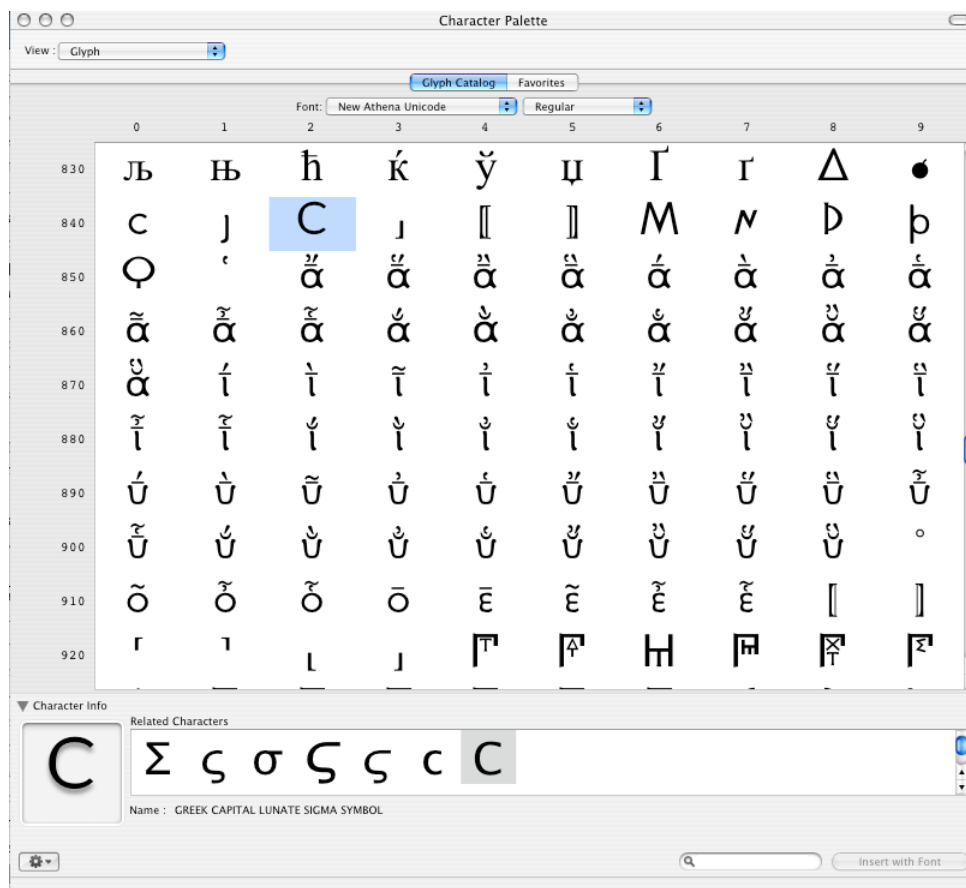


Fig. 20: Glyph view in OS X Character Palette

You can search by code point or by a part of the official Unicode name, if you know it (for instance, if you are looking for a small s with circumflex, you can search for “letter s” or for “circumflex” and locate it. Then you can determine which of your installed fonts include this character. The Character

Palette can be used to insert the SMP code points, which Unicode Hex input does not handle.

Windows Character Map is a small application (located in Programs:Accessories:System Tools) displaying the characters in a specified font in Unicode code point order. First set the font, and then select a character, and click the Copy button so that you can paste it into a document of your choice. If you know the code point, you can enable Advanced View and enter the code point to locate the character in the table. Character Map has limitations: SMP (five-digit) code points are not available for selection under either XP or Vista; under XP this utility shows only the BMP (four-digit) code points that were present in Unicode 4.0, and ignores Unicode 4.1 and 5.0 and pipeline characters that may actually be present in the font; under Vista, however, it shows all BMP code points present in the selected font, including pipeline characters.

Another useful utility in Windows that sometimes outperforms the Character Map is the Insert Symbol palette of MS Word, or the comparable Insert Special Character palette of OpenOffice Writer. With this palette you can specify the font you want to draw from, and the characters will be displayed in Unicode order. There are limitations and inconsistencies in how well these work, depending on whether you are running XP or Vista and on what version of Word is used (and perhaps other factors I have not been able to identify). First, five-digit code points are not shown in Word's palette (and OpenOffice showed some inconsistency between setups), so the ALT-x method may be the only way to do so. Second, this palette may fail to display even some of the 4-digit code points added in recent versions of the Unicode standard, or may display them only after you search for them by code point.



Fig. 21: Insert Symbol Palette of MS Word for Windows

I want to conclude by referring briefly to the problems and prospects for use of Unicode. First, for much of this decade there have been applications that are defective in their treatment of Unicode: Quark XPress before version 7, PowerPoint 2004 for OS X, various browsers, WordPerfect. Second, most applications have been unable to take advantage of smart features in modern fonts (Microsoft Word 2004 for OS X was a big offender in this). As of early 2008, the OSes, major applications, and major browsers (especially if you are running the latest versions, which for reasons

of security everyone should try to do) have improved greatly.³ The major inconvenience now will be when users with the latest software share items with users who have not upgraded.

In the future, fonts are supposed to become smarter: this means that they will contain internal mechanisms for displaying the correct precomposed glyph even when a decomposed sequence of Unicode code points is used. In fact, the roadmap of the Unicode Consortium provides that eventually all data should be maintained in decomposed form, using only base character and combining diacritics. That means that eventually one should not use the Greek Extended block to express polytonic Greek, but only the original Greek block with combining diacritics. At present, however, when you want to use a non-standard combination (such as smooth and circumflex over omicron), although it is certainly possible to express the desired character as a combination of official Unicode code points, the result you see on your screen or printed on paper is likely to be unacceptable, if in fact the application allows you to see anything at all. That is why, for the moment, GreekKeys Unicode fonts, in agreement with some other fonts produced for scholars, use Private Use Area code points for a number of special characters, despite the fact that using PUA interferes with the universality of access to the characters. But smart font features are now also supported by GreekKeys 2008, and cross-platform use of at least partially decomposed input is becoming gradually more practical. Unfortunately, there are two systems for creating smart features, one from Apple (AAT = Apple Advanced Typography) and another from Adobe and Microsoft

³ With Firefox, good support of Unicode Greek and OpenType will arrive with the release of version 3. The version 3 beta release already is vastly superior to Firefox 2.x, which sometime does very odd things to polytonic Greek.

(OpenType), and until recently a font maker who wants to serve a cross-platform audience needed to include both types of features. Apple, however, is now gradually extending OpenType support in OS X, so OpenType features alone will soon work broadly enough.⁴

Precomposed form (generally used now): U+1F87

Decomposed form (eventually to be preferred): U+03B1 (=alpha), U+0314 (=combining rough breathing), U+0342 (=combining circumflex accent), U+0345 (=combining iota subscript)

MS Word 2004 (ignores smart features)			TextEdit (uses smart features)		
Font	precomposed	decomposed	Font	precomposed	decomposed
BosporosU	ᾀ	ᾀ	BosporosU	ᾀ	ᾀ
Lucida Grande	ᾀ	ᾀ	Lucida Grande	ᾀ	ᾀ
NAU	ᾀ	ᾀ	NAU	ᾀ	ᾀ
Times	ᾀ	ᾀ	Times	ᾀ	ᾀ

Fig. 22: Precomposed vs. Decomposed

⁴ The compatibility of various applications with OpenType features is tracked on the GreekKeys support site: <https://webfiles.berkeley.edu/~pinax/greekkeys/compatibility.html>

Some URLs for further information:

GREEKKEYS SUPPORT SITE

main page:

<https://webfiles.berkeley.edu/~pinax/greekkeys/>

Troubleshooting GreekKeys 2005 and 2008:

<https://webfiles.berkeley.edu/~pinax/greekkeys/GKUFAQmac.html>

<https://webfiles.berkeley.edu/~pinax/greekkeys/GKUFAQwin.html>

GREEKKEYS SALES SITE:

<http://www.esellerate.net/apa/apagreekkeys>

NOTE: APA members can get discount on individual licenses by accessing a private sales page through the Members Only link at the APA web site.

UNICODE

main page:

<http://www.unicode.org>

code charts (PDF for each block):

<http://www.unicode.org/charts/>

searching for chart when you have a codepoint:

<http://www.unicode.org/charts/>

searching for chart or codepoint by character name (not a complete list!):

<http://www.unicode.org/charts/charindex.html>

new characters “in the pipeline” for approval and inclusion in the standard:

<http://www.unicode.org/alloc/Pipeline.html>

TLG’s Beta to Unicode Quick Guide:

<http://www.tlg.uci.edu/quickbeta.pdf>

Script Encoding Initiative:

<http://www.linguistics.berkeley.edu/sei/>