

Re: Representation of Greek mute iota
Response to L2/07-068 (Eric Muller)

From: John Hudson, Tiro Typeworks (individual submission)
Date: 1 June 2007

The following comments are edited from an email correspondence with Eric Muller, some of his colleagues in the type group at Adobe Systems, and with a number of Greek typography experts including David-Artur Dax, Yannis Haralambous, Gerry Leonidas, and Nick Nicholas. The discussion was initiated by David Lemon at Adobe Systems, in response to the recent decision by the UTC, in response to L2/07-068, that Option B as documented therein should be adopted as the correct representation of mute iota (*i.e.* a variable encoding dependent on the desired visual result). David Lemon's main concern was to understand the implications of this decision for Adobe's polytonic Greek fonts, particularly in light of existing practice in particular user communities. I believe that Option B in L2/07-068 is an unviable solution that overlooks important aspects of Greek text processing by conflating the encoding of mute and voiced iota. Therefore, I am asking that this question be re-opened and that the action item (B.17) resulting from the decision taken at the UTC meeting in February be postponed until the matter has been reviewed.

Edited excerpt from the email correspondence:

As Eric reports, following his initial message to the Unicore discussion list

I did not hear back a consensus, so I presented the question to the UTC.... The discussion concluded that option B was the best approach, and I was instructed to document that decision in the Unicode Greek FAQ....

I think there is an important step that is missing from this process: an attempt to build consensus that takes into account existing practice and involves a wider range of participants than that to whom the original question was directed. Instead, we are having that discussion now, when the UTC decision is being presented to us as a *fait accompli*. I do believe that there is a good chance of consensus being reached if the question had been more fully examined by interested parties. But even if consensus could not be reached I think it was premature to take the issue to the UTC for an executive decision without at least making the

effort. The document that Eric submitted to the UTC is notable in not making any attempt to record existing practice or to consider in any way whether the encoding of mute and voiced iota might need to be always distinguished. I am concerned that the UTC have made a decision on this issue without full understanding of the issues involved.

Gerry Leonidas writes

To allow an encoding scheme to confuse mute and voiced iotas is short-sighted and misguided beyond description. If the fonts are to be usable by their intended audiences this distinction must be maintained unambiguously.

This is echo'd by Yannis Haralambous' further assertion that the distinction between mute and voiced iota must always be visually possible, *i.e.* that regardless of the encoding there should be a layout mechanism to display the text in a way that represents the distinction between mute and voiced iota. In effect, this raises the possibility of further, more subtly different representations of mute iota in the 'continuum' of possibilities presented in L2/07-068, the correct encoding of which may be uncertain according to Option B.

This distinction must be maintained unambiguously. That is to say, there must never be an instance in which the mute iota is encoded as the same character as the voice iota. This disqualifies L2/07-068's encoding Option B, which obliges users who want an adscript mute iota to conflate the encoding with the voiced iota character U+039B ι or U+0399 I. For this reason, I believe the UTC decision in February was incorrect and prematurely made, and that the question should be re-opened.

In addition to conflating the encoding of two distinct text entities, the Option B encoding introduces problems for text interchange and case conversion, as discussed below.

So what are possible solutions for encoding mute iota in text in a way that maintains an unambiguous distinction between mute and voiced iota, while enabling a range of preferences in display? The following ideas may not be exhaustive, but are offered as a contribution to further discussion of this question.

One solution is L2/07-068's Option A, *i.e.* to always encode mute iota as U+0345, regardless of desired appearance, and to handle variations in display preference at the glyph level.

Another solution would be to encode in Unicode two new characters (adopting Unicode's existing character naming conventions):

GREEK SMALL LETTER PROSGEGRAMMENI
GREEK CAPITAL LETTER PROSGEGRAMMENI

I favour the first of these options because a) it is already encoded, b) it is already in use, c) it provides a consistent interchangeable text encoding independent of display preferences, d)

it provides clean and reversible case conversion, and e) it limits security issues to an existing and known problem without introducing new ones.*

In almost all text the mute iota is actually encoded using the precomposed polytonic diacritic letters, e.g. U+1FBC Αι and U+1FB3 α, and this encoding should not be expected to change in order to facilitate a particular display convention, especially since the same texts may be exchanged between users in different communities, who may favour different display conventions. This encoding using precomposed diacritic letters is encouraged by the most popular polytonic input methods, and is used in major corpuses of classical, *koine* and later Greek text. Members of different user communities with variant typographic conventions for display of mute iota are used to selecting appropriate fonts and/or layout features. The layout features necessary to handle flexible display of polytonic Greek text are complex—and not limited in their complexity to the issue of the mute iota—but they are being documented and have been successfully implemented by font developers using current font formats *without the need to alter the text encoding in order to achieve particular display results*.

If an uppercase alpha Α followed by mute iota is encoded as <U+0391 U+03B9> Αι instead of U+1FBC Αι, as suggested in Option B, this introduces a case conversion problem, since the distinction between the mute and voiced iota, which is always consistent in lowercase text, is lost. This problem may not be any worse than those already afflicting Greek case conversion, but it seems to me unwise to add further difficulties to what is already widely lamented and poorly implemented.

The second possible solution—encoding new uppercase and lowercase adscript mute iota characters—seems to me less desirable, although at least it is a viable option in maintaining the unambiguous distinction between mute and voiced iota. These characters would provide users who wanted to make a display preference explicit in plain text with a means to do so. However, these additions would add a security issue, since the glyph representation would in many—but not all—fonts be identical to U+039B ι and U+0399 Ι. While the security issues associated with some display level preferences for mute iota are real, present and unavoidable; this additional security issue can be avoided if mute iota is consistently encoded as U+0345 and/or the precomposed diacritic letters.

John Hudson

* I think the security issues regarding representation of U+0345 and the precomposed diacritic letters with mute iota using adscript forms are unavoidable and it is too late to close the stable door on that particular horse: there are already too many fonts that implement mute iota support in this way, and demand from user communities will result in more fonts that work this way.

** Note that the font used here, like some others, employs a fairly subtle typographic distinction between mute and voiced iota following an uppercase letter: Αι Αι. This is a display variant somewhere between illustrated forms 3 and 4 in L2/07-068. Option B seems to me ambiguous as to how such a variant should be encoded, and depends on the criteria one uses to cut up the continuum of display variants.