NAME

unibetacode – Format for polytonic Beta Code files

SYNOPSIS

source_file.beta

DESCRIPTION

Unibetacode is an implementation of Beta Code, as adopted by the University of California, Irvine Thesaurus Linguae Graecae (TLG) Program and the Tufts University Perseus Project, among others. Beta Code provides a way of encoding polytonic Greek characters using plain ASCII characters. The **unibetacode** package contains three utility programs: **unibetaprep**(1) converts TLG-unique numeric codes to Unicode code points, **beta2uni**(1) converts a Beta Code file to UTF-8 Unicode, and **uni2beta**(1) converts a UTF-8 Unicode file to Beta Code. These programs can also process Coptic and some Hebrew, but historically the focus of Beta Code documents has been classical Greek. The package also contains **libunibetacode**, which provides functions for conversion between Beta Code and UTF-8 Unicode.

A Unicode *code point* is an assignment to a specific numeric value for glyphs and other entities in Unicode fonts. Throughout this document, Unicode code points are given by their Unicode numeric values in the form U+xxxx, where "xxxx" is a string of four hexadecimal digits representing a glyph in the Unicode Basic Multilingual Plane. This is the notational convention in The Unicode Standard and elsewhere.

Note: Thesaurus Linguae Graecae and TLG are registered trademarks of the University of California.

GENERAL PUNCTUATION

PUNCTUATION COMMON TO ALL MODES

Regardless of the language mode (Greek, Latin, Coptic, or Hebrew), several punctuation marks are retained as is between the input file and the output file. They are as follows:

.	Full Stop (Period)
,	Comma
?	Question Mark (except in Greek mode, where it becomes a Combining Dot Below)
!	Exclamation Mark
;	Semicolon / Greek Question Mark (see the Greek section below)
[ ]	Square Brackets

QUOTATION MARK STYLES

The TLG Beta Code specification supports nine different styles of quotation mark. While support of different styles is beneficial, this complicates round-trip conversion from Beta Code to Unicode and back. This is further complicated by the same Unicode character being used as an opening quotation mark in one style and as a closing quotation mark in another style.

Double quotes in the **unibetacode** package just use ASCII quotation marks in a Beta Code source file; the quotation style is determined by the language mode. This greatly simplifies round-trip conversion between Beta Code and a Unicode UTF-8 document. Double quotation marks must be balanced. The first quotation mark encountered will be interpreted as a left quotation mark for the chosen style, the second will be interpreted as a right quotation mark, and so on. Double quotation marks are converted as follows:

Greek, Coptic
: The opening double quotation mark is rendered as U+00AB, LEFT-POINTING DOUBLE ANGLE QUOTATION MARK. The closing double quotation mark is rendered as U+00BB, RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK.

Hebrew
: The opening double quotation mark is rendered as U+201E, DOUBLE LOW-9 QUOTATION MARK. The closing double quotation mark is rendered as U+201D, RIGHT DOUBLE QUOTATION MARK.

Latin
: The opening double quotation mark is rendered as U+201C, LEFT DOUBLE QUOTATION MARK. The closing double quotation mark is rendered as U+201D, RIGHT DOUBLE QUOTATION MARK.

Single quotes are specified explicitly. For Latin and Hebrew, they are specified with a grave accent (U+0060) for an opening single quote and an apostrophe (U+0027) for a closing single quote. For Greek and Coptic, they are specified with a "<" for an opening single quote and a ">" for a closing single quote. The use of "<" and ">" for Greek and Coptic was a compromise so that an ASCII apostrophe in Greek mode would render as U+02BC, MODIFIER LETTER APOSTROPHE. As with double quotation marks, the rendering of these characters is dependent on the language mode. Single quotation marks are converted as follows:

Greek, Coptic
: The opening single quotation mark is rendered as U+2039, SINGLE LEFT-POINTING ANGLE QUOTATION MARK. The closing single quotation mark is rendered as U+203A, SINGLE RIGHT-POINTING ANGLE QUOTATION MARK.

Hebrew
: The opening single quotation mark is rendered as U+201A, SINGLE LOW-9 QUOTATION MARK. The closing single quotation mark is rendered as U+2018, LEFT SINGLE QUOTATION MARK.

Latin
: The opening single quotation mark is rendered as U+2018, LEFT SINGLE QUOTATION MARK. The closing single quotation mark is rendered as U+2019, RIGHT SINGLE QUOTATION MARK.

EXTENSIONS FOR ASCII AND UNICODE

The **unibetacode** package includes two extensions to TLG Beta Code: one for efficiently inserting an ASCII string into text when not in Latin mode, and the other for inserting a Unicode code point in any language mode. These are described in the following two sub-sections.

ASCII STRING INSERTION

An ASCII string can be enclosed in curly brackets when in a non-Latin language mode. The string will be output verbatim. This can be useful if a Greek text uses ASCII symbols, in order to produce a Greek document with an ASCII colon (':') rather than a Unicode "GREEK ANO TELEIA" character, U+0387. The format is as follows:

{*ASCII-string*}

This is an extension to standard Beta Code; the TLG specification assigns a different use to '{'. By itself, the use of '{' not followed by a decimal number is deprecated in the TLG specification, so this should avoid some conflict. Curly brackets are not allowed in the string.

SPECIAL UNICODE CHARACTER INSERTION

The original Beta Code specification lists many numeric codes for producing special symbols that today have become part of The Unicode Standard. In the future, it will likely be most beneficial if any specialized numeric codes for characters use Unicode code points rather than the historical TLG Beta Code numeric assignments.

The **unibetacode** package allows any Unicode code point to be specified in hexadecimal (which is how The Unicode Standard provides them) as a string inside an ASCII '{'...'}' escape sequence. The Unicode hexadecimal code point of one to six digits is preceded by "\u", taking the form "\u*x...x*". For example, strings such as

{\u3d8} or {\u3D8} or {\u03D8}

can be used to insert the Unicode character U+03D8, GREEK LETTER ARCHAIC KOPPA, which does not have an associated Beta Code letter assignment. Such Unicode code point strings can be mixed with other characters in the same string, as long as any character that follows the Unicode code point is not a hexadecimal digit of '0'–'9', 'A'–'F', or 'a'–'f'.

NUMERIC DIGITS

The numeric digits '0' through '9' are simply entered as '0' through '9', respectively, in any language mode. The language modes are Coptic, Greek, Hebrew, and Latin. They are described in the following sections.

GREEK

LETTERS

Capital and small letters can take the same set of accent marks, but the order in which these are specified differs between capital and small.

Small letters are given in Beta Code in this order: (1) letter, (2) breathing marks, (3) accents, and (4) iota subscript. This follows the traditional typed appearance of small polytonic Greek letters, where breathing marks and then accent marks appear on top of the small letter, and iota subscripts appear below small long vowels.

Capital letters are given in Beta Code in this order: (1) asterisk (which denotes a Capital letter), (2) breathing marks, (3) accents, (4) letter, and (5) iota subscript. This follows the traditional typed appearance of capital polytonic Greek letters, where breathing marks and then accent marks appear to the left of the capital letter, and iota subscripts appear to the right of capital long vowels.

The letter mapping is as follows, in Greek alphabetical order. Letters can be capital or small; generally speaking, small is easier to read, so it is the default output from **uni2beta**(1):

*a or a	Capital or Small Alpha, respectively
*b or b	Capital or Small Beta, resp.
*g or g	Capital or Small Gamma, resp.
*d or d	Capital or Small Delta, resp.
*e or e	Capital or Small Epsilon, resp.
*z or z	Capital or Small Zeta, resp.
*h or h	Capital or Small Eta, resp.
*q or q	Capital or Small Theta, resp.
*i or i	Capital or Small Iota, resp.
*k or k	Capital or Small Kappa, resp.
*l or l	Capital or Small Lambda, resp.
*m or m	Capital or Small Mu, resp.
*n or n	Capital or Small Nu, resp.
*c or c	Capital or Small Xi, resp.
*o or o	Capital or Small Omicron, resp.
*p or p	Capital or Small Pi, resp.
*r or r	Capital or Small Rho, resp.
*s or s	Capital or Small Sigma, resp. Note: a small "s" is interpreted as middle (medial) sigma or final sigma depending upon the context. To force one or the other, see the following two entries.
s1	Small Middle (Medial) Sigma
s2 or j	Small Final Sigma
*s3 or s3	Capital or Small Lunate Sigma, resp.
*t or t	Capital or Small Tau, resp.
*u or u	Capital or Small Upsilon, resp.
*f or f	Capital or Small Phi, resp.
*x or x	Capital or Small Chi, resp.
*y or y	Capital or Small Psi, resp.

*w or w	Capital or Small Omega, resp.
*v or v	Capital or Small Digamma, resp.

Example: "*to fws", "the light" (without accent marks). This could also be written as "*TO FWS"; both capital and small letters give the same conversion into UTF-8.

BREATHING MARKS AND ACCENTS

These are the encodings of breathing marks and accents. In Beta Code (as in written Greek), breathing marks appear before accents.

)	Smooth Breathing
(	Rough Breathing
\	Grave accent
/	Acute accent
=	Circumflex
+	Diaresis
&	Macron
´	Breve
?	Combining Dot Below

Example: "*to\ fw=s", "the light", with a grave accent, or *varia,* over the omicron and a circumflex accent, or *perispomeni,* over the omega. This could also be written as "*TO\ FW=S". **N.B.:** Note that the case of the Latin letter does not matter for accent placement; it is only the case of the Greek letter that matters. Greek capital letters are encoded with a preceding asterisk, so in this example, "O\" and "W=" will appear as small UTF-8.

IOTA SUBSCRIPT

The iota subscript is the last character written after a long vowel with which it appears, whether the letter is capital or small. It is denoted by a vertical bar:

|	Iota subscript

GREEK PUNCTUATION

These are the punctuation symbols that the **unibetacode** package supports:

.	Period (*Teleia*)
,	Comma
:	Middle Dot (*Ano Teleia*)
;	Question Mark (*Epotematiko*)
´	Apostrophe (*Apostrophos*)
– (hyphen)	Hyphen (*Pavla*)
_ (underscore)	Em Dash
#	Greek Number Sign

UNICODE GREEK

The Greek Extended range of The Unicode Standard, U+1F00 – U+1FFF, contains 16 small and capital vowels that have identical representation in the Greek and Coptic range, U+0370 – U+03FF. These are vowels with an "oxia" (acute) accent in the Greek Extended range; they have equivalent glyphs with a "tonos" (acute) accent in the Greek and Coptic range. Because of this duplication, the use of these 16 Greek Extended glyphs is deprecated. **uni2beta**(1) will convert those 16 characters to Beta Code, but **beta2uni**(1) will convert the resulting Beta Code into characters in the Greek and Coptic range (U+0370 – U+03FF); it will *not* convert them back into Greek Extended glyphs. The default settings in the **libunibetacode** library *is* to employ these 16 deprecated Unicode code points, but that can be modified; see the

libunibetacode(3) man page for details.

Also in the Greek Extended Unicode range, the TLG Project considers U+1FBF to be the equivalent of a smooth breathing mark, and **uni2beta**(1) will convert it as such.

LATIN (ASCII)

To display ASCII characters, including the Latin letters 'A' through 'Z' and 'a' through 'z', begin with an ampersand ('&') character. Switch back to Greek mode with a dollar sign ('$') character.

ASCII characters can also be surrounded with curly brackets; for example, "{Here is some ASCII!}". This is non-standard though; the TLG specification uses '&' and '$' to enter Latin and then switch back to Greek.

For efficiency, **beta2uni**(1) is conditioned to interpret sequences that look like accented Greek as accented Greek. Curly brackets can also be useful for overriding such interpretations. For example, if a document contained the text

. . . (this is an example)

The "e)" could be interpreted as a small epsilon with a smooth breathing mark above it. To break this behavior, type

. . . (this is an example{}) *or* . . . (this is an example{)}

and the Unicode output from **beta2uni**(1) will appear as intended. This technique will appear familiar to TeX users.

COPTIC

To display Coptic letters, begin with the character sequence "&100". Switch back to Greek mode with a dollar sign ('$') character. As with Greek Beta Code, capital Coptic letters in Beta Code begin with an asterisk ('*') and small Coptic letters do not.

Note that unlike in Greek mode, the Coptic Beta Code letters are case-sensitive. In general, Coptic letters derived from Demotic use lowercase Beta Codes and map to the Greek and Coptic Unicode script in the range U+03E2 – U+03EF; the rest of the Coptic letters use uppercase Beta Codes and map to the separate Coptic Unicode script in the range U+2C80 – U+2C8D.

The encoding is as follows:

*A or A	Capital or Small Alfa, respectively
*B or B	Capital or Small Vida, resp.
*G or G	Capital or Small Gamma, resp.
*D or D	Capital or Small Dalda, resp.
*E or E	Capital or Small Eie, resp.
*V or V	Capital or Small Sou, resp.
*Z or Z	Capital or Small Zata, resp.
*H or H	Capital or Small Hate, resp.
*Q or Q	Capital or Small Tethe, resp.
*I or I	Capital or Small Iauda, resp.
*K or K	Capital or Small Kapa, resp.
*L or L	Capital or Small Laula, resp.
*M or M	Capital or Small Mi, resp.
*N or N	Capital or Small Ni, resp.
*C or C	Capital or Small Ksi, resp.
*O or O	Capital or Small O, resp.

*P or P	Capital or Small Pi, resp.
*R or R	Capital or Small Ro, resp.
*S or S	Capital or Small Sima, resp.
*T or T	Capital or Small Tau, resp.
*U or U	Capital or Small Ua, resp.
*F or F	Capital or Small Fi, resp.
*X or X	Capital or Small Khi, resp.
*Y or Y	Capital or Small Psi, resp.
*W or W	Capital or Small Oou, resp.
*s or s	Capital or Small Shei, resp.
*f or f	Capital or Small Fei, resp.
*k or k	Capital or Small Khei, resp.
*h or h	Capital or Small Hori, resp.
*j or j	Capital or Small Gangia, resp.
*g or g	Capital or Small Shima, resp.
*t or t	Capital or Small Dei, resp.
\	Jinma (Grave) Accent

Switch back to Greek mode by ending with a dollar sign ('$') character.

HEBREW

The TLG specification only covers the basic Hebrew letters aleph through tav. These letters map to the Hebrew Unicode script in the range U+05D0 – U+05EA. Beta Codes are not defined in the specification for cantillation marks, Yiddish digraphs, etc.

To display Hebrew letters, begin with the character sequence "&300".

Note that unlike in Greek mode, the Hebrew Beta Codes are case-sensitive and they never *begin with an asterisk ('*').*

The encoding is as follows:

A	Alef
b	Bet
g	Gimel
d	Dalet
h	He
v	Vav
z	Zayin
H	Het
Q	Tet
y	Yod
k1	Middle Kaf
k2	Final Kaf
l	Lamed

m1	Middle Mem
m2	Final Mem
n1	Middle Nun
n2	Final Nun
S	Samekh
a	Ayin
p1	Middle Pe
p2	Final Pe
T1	Middle Tsadi
T2	Final Tsadi
q	Qof
r	Resh
s	Shin
t	Tav

Switch back to Greek mode with a dollar sign ('$') character.

SAMPLES

The directory "`examples`" in the source distribution contains samples with mappings from Beta Code to UTF-8 and vice versa. The "`genesis-1-1.beta`" and "`genesis-1-1.utf8`" files show the verse Genesis 1:1 in Koine Greek (from the Septuagint), Hebrew, and Bohairic Coptic in Beta Code and UTF-8, respectively.

SEE ALSO

unibetaprep(1), **beta2uni**(1), **uni2beta**(1), **libunibetacode**(3)

AUTHOR

The **unibetacode** package was created by Paul Hardy.

LICENSE

The **unibetacode** package is Copyright © 2018, 2019, 2020 Paul Hardy.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

BUGS

The format is very straightforward and no known bugs exist. However, Beta Code has been evolving for almost 50 years, especially since the advent of Unicode. As a result, many Beta Code-encoded documents exist in versions of the standard much older than the current version. This version also does not implement many numbered codes that are contained in the TLG Beta Code specification. There are no plans to support the TLG Beta Code formatting codes, as that is beyond the scope of Unicode.