Using the Unicode Standard for Linguistic Data: Preliminary Guidelines*

Deborah Anderson
UC Berkeley

1 Introduction

A core concern for E-MELD is the need for a common standard for the digitalization of linguistic data, for the different archiving practices and representation of languages will inhibit accessing data, searching, and scientific investigation.[1] While the context is expressed in terms of documenting endangered languages, E-MELD will serve generally as a "showroom of best practices" for linguistic data from *all* languages.

The problem posed by the lack of a single, common standard for text data in various scripts was already evident in the business world in the 1980s, when competing industry and governmental body character encoding standards made the exchange of text data chaotic. This situation led to the creation of a single, universal character encoding standard, Unicode.[2] That standard was synchronized with the ISO/IEC 10646, the parallel International Standard maintained by the International Organization for Standardization (ISO). Unicode is now the default character encoding standard for XML.

Fortunately, Unicode provides a single standard that can be used by linguists in their work, either when working with transcription, languages with existent writing systems, or those languages that have no established orthography. Indeed, its use is advocated in "Seven Dimensions of Portability for Language Documentation and Description" (Bird and Simons 2002). But specifying "Unicode" as the character encoding does not provide enough guidance for many linguists, who may be puzzled by the structure of the Unicode Standard and how to use it, are unsure which Unicode characters ought to be used in a particular context and whom to ask about this, are uncertain how to report a missing character, etc. A number of very helpful papers and guides have been written (Constable 2000; Wells) and are accessible on the Web, but a general set of guidelines which can draw together all these resources, as well as answer other questions, is necessary.

This paper aims to lay the foundation for such a guide, upon which additional information and improvements can be added, so that a "best practices" set of recommendations for linguists using Unicode will result.

2 Background: What is Unicode?

Unicode is an international character encoding standard.[3] It stands at the bottom level of a multi-layered model of text data and is concerned with the digital representation of characters. Above this is markup (e.g., HTML, XML, or TEI), which can convey the hierarchical structure of a document and the content that comprises it, and at the top level is metadata, which is structured data about data structure (and can document information such as the content, quality, condition, etc., of a file, or the conventions used).

In Unicode, each character receives a unique number ("character code") that remains the same, regardless of computer platform, language, or program. It is this number that the computer stores and uses to refer to the character. The Unicode character code for the Latin capital letter "A" is U+0041. The format "U+[hex

---

* This paper should be viewed as a work in progress. It has drawn heavily from papers and comments by Peter Constable, Ken Whistler, and Rick McGowan.

[1] E-MELD: Electronic Metastructure for Endangered Languages Data, 1. Introduction, http://linguist.emich.edu/%7Eworkshop/E-MELD.html

[2] Unless otherwise indicated, "Unicode" in this paper refers to "the Unicode Standard," as opposed to the "Unicode Consortium," the organization which oversees and maintains the Unicode Standard and its decision making body, the Unicode Technical Committee.

[3] For more detailed information, consult the latest edition of the Unicode Standard, which is available online from the Unicode website (www.unicode.org) and in print. The website also includes Technical Reports and Annexes, which add important additional information, particularly on normalization (UAX #15). A more in-depth discussion on Unicode for linguistic data is Constable 2000.

number]" is a Unicode convention used to distinguish a reference to a Unicode character from some other hexadecimal reference.

Unicode is used for plain text representation. Plain text is a sequence of character codes; the plain Unicode encoded text for the word "Unicode" is: 0055 006E 0069 0063 006F 0064 0065. Plain text may be contrasted with "fancy" text or "rich" text, which is plain text with additional information (including formatting information, such as font size, styles [bold], etc.). The advantage to using plain text is that it is standardized and universally readable, whereas fancy text may be proprietary or specific to a particular implementation. Hence, for example, when linguists are using diacritic superscripts as part of a transcription or other technical orthography and wish to maintain the superscript status of the diacritic as part of the basic text information, they should use the already encoded characters for such superscript modifier letters in Unicode, rather than using markup or a rich text mechanism (i.e., by applying a superscript style to a base character), if the document will be accessed and searched by automatic processes. Plain text will enable regular expressions or other search specification mechanisms to be used more easily. Also, for many purposes of daily communication, plain text is the most simple and efficient means. A plain text document contains enough information for the text to be legible for most purposes.

Unicode is the standard upon which many current fonts, keyboards, and software are based. It is widely supported by the computer industry and by governmental bodies. The process of incorporating new characters can be lengthy; it can take over two years for a character to be approved by the two standards bodies: the Unicode Technical Committee (UTC) of the Unicode Consortium, and Working Group 2 (WG2) of the ISO/IEC Joint Technical Committee 1 Subcommittee 2 (JTC1/SC2). Even after characters have officially been approved, there is a lag-time before they appear in fonts and in subsequent printed editions of the standard. Still, the latest software and fonts generally have more—and better—support for the Unicode characters.

There are instances where it is possible to use markup *or* character encoding. For a discussion and guidelines, see Unicode Technical Report #20, "Unicode in XML and other Markup Languages" at: http://www.unicode.org/reports/tr20/

3.0 Core concepts in Unicode

There are a number of central concepts in Unicode that are important to grasp, for these underlying ideas can help explain why certain characters are included, and others are not.

It may appear that Unicode is inconsistent in terms of its "core concepts," described below. Because Unicode has absorbed older legacy character encodings, it took on characters which by the Unicode Technical Committee's current policies would no longer be considered for inclusion (such as presentation forms or precomposed letters). But had the UTC opted not to include these characters, major interoperability issues would have resulted for Unicode and all the preexisting data in those character encodings.

The discussion below draws on examples from IPA, but the comments are applicable to other writing systems (and can be used by linguists contemplating the creation of an orthography for an as-yet unwritten language).

3.1 Characters, not glyphs

A critical concept in understanding Unicode is that it encodes characters, not glyphs. A character is "the smallest component of written language that has semantic value" (*The Unicode Standard* 3.0, p. 13), whereas glyphs are what appear on the printed page or on your monitor; they are the images we see, the surface representations of abstract characters. The pictures in the Unicode code charts are only representative, and should not be taken as definitive. For example, the capital letter eng in the Unicode codecharts is represented by Ŋ, though the glyph for this character can also appear with the shape of Ɲ. A font can provide either of these shapes. Since the glyphs in the chart are not definitive, the annotation or character name can help determine which Unicode character is the correct one.

It is important to note that Unicode characters are not necessarily the same as graphemes, the fundamental units in an orthography or writing system. The Spanish grapheme *ch*, for example, which is used in sorting and often considered a separate letter, is covered in Unicode by two characters, *c* + *h*, not a single *ch* character.

Also, there is not always a one-to-one relationship between characters and glyphs: a single Arabic letter, for example, can have different glyph shapes depending upon the letter's position in a word (or as an isolate). In Devanagari, several characters can be merged into a single glyph: the glyph for *kṣa* made up of three characters: *ka* + virama + *ṣa*.

3.2 No new precomposed forms or digraphs

The Unicode Technical Committee (UTC) will no longer accept precomposed forms or digraphs unless there is a very convincing argument to the contrary (see FAQ on "Ligatures, Digraphs, and Presentation Forms" on the Unicode Consortium website, www.unicode.org). A precomposed form is a character that can be broken down ("decomposed") into a series of characters: *ǧ* is precomposed form, made up of a *g* and a combining haček. In the early days of Unicode, being able to dynamically generate forms with a base character and combining mark was difficult or impossible, and many such precomposed forms were included in older characters sets. As a result, a number of these precomposed forms were added to Unicode. However, current rendering engines and fonts are able to create the base character and combining mark combinations dynamically now and the position of the UTC is to rely on this productive method of composition, and to not encode more precomposed forms. Adding new precomposed or digraph characters will cause significant overhead: the need for upgrades in software (for decomposition), inconsistent handling of the character, etc.

As a further example, many indigenous languages of the Americas might seem to require precomposed characters, such as the Navajo high-toned nasalized vowel ą́. But the single precomposed character ą́ is not in Unicode, for it can already be handled by three Unicode characters: a (0061) + combining ogonek (0328) + combining acute (0301). One could similarly create other combinations.

Similarly, ligatures, which are two (or more) glyphs fused together, are also not eligible for character encoding. In general, ligatures can be handled by a font or rendering engine. Six digraph ligatures are included in the IPA block (02A3-02A8). These have been included because they are defined in the IPA for the transcription of the coronal affricates and can be chosen by a transcriber in order to convey a semantic distinction made by the transcriber about the phonetic status of the affricate.

3.3 No variants

Unicode includes characters, not variants. Variant selection can be handled by a font or markup. A few apparent typographic variants appear to be included in the IPA block, such as 0278 LATIN SMALL LETTER PHI. This character, however, was demonstrated to have distinct semantics, and hence was included.

Note: Unicode includes a provision for a Variation Selector. However, the assignment for Variation Selectors is under the strict control of the Unicode Technical Committee, and in order to propose a Variation Selector a full proposal like that used for new characters must be submitted to the UTC demonstrating why the variation needs to be identified in plain text. Note that variants can already be handled by a font or markup.

3.4 No idiosyncratic characters

Unicode does not cover, "idiosyncratic, personal, novel, or private-use characters, nor does it encode logos or graphics" (*TUS* 4.0, final draft, p. 2). Hence, if a field linguist devises his own system of phonetic transcription symbols that is not used by anyone else, such symbols are not eligible for encoding. In order for a character to be accepted into Unicode, it must be shown to have appeared in print and have distinct semantics. Preferably it should also be a symbol that is actively used by the linguistic community.

3.5 Unify whenever possible

If an IPA symbol or other needed letter/symbol is identical to an already encoded character, it is often unified with it. In other words, only one character is included. Duplication would cause confusion on the part of the user, who wouldn't be able to tell the difference between the two. Two (or more) characters that seem to be duplicates in Unicode generally reflect semantic distinctions or differences in character properties, so that though they appear the same, they are not.

For example, 0283 LATIN SMALL LETTER ESH, which is used for the voiceless post-alveolar fricative, resembles the integral symbol, 222B. But the integral symbol is a math symbol (with the character property "Symbol, math," abbreviated to "Sm") and hence there may be problems in using this character in linguistic texts, such as improper line-breaking. The integral symbol also has specific layout properties and cooccurrence constraints (i.e., it takes mathematical expressions on the top and bottom to express limits).

Also, symbols may occasionally be duplicated; this is only because a source standard included two separate characters.

Note: To view a character's properties, see a list of the characters at http://www.unicode.org/Public/UNIDATA/UnicodeData.txt and the information about the properties at http://www.unicode.org/Public/UNIDATA/UCD.html. A more user-friendly way to display the character properties is available in a number of utilities, which can be found on the Useful Resources page of the Unicode Consortium website. Peter Constable's Excel spreadsheet on Unicode character properties is another useful tool: http://www.sil.org/computing/ > "Non Roman Script Initiative" > under Encoding Resources, select "Unicode Character Properties Excel Workbook."

4 Practical Issues: How do I get Unicode to work?

In order for Unicode to work properly, users need:

- A recent operating system (Mac OS 9.2, X, Windows CE, NT, 2000, XP, GNU/Linux with glibc 2.2.2 or newer)
- A recent browser (IE, Safari, OmniWeb, Mozilla/Netscape)
- A Unicode text editor (Word 2000, 2002, Unipad, Apple "TextEdit")
- An input mechanism (a keyboard, the "insert symbol" mechanism, Keyman [configurable keyboard for the PC], copying and pasting symbols from a page such as http://www.phon.ucl.ac.uk/home/wells/phoneticsymbols.htm, LinguistList's Charwrite [an input tool for the Web, http://www.emeld.org/tools/charwrite.cfm])
- A Unicode-enabled font (i.e., Code 2000, Lucida Sans Unicode, SIL's Doulos Unicode font [part of the Encore Font system], Arial Unicode MS, Gentium)

For general information on Unicode-enabled products, with directions on creating webpages with Unicode characters and setting up browsers, etc., see Alan Wood's website: http://www.alanwood.net/unicode/index.html. See also the Enabled Products page on the Unicode Consortium website: http://unicode.org/onlinedat/products.html

Note 1: A useful guide specifically tailored to working with phonetic symbols is provided by J.C. Wells at http://www.phon.ucl.ac.uk/home/wells/ipa-unicode.htm as well on various web pages linked to the IPA website, http://www2.arts.gla.ac.uk/IPA/ipa.html.

Note 2: For a fuller explanation of the entire text-processing model, which covers encoding, input, rendering, analysis, and conversion, see Constable 2000.

5 The Organization of the Unicode Standard

Unicode is organized into blocks of characters, often grouped by scripts or characters with similar features (i.e., math symbols, geometric shapes, etc.). Code charts are viewable in the print version and on the Web.

There are two parts to the code charts:
(a) the code chart proper with representative pictures—or glyphs—of the characters and (b) a names list.

The names list often includes additional information for the users:

• informative note (additional information)

0259 ə LATIN SMALL LETTER SCHWA
· mid-central unrounded vowel

= alternative name

0268 ɨ LATIN SMALL LETTER I WITH STROKE
= barred i, i bar

→ cross-reference (either glyphs are very close or are identical, but the characters are not the same or other linguistic relationships are being indicated)

0259 ə LATIN SMALL LETTER SCHWA
→ 01DD ǝ latin small letter turned e

≡ identical to (used to show *canonical* mapping; the characters that appear after the "identical to" sign can be interchanged unconditionally with the other character listed without any loss of information)

1EA0 Ạ LATIN CAPITAL LETTER A WITH DOT BELOW
≡ 0041 A 0323 ◌̣

≈ almost equal to (used to show *compatibility* mapping, usually to earlier standards; additional formatting information may be contained within angle brackets, such as <font> or <super> for "superscript."

1D2C ᴬ MODIFIER LETTER CAPITAL A

≈<super>0041 A

A very important concept with regard to canonical and compatibility characters is *normalization*, which is discussed in Appendix 3. Because Normalization Form C, which uses precomposed characters as much as possible, is specified for use on the Web, it is recommended that linguistic data should probably be stored in this format. For a helpful chart of the NFC forms, see http://www.unicode.org/charts/normalization/.

6 Finding a character you need:

1. See if it is in Unicode:

- Check on the code charts on the Unicode Consortium website (www.unicode.org/charts), as the website is usually more up-to-date than the print version.

  Most symbols for the IPA and the Americanist systems are already covered by Unicode. Note that some symbols are represented by using a sequence of two Unicode codepoints:

  Example: the voiceless dental plosive, t̪, is covered by:

  0074 t LATIN SMALL LETTER T

  032D ◌̪ COMBINING CIRCUMFLEX ACCENT BELOW

  Though most IPA symbols are contained in the IPA Extensions block, characters also come from other blocks (Latin and Greek, for example). These characters are listed at the beginning of the names list in the IPA Extension block. A new Phonetic Extensions block has been added with Unicode 4.0 (http://www.unicode.org/charts/PDF/U1D00.pdf), created primarily for the Uralic Phonetic Alphabet. Note that characters for linguistic transcriptions may also be created from a base character and characters contained in the Spacing Modifier Letters block or Combining Diacritics block.

Also useful are:
(a) the Unicode Character Names Index, which lists the formal character names, alternative character names, and character group names alphabetically. The page is located at: http://www.unicode.org/charts/charindex.html.
(b) the Collation Charts on the Unicode Consortium website (http://www.unicode.org/charts/collation/), which graphically group similar characters (but separate the differences between them with colors). The differences are determined using the Unicode Collation Algorithm.

- Check Appendix 2 of the *Handbook of the International Phonetic Association* (pp. 161–185), where Unicode character codes for IPA symbols are listed. On the web, a listing is also found at: http://www.phon.ucl.ac.uk/home/wells/ipa-unicode.htm. Another option is to check the IPA chart posted at http://web.uvic.ca/ling/resources/ipa/charts/unicode_ipa-chart.htm, which displays the Unicode character numbers when the mouse hovers over the symbol.

  Note: In looking through Unicode charts and when using "insert Symbol" or font charts, be careful of "spoof buddies" or "confusables":
  Example:
  "Barred o," properly 0275, has a look-alike: 04E9 ө (CYRILLIC SMALL LETTER BARRED 0)
  For a list of such "confusables," see http://www.phon.ucl.ac.uk/home/wells/confusables.htm

2. See if the character is in the process of being proposed:
   - Check on Unicode's Proposed New Characters page: http://unicode.org/alloc/Pipeline.html
   - Ask on the Transcription email list
     To subscribe, send email to the following address: majordomo@listlink.berkeley.edu, with the following command in the body of your email message: subscribe transcription email@address replacing "email@address" with your actual email address. If you ever need to get in contact with the owner of the list, send email to: owner-transcription@listlink.berkeley.edu.
   - Ask on the Unicode email list (directions: http://unicode.org/consortium/distlist.html)
   - Verify the character you need is a true character, and not a variant.

3. If you find a character that is missing
The Unicode Standard offers a huge array of encoded characters that are able to serve most linguists' needs, and because they are already in Unicode—which has been adopted by many software and font companies—they can currently be used in documents. "Inventing" a new character is, however, not recommended, for problems will arise in short and long term accessibility (i.e., sending, receiving, and printing) such non-standard characters.

If a particular character is needed and is not covered by Unicode, then it is advisable to work directly with the Unicode Technical Committee and Peter Constable of SIL (Peter_Constable@sil.org) so a proposal can be put forward. Particularly helpful for proposals are Xeroxes or scans of pages from books or journals that show a particular character in context (with the bibliographic information included). Though the full approval process can take several years, it will provide a means for others in the future to access the character in the international character encoding standard.

General guidelines on how to produce a Unicode proposal are located at: http://www.unicode.org/pending/proposals.html. Proposals include: a list of characters (with their names, a representative glyph for each, and information on each character's properties), a representative sample of the characters in context (i.e., in texts), and a short bibliography with references.

7 Other practical issues

7.1 How can I use a character not yet in Unicode?

(Please read comments above, 6.3.)

- Use FontLab or work with a font foundry to create a font with the needed glyph(s) in the interim, using the Private Use Area (PUA). Fully document the characters that have been placed in the PUA: their names, character properties, and their specific locations in the PUA. For guidelines on which parts of the PUA to use, and which to avoid, check on Peter Constable's articles http://www.sil.org/computing/ > "Non Roman Script Initiative" > under "Unicode"
- Use markup / entities. TEI is preparing guidelines on this topic, but nothing has yet been finalized. The latest drafts are available at: http://www.tei-c.org/Activities/CE/.

  Note: Scalable Vector Graphics is another possibility. See http://www.w3.org/Graphics/SVG/Overview.

7. 2 For those languages without an orthography

Linguists who are dealing with languages that currently have no orthography can make use of the Unicode characters. Indeed, using already encoded characters will assure interoperability with current applications, especially if the glyphs for the characters are represented in fonts that are already widely used (i.e., Arial Unicode MS). However, creating new ad hoc symbols will impede others from having easy access to the data.

Linguists designing a Latin orthography can draw from various code blocks of Unicode, being careful to pay attention to that the properties of a character, which are defined in the Unicode Character Database (UCD). The properties of a character include: Name, General Category, Casing (upper/lower/title, etc.), as well as other properties. The General Category assigns Unicode code points into one major class, and these are designated with a two-letter abbreviation, the first letter of which is a major class (L=Letter, N=Number, C=Other, P=Punctuation, M= Mark, S=Symbol), the second letter being a subclass (Lu=Letter, uppercase, Ll=Letter, lowercase, Lt=Letter, titlecase, Lm=Letter, modifier, Lo=Letter, other, Sm=Symbol, math, So=Symbol, other).

Examples include:
Lu : 0041 A LATIN CAPITAL LETTER A

Ll : 0287 ʇ LATIN SMALL LETTER TURNED T

Lo : 01C3 ǃ LATIN LETTER RETROFLEX CLICK
(= Latin letter exclamation)

Lm : 02B0 ʰ MODIFIER LETTER SMALL H

If the symbol is used for a notation, and is not part of a language orthography, using characters from the Mathematical Operators block is acceptable, as long as they have Unicode So (Symbol, other) or Sm (Symbol, math) properties.

It is best to stay away from presentation forms, letterlike symbols, or number forms. Hence, users should avoid the following:

CJK Compatibility Ideographs F900-FAFF (except the 12 enumerated on p. 267 of *TUS* 3.0)
Arabic Presentation forms FB50-FDFB and FE70-FEFC, instead use the counterparts in the Arabic block, 0600-06FF (however it is acceptable to use the ornate parentheses at FD3E and FD3F);
Alphabetic Presentation Forms (FB00–FB4F);
Combining Half Marks (FE20-FE2F);
CJK Compatibility Forms (FE30-FE4F);
Small Form Variants (FE50-FE6F);
Half-width and Full-width Forms (FF00-FFEF);
characters in the Kang Xi Radicals (2F00-2FDF) and CJK Radicals Supplement (2E80-2EFF) (for ideographic texts);
Hangul Compatibility Jamo range (3130-318F) for Korean texts.

7.3 How do I tell if my font is Unicode compliant?

Not all fonts publicized as "Unicode" are truly Unicode-enabled, in part because the standard is continually evolving, with new characters being added. A useful method to check whether your font is up-to-date or missing glyphs, is to set the font as the default on your browser, then look at a test page, such as: http://www.alanwood.net/unicode/ipa_extensions.html. Alternatively, font utilities are available that can check the fonts installed on your system. A description of these font utilities is available at: http://www.alanwood.net/unicode/utilities.html. A few additional utilities are listed at: http://unicode.org/onlinedat/resources.html.

7.4 Handling data in a non-Unicode encoding or font

If your data will be needed in the future, it is best to convert it to Unicode for long-term preservation. Conversion tools and tutorials are available at: http://www.sil.org/computing/ > "Non Roman Script Initiative" > "Unicode Tutorials."

If you create a font converter that converts a commonly used font to Unicode (particularly one used by linguists), consider making it publicly accessible, such as on the LinguistList or Unicode Consortium websites.

7.5 Encoding Forms

Unicode assigns integers to characters, but there are a number of different ways to represent the hex-based integer (the Unicode scalar value) as a sequence of bytes. These various "encoding forms" differ in their size: the integer can be mapped to a series of 8-bit values (Unicode Transformation Format 8, or UTF-8), a 16-bit value (UTF-16), or a 32-bit value (UTF-32). UTF-16 is a default encoding form for XML. UTF-8 uses one to four 8-bit units, UTF-16 can use one or two 16-bit code units to represent the scalar value, and UTF-32 maps to a single 32-bit code unit.

The multiple encoding forms reflect different implementation needs; some software may support one encoding form and not the other, and there can be some tradeoffs for storage and processing. For data in phonetic transcription, either UTF-8 or UTF-16 is acceptable, and according to Peter Constable UTF-8 will give a slight improvement in storage needs.

Note: XML requires XML processors accept UTF-8 and UTF-16 encodings, though Internet protocols must be able to use UTF-8. HTML does not assume any encoding form as the default, but most Web browsers currently support UTF-8.

8 Further Recommendations

1. Groups of users should publicly document their specific orthographic usages. For example, Athabaskan specialists should create a webpage documenting the Unicode values for Athabaskan orthography and provide font recommendations.

2. Encourage linguists to contact Peter Constable if a character is known to be missing from Unicode. Also, suggestions on helpful annotations that should be included in the Unicode Standard should be directed to Peter Constable or myself (dwanders@socrates.berkeley.edu).

Appendix 1: Linguistic letters and symbols in Unicode

For the IPA: See chart on http://www.phon.ucl.ac.uk/home/wells/ipa-unicode.htm

Additional characters not included on the charts (new with Unicode 4.0) (by block):
Latin Extended-B Block:
0221 LATIN SMALL LETTER D WITH CURL (for Sinology)
0234 LATIN SMALL LETTER L WITH CURL (for Sinology)
0235 LATIN SMALL LETTER N WITH CURL (for Sinology)
0236 LATIN SMALL LETTER T WITH CURL (for Sinology)

IPA block:
02AE LATIN SMALL LETTER TURNED H WITH FISHHOOK (for Sinology)
02AF LATIN SMALL LETTER TURNED H WITH FISHHOOK AND TAIL (for Sinology)

Spacing Modifier Letters block:
02EF – 02FF UPA MODIFIERS

Phonetic Extensions (New Block)
1D00–1D7F Non-IPA phonetic extensions, used mostly for the Uralic Phonetic Alphabet
and includes the 1D4A MODIFIER LETTER SMALL SCHWA

Note 1: The IPA tone letters (rising contour, falling contour, high rising contour, low rising contour, rising-falling contour, amongst others) can all be handled with the existing encoded tone letters, using font ligations. Peter Constable is writing up a discussion of this, which may appear as a Unicode Technical Note. He already has a Graphite font implementation of it.

Note 2: For a listing of all the forms cited in Geoffrey Pullum and William A. Ladusaw's *Phonetic Symbol Guide* (2nd edition, Chicago, 1996) and their Unicode encoding, see Peter Constable's http://www.sil.org/computing/ > "Non Roman Script Initiative" > "Unicode" and select "Symbols in Phonetic Symbol Guide 2nd edn. in relation to Unicode 4.0"

Appendix 2: Characters known to be missing from Unicode

- Certain tone diacritics (contours)
- Hooktop right-tail D (voiced retroflex implosive)
- Raised theta

Note: In June, the Unicode Technical Committee accepted 11 phonetic letters with the middle tilde: b, d, f, m, n, p, r, s, t, z, and r with fishhook and middle tilde. These proceed on to the ISO Working Group 2 in October.

For a full listing of identified—but missing—characters needed for phonetics, see Peter Constable's http://www.sil.org/computing/ > "Non Roman Script Initiative" > "Unicode" > "Unicode-Related Documents" > pdf proposals are listed under "Phonetics." For further information, contact Peter Constable, Non-Roman Script Initiative, SIL, at: Peter_Constable@sil.org.

Appendix 3: Normalization

If one document uses LATIN CAPITAL LETTER A WITH DOT BELOW (1EA0) and another document uses the alternative, but equivalent representation—its *canonical* equivalent—Latin CAPITAL LETTER A and a COMBINING DOT BELOW, a potential problem could arise when doing string comparisons between the documents, since there are two different representations being used. Unicode handles this by defining normalization forms which remove these distinctions. The normalization forms are defined in the Unicode Standard Annex #15 (http://www.unicode.org/reports/tr15/). UAX#15 defines two normalization forms for canonical equivalences: normalization form D (NFD) and normalization form C (NFC). NFD specifies that the text be maximally decomposed, so no remaining character has a canonical decomposition. NFC is conceptually opposite to this notion: text is normalized using precomposed characters as much as possible. Hence for the example Latin capital letter A with dot below, NFC is 1EA0, but NFD is 0041 + 0323. For any string, there is only one normalization form C and one normalization form D defined, and either can serve as a reference point for comparison. NFC is the form that is to be used on the Web, as set out by the W3C.

Note that the order of combining marks above a character can be meaningful. As a result, Unicode assigns combining marks to a particular *combining class*, which is a normative property of the characters. When the order of the combining characters is not important, the artificial distinction in the order is erased as part of the normalization process.

Compatibility characters are not unconditionally interchangeable with their counterparts: the ARABIC LETTER HEH FINAL FORM (FEEA) is not completely identical with 0647 ARABIC LETTER HEH (0647), as there are some cases where the distinction between these two needs to be maintained. Compatibility decompositions are included in the Unicode annotations list, often with non-character information contained in brackets:

FEEA ﻪ ARABIC LETTER FINAL FORM

≈<final>0647

In this case the contextual rendering behavior is provided in the brackets, but font or superscript information (etc.) can also be included in brackets.

When text is normalized in NFC and NFD, the distinction between the compatibility characters and their near synonyms is kept. But there are two other normalization forms defined by Unicode which do not retain the distinctions between compatibility characters and their near synonyms: normalization form KD (NFKD) and normalization form KC (NFKC). NFKD is a precomposed form and NFKC is the decomposed form. Because important distinctions reflected in compatibility characters may be lost with NFKC and NFKD, using these normalizations should be used with caution. If such forms are used, users should perhaps include the compatibility information in markup or in some other way.

Note: compatibility normalization without notice is discouraged, because you can lose important formatting information if text containing these characters goes through an NFC or NFKD normalization.

References

Bird, Steven and Gary Simons, 2002. "Seven Dimensions of Portability for Language Documentation and Description." Proceedings of the Workshop on Portability Issues in Human Language Technologies, Third International Conference on Language Resources and Evaluation, Las Palmas, Canary Islands. Available at: http://arxiv.org/abs/cs/0204020. Revised version: http://www.ldc.upenn.edu/sb/home/papers/0204020/0204020-revised.pdf

Constable, Peter. "Unicode Character Encoding of Archived Linguistic Data". Paper presented at the workshop on Web-Based Language Documentation and Description, 12-15 December 2000, Philadelphia, PA http://www.ldc.upenn.edu/exploration/expl2000/papers/constable2/constable2.html

Unicode Consortium. *The Unicode Standard 3.0.* Reading, MA: Addison Wesley, 1991-2000.

Wells, J.C. "Phonetic Symbols in Word Processing and on the Web" http://www.phon.ucl.ac.uk/home/wells/ICPhS_18.pdf .

———. "The International Phonetic Alphabet in Unicode." http://www.phon.ucl.ac.uk/home/wells/ipa-unicode.htm