LIST OF PRECOMPOSED GREEK CHARACTERS & CODEPOINTS PROPOSED FOR INCLUSION IN THE PUA

PROPOSAL FOR COORDINATED USAGE OF THESE GLYPHS IN THE PUA AMONG DIFFERENT UNICODE FONTS WITH THE FULL SET OF THE GREEK EXTENDED BLOCK (Vusillus, ALPHABETUM Unicode, New Athena Unicode, and others which wish to follow this recommendation).

January 2006

submitted by

Juan-José Marcos.
juanjmarcos@yahoo.es
Professor of Classics.
Developer of the ALPHABETUM Unicode font
http://guindo.pntic.mec.es/~jmag0042/alphabet.html
Plasencia. SPAIN.

INTRODUCTION

Firstly let me introduce myself.

My name is Juan-José Marcos (42 years), I teach Latin and ancient Greek in Plasencia (Spain) and I am currently developing a Unicode font named ALPHABETUM (which contains around 4,250 characters) intended for classicists and other scholars interested in ancient languages in general, since it includes: Classical and Medieval Latin, Metrical Symbols, Classical Greek, Old & Middle English, Gothic, Coptic, Devanagari, Hebrew, Phoenician, Old Nordic, Old Italic, Kharosthi and Iberic & Celtiberic among other scripts.

The next version of ALPHABETUM (v. 8.00 which will be released probably in May 2006) will include Glagolitic and Old Church Slavonic.

This font is the result of a personal interest dating back many years in the problems faced by classicists who need special characters to type ancient languages.

In this sense ALPHABETUM tries to be useful in helping scholars get access to characters that they need for their works.

Unfortunately, ALPHABETUM is not free. I am very sorry about this, but I am not endorsed or sponsored by institutions, (here in Spain, in spite of my efforts to spread the use of Unicode - talks at Universities, user manuals in Spanish for almost all keyboard utilities such as Antioch, Multikey, Keyman, Thessalonica etc - , there is not interest in Unicode at all, no universities want to spend *any* money on technology for humanities, to my surprise most of teachers and professors haven't heard about Unicode, nor Antioch, nor Multikey, nor Keyman, nor BabelMap, nor GreekKeys etc, they usually type Greek using the old Graeca font by Linguist's Software running Windows 95/98!!!, or even MS Symbol font putting the different accents and breathings by hand!!!!!), therefore I am developing ALPHABETUM on my own without any support, and in consequence I charge users a small fee (15 €), not even it is enough, since this scarcely covers the expenses of purchasing programs to produce (FontLab and FontCreator) and to test the font (Word, Acrobat, InDesign etc), software very expensive by the way. I hope you will understand my attitude. In any case my apologies.

See ALPHABETUM homepage for more details:
http://guindo.pntic.mec.es/~jmag0042/alphabet.html

THE UNICODE STANDARD BLOCK AND THE PRIVATE USE AREA (PUA)

Let me start with the observation that the ideal to reach is the incorporation into the Unicode standard block of a number of ancient scripts, including Latin, Greek, Old Italic (Etruscan, Oscan, Umbrian etc), Iberic, Celtiberic, Aramaic, Old & Medievall Norse, Old English, Glagolitic etc.

Unicode provides precomposed characters for most characters of interest to classicists, however, a number of useful precomposed combinations are not included.

There are a lot of important missing characters: Greek letters with underdot, capital upsilon with smooth breathing, epsilon with circumflex accent, uppercase omega with circumflex accent, Greek letters with underdot, symbol for denarius, vowels and diphthongs with macron or breve in combination with stress mark etc.

I am fully aware that most of these combinations can be obtained via the combining diacritics, but, as you very well know, in most cases if you use the combining diacriticals, by default the multiple diacritics of polytonic Greek stack on top of each other, whereas an acute or grave appears next to the breathing mark or in combination with macrons and breves. You also will find that the different diacritics usually do not line up properly over or under the base character, so it is safer for now to use the precomposed characters, where available, rather than combining diacritics, to guarantee optimal display.

In the future the combining diacriticals may well become more useful when OpenType fonts become widely supported.

It is true that formal proposals for additional characters to the standard block can be submitted to the Unicode Consortium, however, to be realist, it is unlikely that the Unicode Consortium will in the near future include all these glyphs in the standard. On the other hand, simply there is not enough room for such as numerous additions in the Greek Extended block.

As a result, those of us that would like to make use of base scripts, glyph variants or even precomposed characters need to trade off waiting for a Unicode adoption (which in some cases may never happen) or “cutting some corners” in the short-term.

Do not get me wrong, I would very much prefer to adhere to a standard, however, there are current needs in a number of communities, such as the above, that cannot wait years to be satisfied.

In order to balance “order” and need, I see few possibilities other than the use of Unicode standard ranges where they have been defined and the use of the Private Use Area (PUA) where they have not.

USE OF THE PRIVATE USE AREA (PUA): AN INTERIM SOLUTION

If there are no codepoints already assigned to characters we need in Unicode Standard, the font designers can place them in a block of Unicode called Private Use Area (PUA)

Unicode reserves this block for characters which will never have codepoints assigned to them by the Unicode Standard.

Therefore, this area is available to users for their own needs.

Thus the PUA is a solution, but also origin of incompatibilities since this area is not standardized.

At this point there is a simple question:

Are several font developers that implement the same scripts willing to locate the PUA glyphs at the same code points?

Simple right?, however, I do not know of two Unicode fonts that agree in their PUA.

From the beginning, when I started developing the ALPHABETUM font, the use of the PUA was a serious concern to me. For this reason I coordinated ALPHABETUM private codepoints (additional characters for Greek, Devanagari, Hebrew and Latin scripts, where available) with TITUS Cyberbit and Junicode, two excellent, widely available “ancient fonts”.

This was not possible with Cardo because its PUA has not a planned order (only sequential, I believe) and its PUA codepoints conflict with those in TITUS.

ALPHABETUM also shares some codepoints in the PUA with Vusillus (about 50 characters recently added and supported by Antioch version 2.0: E197 up to E1CF, mainly combinations of epsilon and omicron with macron plus different diacritics).

I think that it is of de utmost importance to get a consensus in the use of PUA as widely as possible. Ideally, usage of the PUA should, at least within the “ancient fonts” community, be coordinated in such a way that when switching from one “ancient font” to another, conflicts in the PUA are minimized.

I understand that the basic multilingual plane PUA is becoming crowded in some instances and using this PUA without conflicts it may be a difficult thing to do.

However, in my opinion a little effort from all of us to try a sort of *de facto* standard in the allocation of "ancient" characters in the PUA is worthwhile. Isn't it?

I think it is not too late yet to do so.

Failure to do so will result in many unnecessary incompatibilities to the easy exchange of information and, on the other hand, users will be constricted to use only one font.

One has only to look at the large number of pre-Unicode Greek fonts with many different and incompatible encoding systems (SGreek, GreekKeys, WinGreek, Graeca, LaserGreek, SPIonic etc) to see why this is not a desirable situation for the PUA in the Unicode fonts.

For this reason, I have coordinated ALPHABETUM private codepoints (where available, as far as possible) with TITUS Cyberbit Basic (U+EA00-U+EB9B / U+ E900-U+E9FF) and with JUNICODE (U+F109 - U+F191) – two excellent, widely available, Unicode fonts –, so it will be easy to convert one into the other by using your word-processor's Search and Replace function.

This means that if you receive a text written in TITUS or JUNICODE (and Vusillus as well), you can change the font to ALPHABETUM and the text would, in essence, be the same.

All of us are fully conscious that the PUA may not be a permanent, but a temporary solution.

However, it is also clear that the Unicode track is long and tortuous and that it may take a long time until new characters are accepted, furthermore, no doubt it should be noted that there is even very little chance of many characters being added.

As David J. Perry said very well: "People need access to characters now, and the more we can help avoid duplication and conflicting character assignments, the better".

Over time one could convert and migrate to "true" Unicode, as Unicode evolves.

In the end, we face a election between a purist Unicode standard that may never happen for rich ancient language scripts, or a somewhat extended Unicode character set that is not pure but eminently usable over the next years.

My preference is clear: let's be pragmatic. We should adhere to standards as much as possible, but not slavishly, at the expense of usability.

As I said, I like very much to be pragmatic and to solve problems with practical solutions. So I hope to find agreement among classical and ancient fontmakers for a de facto solution (a sort of an unofficial de facto standard).

If so, we would have achieved a lot for this small community and we would be better off than with the existing and individual solutions.

THE WEAK POINT OF THE PRIVATE USE AREA.

As I said, the Private Use Area is a solution for our needs, but it is not exempt from problems.

I agree with David J. Perry when he says:

"It's very important that everyone be clear about the limitations of the PUA, particularly searching. If, for instance, one set up a database of medieval texts and used ligatures in the PUA, someone searching the database might not find what he wanted unless he knew the exact codepoints for the ligatures and realized that he had to use them in his searches. I'm not saying that we shouldn't use the PUA, but we need to be very clear about its limitations and act accordingly".

David J. Perry is right, however, unfortunately this is not the only problem with PUA.

The PUA is a special block of Unicode and therefore is treated in a different way than standard blocks by software applications.

It is not always easy to find out how software vendors have made use of the Unicode private use area.

Handling of PUA Characters in Microsoft Software

For people with PUA character needs, a major concern is whether the software products that they use will make particular assumptions about the semantics of PUA characters that affect how their text is processed. If a software product assumes different semantics for a code point than the user is assuming, the user may experience unexpected text-processing behaviours, such as lines breaking at undesirable points within the text. At worst, though, irreparable data loss might also occur.

Between them, major software vendors have made use of the entire PUA area in the Basic Multilingual Plane (BMP) of Unicode (U+E000 up to U+F8FF) for one purpose or another. That does not mean that none of these code points are usable in popular software products, though: the fact that vendors have made use of these code points does not necessarily imply that software products made by these vendors will assume certain semantics. It does raise concern for users of those software products, however, and point to the need for testing.

Briefly, each application made assumptions about the semantics of PUA code points, though this was not always insurmountable. Users should be able to work with PUA characters in these applications, though there are some exceptions and special considerations they need to be aware of:

- If you try to display PUA characters using a font that doesn't support those characters, many of the PUA characters used by Microsoft are likely to be displayed in Microsoft software using glyphs from other fonts.

- When PUA characters are entered into Word 2003, it may at first change the font, and you may need to reformat the text with the font you use for your PUA characters.

- Access 2003, Excel 2003, FrontPage 2003 and Publisher 2003 all display PUA characters without problems using an appropriate font. Excel 2003, FrontPage 2003 and Publisher 2003 were very well-behaved with regard to PUA characters — better than Word 2003, in fact.

- PowerPoint 2003 enforces its own semantics for PUA characters, and so is useless for users that have defined their own PUA characters.

MY PROPOSAL ON COORDINATED USAGE OF THE PUA

As first step I have had a look at the PUA of Vusillus, New Athena Unicode, ALPHABETUM Unicode and TITUS Cyberbit Basic fonts.

TITUS and ALPHABETUM have codepoints assigned to several characters proposed by Donald Mastronarde in the range U+EBxx. Vusillus and New Athena Unicode does not use this range, hence there are not conflicts.

Therefore no changes needed in the codepoints assigned to these characters already present in TITUS and ALPHABETUM.

Only must be added codepoints to new proposed characters.

So we would avoid unnecessary duplication of glyphs.

In my opinion the coordination with TITUS is very important and adequate for several reasons:

1.- TITUS is a widely available “ancient font” and fairly well-known to the subset of people interested in multilingual computing and ancient scripts.

2.- TITUS is a large and substantial font (around 9,800 characters) which covers many scripts.

3.- TITUS is the only Unicode font I know which has a well-planned PUA: codepoints has not been arbitrary assigned to characters in TITUS PUA.

In the following pages you can find the concrete proposal for coordinated usage of the private use area.

I took great care over the task of assigning codepoints to proposed characters in order to avoid conflicts with those already existing in TITUS or ALPHABETUM, for the moment the only fonts I know which contains some of these signs.

Synopsis of the PUA in the ALPHABETUM Unicode font.

ALPHABETUM Unicode uses extensively the PUA to support ancient languages no encoded by the Unicode yet and is also used to include precomposed glyphs no present in the standard block.

1.- PUA U+E000 up to U+EFFF section

The section U+E000 up to U+EFFF is coordinated with TITUS. It includes additional miscelaneous signs for varios scripts: Latin, Cyrillic, Devanagari, Greeek etc. This is the section I have choosen for inclusion of the precomposed Greek characters suggested by Donald Mastronarde.

2.- PUA U+F000 up to U+F9FF section

The section in the range U+F000 up to U+F0F0 is used for **combining diacritics** in coordination with TITUS.

The section U+F100 up to U+F2F0 is **unused** in TITUS but it is used by Junicode for "**saxon typeface**" (insular style) and several vowels with macrons and breves plus additional signs (medieval punctuation marks, abbreviations etc). ALPHABETUM is coordinated with Junicode.

The section in the range U+F300 up to U+F3FF (256 slots) is a **unused** block in TITUS.
This section is reserved for **Medieval Latin** characters. These characters are coordinated with other fonts which follow the MUFI (*Medieval Unicode Font Initiative*) recommendation.
For more details about MUFI, please visit http://gandalf.aksis.uib.no/mufi/

The section in the range U+F400 up to U+F5FF is used by ALPHABETUM for **Old Italic** script.

The section in the range U+F600 up to U+F6CF is used by ALPHABETUM for **Iberic** script.

The section in the range U+F700 up to U+F760 is used by ALPHABETUM for **metrical symbols**.

The section in the range U+F770 up to U+F7FF is used by ALPHABETUM for miscellaneous characters and scripts (**Old Gothic** script, additional **Ogham** characters, extra **Hebrew**, alchemical signs, **Latin** digraphs with acute and/or macron, V & v with macron, Latin numerals with bar above, I “longa”, denarius and sestertius signs etc.)

The section in the range U+F800 up to U+F84F is used in ALPHABETUM for **Old Persian** script.

The section in the range U+F850 up to U+F8FF (176 slots) will be used in ALPHABETUM for **Old Chuch Slavonic** and missing Cyrillic signs. I am currently working on this.

The section in the range U+F900 up to U+F9FF is reserved by Unicode for **CJK compatibility ideograph**, so it is safer to avoid this range.

List of new precomposed Greek characters.

Precomposed glyphs in E1xx (E197 up to E1CF) already established for Antioch

Most of these characters are combinations of epsilon and omicron plus different diacritics recently supported by Antioch and consequently included in Vusillus.

The codepoints choosen by Ralph Hancock for these glyphs conflict with those already present in TITUS (L and M Latin letters with different diacritics).
Unfortunately maintaining the cooordination with TITUS here is not possible, it is too late to do so, since doubtless many users have been already used these signs since Antioch is the most important imput method for ancient Greek running Windows. For this reason I have used these codepoints in ALPHABETUM instead of those in TITUS. Nevertheless TITUS has a few of these precomposed signs in EB22 and following.

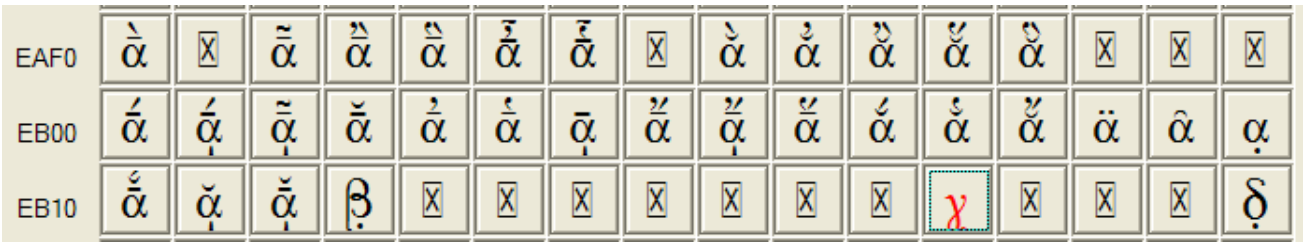
New precomposed glyphs suggested by Donald Mastronarde

These precomposed glyphs are exclusively lowercase alpha, iota and upsilon in combination with different diacritics.

The coordination with TITUS here is not problematic at all, since most of them are already present in this font and there are empty boxes for additions. Hence these codepoints have been choosen in my proposal.
When a sign is not present in TITUS I have assigned empty spaces near or beside these.

I have tried that all characters are as close as possible, mainly to facilitate the use of the Symbol/Insert command.

Please have a look at the following screenshot taken from my computer which shows the final results with letter alfa:



As you can see all characters are grouped and even there are a few empty boxes for future additions.

In the case of iota and upsilon the result is not as good, but, at least in my opinion, also acceptable.

Alpha

Description	Glyph	Codepoint
alpha with macron and acute	ᾱ́	EB00
alpha with macron and grave	ᾱ̀	EAF0
alpha with macron and circ	ᾱ͂	EAF2
alpha with macron and smooth	ᾱ̓	EB04
alpha with macron and rough	ᾱ̔	EB05
alpha with macron and smooth acute	ᾱ̓́	EB07
alpha with macron and smooth grave	ᾱ̓̀	EAF3
alpha with macron and rough acute	ᾱ̔́	EB09
alpha with macron and rough grave	ᾱ̔̀	EAF4
alpha with macron and smooth circ	ᾱ̓͂	EAF5
alpha with macron and rough circ	ᾱ̔͂	EAF6
alpha with breve and acute	ᾰ́	EB0A
alpha with breve and grave	ᾰ̀	EAF8
alpha with breve and smooth	ᾰ̓	EAF9
alpha with breve and rough	ᾰ̔	EB0B
alpha with breve and smooth acute	ᾰ̓́	EB0C
alpha with breve and smooth grave	ᾰ̓̀	EAFA
alpha with breve and rough acute	ᾰ̔́	EAFB
alpha with breve and rough grave	ᾰ̔̀	EAFC

Additional precomposed combinations proposed by Juan-José Marcos:

Description	Glyph	Codepoint
alpha with macron, acute and iota	ᾱ́ͅ	EB01
alpha with macron, circ and iota	ᾱ͂ͅ	EB02
alpha with macron and iota	ᾱͅ	EB06
alpha with macron, smooth acute and iota	ᾱ̓́ͅ	EB08
alpha with macron and breve	ᾱ̆	EB03
alpha with macron, breve and acute	ᾱ̆́	EB10
alpha with breve and iota	ᾰͅ	EB11
alpha with macron, breve and iota	ᾱ̆ͅ	EB12

Iota

Description	Glyph	Codepoint
iota with macron and acute	ᾱ́	EB39
iota with macron and grave	ᾱ̀	EB38
iota with macron and circ	ᾱ̃	EB36
iota with macron and smooth	ᾱ̓	EB3C
iota with macron and rough	ᾱ̔	EB3E
iota with macron and smooth acute	ᾱ̓́	EB3D
iota with macron and smooth grave	ᾱ̓̀	EB54
iota with macron and rough acute	ᾱ̔́	EB3F
iota with macron and rough grave	ᾱ̔̀	EB55
iota with macron and smooth circ	ᾱ̓̃	EB57
iota with macron and rough circ	ᾱ̔̃	EB58
iota with breve and acute	ᾰ́	EB40
iota with breve and grave	ᾰ̀	EB44 *
iota with breve and smooth	ᾰ̓	EB41
iota with breve and rough	ᾰ̔	EB43
iota with breve and smooth acute	ᾰ̓́	EB42
iota with breve and smooth grave	ᾰ̓̀	EB45
iota with breve and rough acute	ᾰ̔́	EB47
iota with breve and rough grave	ᾰ̔̀	EB48

Additional precomposed combinations proposed by Juan-José Marcos:

Description	Glyph	Codepoint
iota with macron and breve	ᾱ̆	EB3A
iota with macron, breve and acute	ᾱ̆́	EB3B
alpha with diaeresis and breve	ᾰ̈	EB46

***EB44.** In the list of Donald the iota with breve and grave is erroneously assigned to EB43.

Upsilon

Description	Glyph	Codepoint
upsilon with macron and acute	ῡ́	EB7A
upsilon with macron and grave	ῡ̀	EB6F
upsilon with macron and circ	ῡ͂	EB70
upsilon with macron and smooth	ῡ̓	EB7D
upsilon with macron and rough	ῡ̔	EB7E
upsilon with macron and smooth acute	ῡ̓́	EB7F
upsilon with macron and smooth grave	ῡ̓̀	EB71 *
upsilon with macron and rough acute	ῡ̔́	EB80
upsilon with macron and rough grave	ῡ̔̀	EB75
upsilon with macron and smooth circ	ῡ̓͂	EB77
upsilon with macron and rough circ	ῡ̔͂	EB78
upsilon with breve and acute	ῠ́	EB81
upsilon with breve and grave	ῠ̀	EB83
upsilon with breve and smooth	ῠ̓	EB84
upsilon with breve and rough	ῠ̔	EB82
upsilon with breve and smooth acute	ῠ̓́	EB85
upsilon with breve and smooth grave	ῠ̓̀	EB88
upsilon with breve and rough acute	ῠ̔́	EB89
upsilon with breve and rough grave	ῠ̔̀	EB8A

Additional precomposed combinations proposed by Juan-José Marcos:

Description	Glyph	Codepoint
upsilon with macron and breve	ῡ̆	EB7B
upsilon with macron, breve and acute	ῡ̆́	EB7C

***EB71.** In the list by Donald the upsilon with macron and smooth grave is erroneously assigned to EB80.

EPILOGUE

This is my proposal for coordinated usage of the PUA.
In my humble opinion it is a reasonable proposal.
Comments will welcome.
I am looking forward to hearing your opinion.

Finally, no doubt it is important that there are many Unicode fonts which incorporate these signs in the PUA, however, equally important is that there is a suitable imput method which allows users to enter them easily. Here the help of Antioch (at least for Windows users) will be imprescindible. If Ralph is willing to support these signs would be wonderful.

SALVETE OMNES!

Juan-José Marcos.
Head of Department of Classical languages.
Plasencia. Spain.
juanjmarcos@yahoo.es

Post scriptum: I have taken care to make this proposal, nevertheless, I am afraid that it is not exempt from faults. Any errors which it contains are exclusively mine, so if you find any wrong code or any other mistake, please let me know.

29 January 2006

LINKS

ALPHABETUM font web page:
http://guindo.pntic.mec.es/~jmag0042/alphaeng.html
Paleographic fonts for Latin script
http://guindo.pntic.mec.es/~jmag0042/palefont.html
NEW: **Fonts for teaching children to write**
http://guindo.pntic.mec.es/~jmag0042/kidfonts.html