L2/02-134

Universal Multiple-Octet Coded
Character Set (UCS) — ISO/IEC 10646

Date: 2002-04-01

Doc type:	Personal contribution
Title:	**Keyboards, fonts and the Unicode Standard**
Source:	Robert A. Simpson

KEYBOARDS, FONTS AND THE UNICODE STANDARD.

Introduction.

The aim of the Unicode Standard, as stated on p. 4*, is that it should be "universal", "efficient", "uniform" and "unambiguous". The opinion of many writers is, however, that in its present incarnation it falls short of achieving the first and second of those very praiseworthy objectives, and it is the purpose of this document to explore some of the reasons why this may be so.

The matters dealt with below have arisen from a preliminary consideration of the Standard as it stands at the time of writing (February 2002). They may both be described as matters of completeness, although the first also involves important matters of consistency. The main point of the first section is that the scope of the Standard is unjustifiably limited by its current exclusion of certain categories of sign (by which I include both "characters" and "glyphs", as differentiated therein), and these exclusions lead to undesirable and unnecessary complications in implementation. The recommendations at the end of this first section are based on both theoretical and practical grounds and are therefore, I believe, worthy of sympathetic consideration.

In the second section I briefly draw attention to some omissions from the Standard which may deserve reconsideration on historical grounds.

Many — indeed, perhaps all — of the points I raise may already have been fully addressed, or they may be the subject of ongoing or planned discussions. All I wish to do is highlight some of the key questions which I feel are likely to occur to a multi-language keyboard designer and font user new to the Standard and hitherto unpersuaded by some of its arguments.

It should be emphasized that the word "text" is used throughout to mean *plain text* only. Fancy text, in the sense of p. 15 of the Standard, is not considered here. For convenience, the word "sign" is used in all parts of this document, except the definitions, to mean any written or printed typographical element (not "text element"), whether that element be considered a "character" or a "glyph".

I. CHARACTERS AND FONTS

1. *The problem of mapping.*

The stated aim of the Standard is to provide "encompass all characters that are likely to be used in general text interchange" (p. 4). This is a laudable and desirable goal, since the lack of any *de facto* coding standard for many languages has resulted in an enormous number of individual font-specific character mappings. English-language writers will scarcely have come across this problem because the basic ASCII/ANSI (Latin) character set is so well-established. In almost every computer font, and all the widely-distributed ones, these characters, including language-specific characters such as Spanish *ñ* or Icelandic *ð*, are given their standard code values. But the situation is less clear-cut for fonts designed for languages other than the most common western European languages. Although Latin-script fonts based on 8-bit encoding generally (though not always) follow the ASCII/ANSI mapping up to 007F, they appear to map non-ANSI characters such as Polish *ż* or Turkish *ı* on a somewhat *ad hoc* basis. Greek and Cyrillic fonts also come with a variety of mappings, while for other non-Latin scripts the situation is frankly chaotic. To take just three Indic characters as examples, we can easily find at least five different

* Unless otherwise specified, all references are to the printed *Unicode Standard Version 3.0* book (edition of 2000), which is referred to throughout as simply "the Standard".

Character	*ASCII/ANSI code point in the selected font (hexadecimal)*				
अ U+0905 DEVANAGARI LETTER A	Devanagari New 41	Kantipur 63	Kautilya 64	Fontasy Himali A1	Webdunia 79
অ U+0985 BENGALI LETTER A	BengaliDhakaSSK 43	ItxBeng 7F	Lipi_Normal 41	Shree-Ban-0552 22	Luit D5
அ U+0B85 TAMIL LETTER A	Amudham 6D	VaigaiAA 40	Tamilfix C7	Vavuniya 61	TAMKalyani DC

TABLE I. *Examples of Indic-font character mapping.*

character mappings for each (see Table I).

Such a plethora of character mappings is extremely inconvenient, since it means that each font requires the creation of its own specific rendering software. Things would be much simpler if every font mapped the same signs to the same code points. It was hoped that Unicode would establish a standard code for a given printed sign, so that it could (and should) be mapped to the same code point in every font. At present, however, this is not the case. Unicode specifically excludes many signs, leaving it to the rendering engine to select from the given font the glyph corresponding to the Unicode value, or combination of values, with which the engine is presented. It does not actually specify a unique code for each and every sign that is eventually printed.

This approach means that, except for western European languages, there will not in general be a one-to-one relationship between the text as displayed or printed, *i.e.* the rendered text, and the code stream which is stored or transmitted. But a rendering engine can only select from the repertoire available in the chosen font, so any sign that has to be printed must be mapped to *some* code point. The Standard states that "Glyph shape and methods of identifying and selecting glyphs are the responsibility of individual font vendors and of appropriate standards and are not part of the Unicode Standard" (p. 13), and that "A font and *its associated rendering process* [my italics] define an arbitrary mapping from Unicode values to glyphs" (p. 14). Similarly, Unicode Technical Report #17 states (in connection with the "fi" ligature): "The choice of whether to use a single glyph [which here presumably = grapheme] or a sequence of two is up to the font containing the glyphs and the rendering software". But if a sign has no code point specified in the Standard, it will have no "natural home", as it were, in a font. In such cases font designers will have no particular incentive to standardise the code point for that sign. The result is that they will map it to whatever code they wish, and the highly desirable aim of guaranteeing uniform coding will be lost.

This greatly complicates the creation of rendering software, which has to deal with printed signs. If each font maps its signs to its own idiosyncratic codes the programmer is forced to rewrite the software for each *font*, rather than for each *script*, and this entails a very great increase in labour. Specifying Unicode values for every *printed sign* would encourage a uniform encoding for every font, and this would greatly facilitate implementation. Hence it is desirable that Unicode be as specific and detailed as possible, since the more omissions there are from the Standard the more likely it is that variations in font encoding will occur.

The fact is that there is a clear need for an unambiguous and fixed one-to-one relationship between code points and printed signs. Such an arrangement would have two advantages: first, it would encourage uniformity in character mapping when creating fonts, thereby facilitating text interchange; and second, it would eliminate a constant need for rendering, "derendering" and re-rendering, thereby reducing processing demands. (This point is further elucidated below.) Anything must be an advance if it simplifies text processing by reducing the number of stages required between input, *i.e.* typing, and output, *i.e.* character display or printing. Ideally, the end-user types in the most naturally intuitive way according to the norms of his/her native language and the output displays the correct sign(s) with as few intervening encoding/decoding or code-translation steps as possible.

2. *Characters and glyphs*

The problem of non-standardised mapping is likely to arise with any printed sign which is currently excluded from the Standard. The principal group of excluded signs is that of conjunct consonants and combining forms in general. As this is large and significant group, it is worth examining the basis for their exclusion in some detail.

The decision to exclude conjuncts and combining forms rests on the crucial distinction made in the Standard between a "character" and a "glyph". Two difficulties, however, immediately arise in this connection: the proper definition of these terms, and the consistency with which the distinction is applied.

Definitions. — The definitions of the terms "character" and "glyph", as explained on p.13 of the Standard and expanded in UTR #17, are unclear and confusing. At first sight, it appears that a "glyph" is merely a typographical or font-specific alternative form of a given character. It is immediately obvious how "*a*" or "**a**" or "ɑ" might be regarded as glyphs of "a", as these are well-known and easily recognisable alternatives. One font will represent the letter one way, another font will represent it the other way, but nobody would dispute that they both represent the same thing, namely the small or lowercase form of the first letter in the ordinary English alphabet. Also, it would be possible to substitute one for another in any text, albeit with some sacrifice of typographical elegance, without losing lexical integrity.

However, it is less immediately easy to see how "ﺣ" might be regarded as a glyph of "ح" [U+062D ARABIC LETTER HAH]*, since (as is made clear elsewhere) these forms are not mere typographical alternatives but contextually-determined variations. *I.e.* one cannot at will substitute "ﺣ" for "ح" wherever it appears in a text, since "ح" only occurs in isolation whereas "ﺣ" only occurs in initial position. To some degree, the same phenomenon occurs in those scripts exhibiting differences of case, since (in a sense) case is contextually determined. For instance, in English proper names the occurrence of the capital letter is position-sensitive: thus *e.g.* the form "America" would be accepted in normal usage, while "americA" would not be. It is in this sense that case differences are contextually determined.

Nevertheless, on reading up to the end of p. 13 it would seem possible to conclude that a grapheme is a "glyph" in the Unicode sense if:

(*a*) it is lexically equivalent to a defined character, *i.e.* it is a typographical variation like "*a*" or "ɑ" is to "a"; or

(*b*) it is a position-dependent or contextually-determined variant of a defined character or specified combination of characters, like "ﺣ" is to "ح" or "र्◌" is to "र्".

At first sight this definition appears to tally with the explanation given in the Standard. However, the situation is confused by the Devanagari example given on p. 14, where it is stated that the glyphs which make up the word "पूर्ति" (*pūrti*) are derived by a rendering process from the "characters" प, ◌ू, र, ◌ि, and त. Yet the forms ◌ू and ◌ि are"vowel signs" (*matras*), not "letters of the alphabet" (*aksharas*), *i.e.* they are position-dependent variants of the vowel letters ऊ and इ respectively. They can only occur in association with a consonant. Hence they may be regarded as "joining forms", given that "पऊ" (*pa-ū*) is not read the same as "पू" (*pū*). (In the same way, Arabic "ه ن" is not read the same as "هن", since in the latter case the use of the joining forms makes a single lexical unit of the combination.)

In fact, a true analysis of "पू" would be "प" + "◌्" + "ऊ". The merit of regarding "पू" as a result of "प" + "◌ू" is that it corresponds to normal writing convention, although in fact such correspondence is nowhere recommended in the explanations of the character/glyph distinction given in the Standard (cf. the discussion of Tibetan subjoined forms below). It is not at all clear, and does not follow from the apparent meaning of "glyph", why "र्◌" should be regarded as a glyph while "◌ू" and "◌ि" are accorded the status of characters.

*At this point the Standard actually shows the four glyphs corresponding to the character U+0647 ARABIC LETTER HEH. This character is not used in the present discussion, however, because the glyph chosen to represent ARABIC LETTER HEH in the basic Arabic code chart is "ﻫ", which most closely resembles the initial form "ﻫ" rather than the isolated form "ه". As all the other letters in the chart, including "ح", are represented by their isolated forms, this appears to be an anomaly, particularly as the cognate U+0629 ARABIC LETTER TEH MARBUTA is represented by the glyph "ة". To avoid any possible misunderstanding, therefore, it was deemed advisable to select a character about which there could be no ambiguity, and "ح" seemed as good as any other for making the point.

The fact that "◌र्" can be decomposed to defined characters ("र" + "◌्") can scarcely be the criterion, since the same is true of the other two ("◌ू" to "◌्" + "ऊ" and "ि" to "◌्" + "इ").

Even if one were to accept the above definition of "glyph", however, it hardly seems to justify the treatment of Indic conjunct consonants such as the Devanagari " क्ष " (*kṣa*). In the Standard they are dismissed as "glyphs, not characters" (p. 221), yet they do not appear to satisfy the above criteria for what constitutes a glyph. They are not position-sensitive, nor are they mere typographical variations of established characters. The sign " क्ष " can hardly be regarded as a variant of either of its constituents (" क् ", *i.e.* " क " + "◌्", and " ष "), since its form is quite different. No learner would have difficulty recognising "*a*" as a variant of "a", but he/she would surely need to be told that "क्ष" is a conventional shorthand for " क्‍ष ". In any Devanagari font, "क्ष" will have to be allocated its own code point, which obviously cannot be the same as that of either " क् " or " ष ". It is therefore hard to see on what grounds it may be regarded as a glyph of either of them.

The letter " श ", on the other hand, is undoubtedly a glyph (in the above sense) of " श " [U+0936 DEVANAGARI LETTER SHA], as is " ल " of " ल " [U+0932 DEVANAGARI LETTER LA]. In both cases the characters are typographically interchangeable but for aesthetic reasons they would not usually appear together in the same text. Yet " क्ष " and all the other Indic script conjuncts are excluded from the Standard.

It is true that most Indic conjuncts are not regarded as "letters of the alphabet" and as such are usually not given a place on the keyboard (although some conjuncts, like Devanagari " क्ष " and " ज्ञ ", are very well established in their own right and may in fact be represented). *E.g.* you would expect to find keys for letters such as (say) "ণ" [U+09A3 BENGALI LETTER NNA] and "ড" [U+09A1 BENGALI LETTER DDA] on a Bengali keyboard but not "ণ্ড" (the conjunct consonant *ṇḍa*). You would expect this last to be rendered from the individual letters plus a specific conjunct-generating keypress, which might be virama but is more likely to be a key sequence defined by the rendering engine. Nevertheless, as far as output is concerned, the font designer must assign a specific code to the conjunct.

Consistency. — The situation is further confused by the fact that distinction between character and glyph is inconsistently applied. Indic conjuncts are excluded on the grounds that they are "the result of ligation of distinct letters" (p. 214). However, the Standard then goes on to assign code points to numerous Latin, Armenian, Hebrew, Arabic and Tibetan combinations, ligatures and combining forms, as well as to the Ethiopic syllabic characters and thousands of precomposed Hangul syllable blocks. Examples include "ﬁ" [U+FB01 LATIN SMALL LIGATURE FI], "ﬕ" [U+FB15 ARMENIAN SMALL LIGATURE MEN INI] and "ﭏ" [U+FB4F HEBREW LIGATURE ALEF LAMED].

The most glaring example of inconsistency arises in connection with the "virama model". In the Standard, Devanagari and other Indic script conjuncts are regarded as being derived from a combination of consonant, virama and succeeding consonant. This view is adopted whether the resulting conjunct is a new sign, as in the example of "क्ष" above, or a combination of an existing character and a subscript form, as would be the case in, say, Myanmar "ကွ" (*kwa*). Most such subscripts, or "subjoined forms", are excluded on the grounds that they are glyphs of the "parent" character.

In Tibetan, however, subjoined forms are encoded between U+0F90 and U+0FBC. *E.g.* "◌ྐ" is U+0F90 TIBETAN SUBJOINED LETTER KA. The rationale given for this is: "First, the virama is not normally used in the Tibetan writing system to create letter combinations. ... Second, there is a prevalence of stacking in native Tibetan, and the model chosen specifically results in decreased data storage requirements" (p. 242). From this, it would seem that the criteria for deciding whether to encode subjoined (*i.e.* conjunct) forms are:

(*a*) does the encoding correspond to the way in which the language is normally written? and

(*b*) does the encoding minimise data storage requirements?

These criteria seem eminently appropriate, yet do not seem to be applied in other cases (cf. the discussion of "पू" above). In Sinhala, virama is often used, yet we find several conjuncts are encoded, *e.g.* "ඹ" [U+0DB9 SINHALA LETTER AMBA BAYANNA]. As far as data storage requirements are concerned, the omission of conjuncts, subscripts and joining forms from the Standard actually creates larger files than would otherwise be the case. (This point is illustrated below.)

It is of course true that virama is *sometimes* used when writing Devanagari and other Indic scripts, but only when the construction of a conjunct form is either really impossible or hopelessly unwieldy. The writer would normally write a conjunct automatically, without considering its derivation. This point is forcefully made by Michael Coulson (*Teach Yourself Sanskrit*, 1976, p. 16): "The use of the virāma stroke to cancel the inherent **a** ... is ... a device contrary to the principles of the script, to be used only in the direst emergencies". Similarly, Rupert Snell (*Teach Yourself Beginner's Hindi Script*, 2000, p. 52) says: "But *virām* isn't much used in real writing; it's mostly restricted to technical contexts".

This attitude to virama is reflected in the way that many Devanagari keyboards operate. Although some types of rendering software will automatically generate a conjunct if a user enters a pair of consonants separated by a virama, most require the user to enter control characters or special character sequences. These considerations suggest that native writers do not automatically think of a conjunct as having arisen through the intermediate stage of a disappeared virama. If this is indeed the case, conjuncts and subscripts certainly fulfil criterion (*a*) for inclusion in the Standard.

Curiously, the virama-based encoding method is advocated for both Gurmukhi and Khmer (p. 251), despite the fact that it is rarely (if ever) used in either script. In the case of Gurmukhi, there are only four common combining forms anyway (◌੍ਰ *-r-*, ◌੍ਹ *-h-*, ◌੍ਵ *-w-* and ਯ *-y-*), although admittedly their paucity is not a criterion for including them. In the case of Khmer, the Standard admits quite candidly that the character ◌្ U+17D2 KHMER SIGN COENG is "arbitrary and not rendered" (p. 475)*. Thus, while all three scripts (Tibetan, Gurmukhi and Khmer) eschew virama, only Tibetan is singled out in the Standard for special treatment.

There is also, by the way, a practical aspect to this question. It is sometimes necessary to print the virama character as such, particularly in Myanmar, where the equivalent character "◌်" *'athaʔ* [U+1039 MYANMAR SIGN VIRAMA] is of frequent occurrence. Moreover, *'athaʔ* is not used only to "kill" the inherent *-a*, it is also used denote certain vowels. *E.g.*, "ယ" is *ya* whereas "ယ်" is not *y* but the simple vowel *e*. Thus "လ" is *la* and "လယ်" is *le*. Rendering software must therefore be able to distinguish between the kind of virama which has to be printed, either as part of a final consonant or as part of a vowel, and the kind of virama which signifies "make a conjunct". While this difficulty might be overcome (the Standard suggests using ZWNJ [p. 250], although there are drawbacks to this approach), it nevertheless introduces a tiresome complication into processing.

Other examples of inconsistency. — (1) Arabic joining forms are encoded (in the Arabic Presentation Forms blocks at U+FB50 - U+FDFF and U+FE70 - U+FEFF), yet Syriac joining forms are not. (Nor do joining forms appear to be proposed for cursive scripts not yet included in the Standard, such as Mandaic.) The rationale given for including the Arabic forms is that "They are included here for compatibility with preexisting standards and legacy implementations that use these forms as characters" (p. 197). The question why this rationale cannot equally be applied to Syriac is not addressed.

(2) Separate codes are provided for Latin-script diacritical marks and modifiers such as " ◌́ " [U+0301 COMBINING ACUTE ACCENT] or " ◌̃ " [U+0303 COMBINING TILDE], yet there are no separate codes for Arabic-script combining dot patterns such as " ◌̈ " or " ◌̤ ". (Fonts exist in which such combining forms are used, *e.g.* Ramna Classique.)

(3) Code points are provided for a number of common abbreviations, *e.g.* "&" is U+0026 AMPERSAND, "၎င်း" is U+104E MYANMAR SYMBOL AFOREMENTIONED. In Tamil, however, the common and important abbreviation ஸ்ரீ "*śrī*" (an honorific title) is omitted, as are other common abbreviations, such as ௳ for தேதி *tēdi* "day", ௴ for மாசம் *mācam* "month", ௵ for வருஷம் *varuṣam* "year", நீ for நிலுவை *niluvai* "remainder", ௸ for மேற்படி *mēṟpaḍi* "aforesaid" and ௹ for ரூபாய் *rūpāy* "rupee".

(4) The Hangul Jamo block includes "final consonants", which might legitimately be regarded as "joining forms", despite the fact that the glyphs in the code charts unaccountably omit the dotted circle usually employed to denote the position of the base character. What is the basis for including a character such as "ᆫ" (*i.e.* "◌ᆫ", Korean *-n*) [U+11AB HANGUL JONGSEONG MIEUN] while excluding "◌్న" (Telugu *-n-*)?

*It is important in this context to note that the "virama method" for Khmer is robustly opposed by the Cambodian Committee for Standardization of Khmer Characters in Computers (ISO/IEC JTC 1/SC 2/WG 2 **N2380R** and **N2406**). These documents also contain other objections to the Unicode Khmer block, not all of which have been adequately rebutted in the *Response to Cambodian official objection to Khmer block (N2380)* (ISO/IEC JTC 1/SC 2/WG 2 **N2385**). However, in order to keep the present document to a manageable length, these matters, though important, are not discussed here.

Many other examples, all of similar type to the above, can be found in the Standard. It is hard to see any justification for these inconsistencies.

3. *Alternative placement forms.*

A related, though less serious, issue is the provision of alternative placement forms. In Hebrew, a vowel sign such as ◌ֶ *-e* [U+05B6 HEBREW POINT SEGHOL] is placed centrally under a wide letter such as ה or א , but it is shifted to the right and placed under the stem of ד or ר (or a narrow letter such as ו). Font designers generally allow for this by providing two or even three glyphs for the same vowel sign, each with a different displacement. At present, however, the Standard provides only a single code point for each Hebrew vowel sign.

Similar placement problems occur in other scripts. In Khmer, for example, a subscript vowel such as "◌ូ" [U+17BC KHMER VOWEL SIGN UU] may occur under a subscript consonant (*coeng*) such as "◌្ន" (*-n-*). As the output will require a separate glyph for the lower-position vowel ("◌ូ") anyway, it would be convenient for the code points of such signs to be standardised.

4. *"Unicode fonts".*

According to the explanation given in the Unicode website font FAQ page, a "Unicode-compliant font" is defined, or at least explained, as

> a font that has a "CMAP" that maps all or part of Unicode characters to glyphs in the font. Both new Type1 and Truetype fonts typically have such a CMAP; older fonts usually do not.

A number of fonts claiming to be "Unicode fonts" have already appeared, and these typically contain Unicode-defined characters mapped to the code points laid down for them in the Standard. However, for many scripts a "pure" Unicode font, *i.e.* one in which *only* Unicode-encoded characters were provided, would be manifestly insufficient, because it would lack vital conjuncts and combining forms. Even if the missing signs are provided, however, there are two disadvantages to the Unicode approach. First, any user wishing to access a text file using the font must be in possession of appropriate rendering software; and second, the text files generated by the font will be significantly larger. As an example, let us take the simple Arabic sentence

سأدفع له الفلوس

sa-ʔadfaʕ la-hu l-fulūs ("I shall pay him the money"). In fact, Unicode allows two encodings of this sentence, according to whether or not one chooses to stick to the basic Arabic block (U+0600 - U+06FF) or avail oneself of the joining forms provided in the Arabic Presentation Forms blocks. Using the latter, the coding is (ignoring matters of directionality)

```
FE B1 FE EE FE E0 FE D4 FE DF FE 8D 00 20 FE EA FE DF 00 20 FE CA FE D3 FE A9 FE 84 FE B3
```

i.e. 30 bytes in all. This string, which we may call the "rendered format" version, is a straightforwardly direct encoding of the rendered forms. The alternative method requires the use of ‌ [U+200C ZERO WIDTH NON-JOINER] and ‍ [U+200D ZERO WIDTH JOINER] to indicate where joining characters are to be connected; the character string as stored, though not as displayed, then becomes

س ‍ أ ‌ د ف ‍ ع ␠ ل ‍ ه ␠ ا ل ‍ ف ‍ ل ‍ و س

which we may call the "Unicode format" version. When encoded, this string is

```
06 33 06 48 20 0D 06 44 20 0D 06 41 20 0D 06 44 06 27 00 20 06 47 20 0D 06 44 00 20 06 39 20 0D 06 41 06 2F 20 0C 06 23 20 0D 06 33
```

i.e. 44 bytes in all. Thus the Unicode format version will be almost 50% larger.

However, at least in this case a worthwhile font should include the joining forms in the Presentation Forms block, which will facilitate rendering. For Syriac, however, no such forms are coded in the Standard, so their mapping to code points in a Unicode Syriac font is entirely open. But they must be provided somewhere, since a Syriac font that did not provide joining form signs would be unusable. Presumably, therefore, the joining forms would be encoded in the Private Use Area. From the user's point of view this is inconvenient, because it is likely to give rise to a variety of character mappings for the same script — precisely the problem that it was hoped Unicode would solve.

It is also worth noting how exactly the encoding takes place in practice. No user would expect to type ZWJ or ZWNJ (which in any case are not present on the keyboard). To enter the Arabic sentence above, the user would press the following sequence of keys:

س و ل ف ل ا SP ه ل SP ع ف د SP أ س

and he/she would expect to see the appropriate ligatures in the displayed text. Rendering must therefore occur at the moment of text entry, so there will have to be a “derendering” of the displayed text into Unicode format when it is stored or saved. Likewise the text will have to be decoded, or re-rendered, whenever it is viewed, so the rendering engine will be required both for storage and whenever the text is accessed for display or printing. This requirement must complicate processing.

It is clear that absolutely analogous difficulties will occur with Devanagari and other Indic and Asian Unicode fonts. If there are to going be such things as “Unicode fonts”, they must still provide a complete range of *printed signs*. The subset of notional “characters” is demonstrably insufficient. A Devanagari font which only provided Unicode characters would, for example, be incapable of reproducing the simple Hindi word “स्त्री” *strī* (“woman”), as it would lack the initial *str-* conjunct. If font vendors are going to include conjuncts in their Unicode fonts anyway it would make more sense for them to be assigned permanent values, *i.e.* they should be included in the Standard.

5. *Conclusions*

For storing and displaying text in most languages other than western European languages, the rigorous application of the existing Standard means that —

(1) A significant proportion of character mapping will be left entirely at the mercy of individual font designers and vendors. The result will be a continuing proliferation of *ad hoc* font-specific character maps and the consequent disuniformity of coding that this must entail.

(2) Users must always be provided with the appropriate rendering software for the font(s) used, even if they only wish to read the text, since a direct one-to-one translation of the Unicode code stream into characters would be unacceptable. Thus the rendering engine must accompany the text at all times.

(3) Unicode format files will in general be larger than rendered-format files and will include complex control characters such as ZWNJ and ZWJ.

(4) Unicode format files will require rendering or derendering whenever they are displayed, printed, modified or saved, which means that they will incur greater processing demands.

6. *Recommendations.*

In order to remove the ambiguities and inconsistencies of the kinds described above, it is requested that the following recommendations be considered:

(1) The definitions of “character” and “glyph” should be made much more precise and then rigorously applied.

(2) The Standard should provide additional code points for alternative placement forms where these are clearly required. (This would be helpful but not essential.)

7. *Proposed definitions of “character” and “glyph”.*

It is recommended that the following, or some formally equivalent, wording be adopted:

Character. — A typographical element is a “character” if it is a member of at least one of the following:

I. the complete collective repertoire (sometimes known as the “alphabet”, “abjad”, “abugida”, “syllabary” or “logosyllabary”) of signs (which may be called “letters”, “syllables”, “aksharas” or “ideographs” depending on their phonetic and semantic values), and any position-dependent or contextually-determined variants thereof, established for writing or printing the text of a particular language*;

Examples: A ק ë स ت ـت ـتـ تـ জ ள Ꮃ ཕ λ ぬ 捌

II. the collective repertoire of signs, including those known as "harakat" or "matras", used (either historically or currently) in conjunction with a character of class I to indicate the phonetic association with that character of a particular vowel, or the absence of any inherent vowel, when writing or printing a particular language†;

Examples: ি◌ ◌ े ◌ु ◌ູ ◌ໍ ◌ໍ ເ◌ ◌ູ

III. the repertoire of signs, often called "accents", "tone marks" or "diacritics", used (either historically or currently) to modify the phonetic values of other characters, to indicate the tone of a syllable, or to make a graphic distinction between two otherwise identical characters:

Examples: ◌̃ ◌́ ◌ׅ ◌ّ ◌ٔ ◌ः ◌๋ ˥

IV. the repertoire of signs, sometimes called "compounds", "ligatures" or "conjuncts", derived from any combination of class I, class II and/or class III characters and commonly treated as individual signs in their own right when writing or printing a particular language;

Examples‡: Ǳ ﬁ ﷲ ﻻ [द्ध] [रु] [ক্ষ] [க்ஷ] [ரு] ඩ 강

V. the repertoire of signs, including "half-forms", "joining forms", "subjoined forms" or "coeng pyunhcana", derived from any character or combination of characters in classes I, II/III or IV and which are written or printed before, under, over or after a character, either to form a conjunct or to conform to any other generally-accepted orthographic practice of the particular language;

Examples: ◌ྒ ◌ྒ ᅥ ᅦ [◌] [◌] [◌] [◌] [◌] [◌] [◌] [◌] [◌]

VI. the repertoire of signs, commonly called "abbreviations", which stand for words or phrases in a particular language, and which may be composed of one or more characters in classes I, II or III, but which are commonly treated as individual signs in their own right when writing or printing that language;

Examples: @ & $ № [ஸ்ரீ] ஷ ฿

VII. the collective repertoire of signs, sometimes called "accents" or "cantillation marks", used to show how the text of a particular language is to be read;

Examples: ◌֔ ◌֩

VIII. the collective repertoire of signs, sometimes called "punctuation marks", used to mark the structure of the text of a particular language;

Examples: ? ፡ । ؟ ¶ ॥ · ※

IX. the collective repertoire of signs, other than the conventional script, used to transcribe or encrypt a particular language or group of languages, including invented scripts, phonetic notation, "shorthand" or "code";

Examples: ɔ ʡ ə ɸ [/] [_] [⌡] [⌒] [/] [/] [ϙ] ⣍ ⠑ [▬••] [•▬••▬]

X. the collective repertoire of signs, sometimes called "numerals" or "digits", used in writing or printing numbers in a particular language;

Examples: 4 ٢ ৫ ๕ ௫ 五

XI. the repertoire of signs, sometimes called "symbols" or "signs", used when writing or printing

* Wherever relevant, it is to be understood that both natural and artificial languages and/or scripts are included.
† Most members of this class may be subsumed in class V, the distinction being more linguistic than typographical.
‡ Characters enclosed in square brackets are not encoded in version 3.0 of the Standard.

mathematical, logical, scientific or technical formulæ or operational descriptions;

Examples: ≅ ∫ √ ℜ ∂ ∇ ∑ ÷ ∴ ↄ Ǝ ∪ ∩ ↓ [⏣]

XII. the repertoire of non-printing control codes used when writing or printing computer software, inasmuch as these are not covered in class XI;

Examples: ZWNJ ZWJ CR LF ESC ≆

XIII. the repertoire of miscellaneous non-language-specific signs, including (but not restricted to) those known as "signs", "symbols", "icons" or "dingbats", often used in written texts as abbreviations or for technical, decorative or symbolic purposes and treated typographically as individual signs in their own right;

Examples: ☎ ☆ ✓ ☞ ✂ ¶ ॐ ੴ 〒

XIV. the repertoire of miscellaneous non-language-specific signs, including those known as "signs" or "symbols" used in a recognised notation for recording visual or auditory phenomena, including music, dance, movement or gesture, inasmuch as these are not covered in other classes;

Examples:* [🤙] [✌] [✋] ♪ [𝄞] [𝆖] [🚦] [⏻] [▙]

XV. the repertoire of miscellaneous graphic elements used for line or block drawing or other typographically-based graphic design, inasmuch as these are not covered in other classes.

Examples: ├ ╖ █ ▄ ▒

Glyph. — A symbol is a "glyph" if it is:

I. a recognised typographical variant of a specified character (*e.g.* ɑ for a, or श for श);

II. a font-specific typographic element required only when rendering that particular font but not generally otherwise (*e.g.* the multitude of glyphs which may be required for rendering *Nastaliq*).

Regarding the above definitions, the following points should be noted:

(1) the classes of character identified above are not necessarily mutually exclusive, and the boundaries between them may not always be sharply defined. There will be numerous instances where a particular character might meet the criteria for inclusion in two, or even three. In some cases the choice will scarcely be open to question; in others it will be a matter of individual preference. Those writers for whom the Tamil script is a syllabary, for instance, would place (say) ரு *ru* in class I, while others would regard it as a "combined form" and place it in class IV. Likewise, some may consider that there is no need to establish a separate class for matras such as "ெ◌", since these might legitimately be regarded as "joining forms" and thus incorporated into class V. The important point is not to which class a character may be allocated, but rather the fact that it can be allocated to at least one of them.

(2) these definitions are only intended to give explicit consistency to the meaning of the terms "character" and "glyph" as they are (or should be) used in the Standard. They do not pretend to judge which characters should actually be encoded: they merely enable a judgment to be made as to the status of a candidate sign, leaving wholly undetermined the question of whether that sign, if found to be a character as so defined, is worthy of inclusion. Indeed, it is not intended to suggest that every conceivable character should be encoded willy-nilly. Although a given sign may be a character according to the above definition, it may well be excluded from the Standard on other grounds. No-one would doubt, for instance, that the Klingon *pIqaD* forms a character set of some kind, yet this set has been rejected nonetheless. The criteria on which this kind of judgment is made lie outside the scope of the present

*Version 3.1 encodes musical symbols in Plane 1, and makes an important distinction between the uses of "♪" in musical and non-musical contexts. In the former it is encoded as a note at U+1D160, while in the latter it retains its original encoding as a dingbat (symbol) at U+266A.

discussion.

(3) representatives of all the classes identified above can be found within the existing Standard. In some cases, however, only a few members of a class have so far been included, and to highlight this inconsistency has been a principal objective of this document.

8. *Consequences of adopting the recommendations.*

Should the above definitions be put in place, an immediate effect would be to enable the Standard to incorporate officially recognised conjunct consonants and combining forms (subjoined forms, subscripts or superscripts) where these can clearly be regarded as an integral part of the script in question. Table II below gives very approximate numbers of such characters for the scripts included in Version 3.0, based on a survey of some 25 fonts.

While implementing these recommendations would increase the number of encoded characters, the actual number of characters that would be added is not overwhelming. The scripts enumerated in Table II would add about 1,100 characters to the total, which is not many when it is remembered that the Hangul Syllable block alone accounts for over ten times that number. But in any case numerical restriction is not, and should never be, a reason within the Standard for excluding any character. It is surely not the goal of the Standard to reduce the number of available characters to a minimum. Such an aim would be incompatible with the desire to produce an all-encompassing character set. The additions here proposed could be encoded in a separate "Indic Extended" block, as there would in general be insufficient reserved characters ("▨") within the present blocks for the above scripts. There are good precedents already within the Standard for such an arrangement: the characters of several scripts, including Latin, Greek and Arabic, are already to be found in various blocks dispersed throughout the BMP.

The aim of these recommendations is purely practical, to avoid unnecessary complications and duplication of effort in the creation and dissemination of fonts and their associated rendering software. Nonetheless, they would also be of benefit in enabling many of the objections to the existing Standard to be overcome without demanding that it be drastically rewritten. It is also worth pointing out that the effect of modifying the Standard in this way would only be positive. Greater choice of format would be afforded to users while existing implementations need not be affected, as none of the present encoding would be removed or altered. But the scope of the Standard and its flexibility of application would be enhanced.

	conjuncts	*half-forms*	*subscripts*	*superscripts*	*combining forms*
Bengali	130	28	12	2	–
Devanagari	118	56	6	1	–
Gujarati	51	31	2	1	–
Gurmukhi	–	1	8	–	–
Khmer	–	–	36	–	–
Kannada	35*	–	44	–	–
Malayalam	78	5	13	–	–
Myanmar	11	–	42	1	–
Oriya	104	–	34	1	–
Sinhala	3	1	–	1	–
Syriac	1	–	–	–	24
Tamil	131*	–	–	–	–
Telugu	51*	–	–	–	–
Total	713	122	235	7	24

TABLE II. *Approximate numbers of conjunct and combining form characters associated with scripts included in Version 3.0. The categories are not sharply distinguished, and are to be taken as guides only.* * *Mainly consonant + vowel ligatures.*

II. HISTORICAL OR STYLISTIC OMISSIONS

A preliminary perusal of the Standard has also prompted the following observations:

(1) *Greek / Coptic.* — While it is true that the Coptic script is a variant of Greek with some supplementary letters, the style of the Greek characters used for Coptic is quite distinct, and is only used for that language. Mapping Greek and Coptic letters to the same codes means that users must either change font when they change language or use fonts in which either the Greek or the Coptic characters are assigned to non-Standard codes. A single font incorporating both would be far more convenient. The provision of codes for both styles would seem more properly to satisfy the aim of providing a truly *universal* character set.*

(2) *Cyrillic.* — The same considerations apply in respect of Old Church Slavonic *vis-à-vis* modern Russian.

(3) *Georgian.*— The *asomtavruli* "upper case" forms are rarely used, and I have been told that most Georgian speakers would be at best hazy as to their correspondence with *mkhedruli*. (Indeed, there is even considerable ignorance as to the exact pronunciations of the obsolete *mkhedruli* letters ჱ, ჲ, ჳ, ჴ, ჵ and ჶ.) Dee Ann Holisky (in Daniels & Bright (eds), *The World's Writing Systems*, 1996, p 364) points out that "an attempt by the linguist Akaki Shanidze (1887-1987) to introduce the characters of the old Georgian alphabet *asomtavruli* to mark proper names and sentence beginnings was unsuccessful, though one finds *asomtavruli* characters used as capitals in his own works and occasionally in works written in his honour." To treat the *asomtavruli* as the "capital letter" forms of *mkhedruli*, while providing no codes for the "lower case" *khutsuri*, would therefore seem to be at odds with common practice. Since the letter forms of *khutsuri* and *mkhedruli* differ so markedly, there is surely an argument for providing code points for both. (A further argument in favour of full encoding is that, just as in the above two cases, a writer may require both sets of characters in the same document.)

(4) *Malayalam.* — The increasingly widespread use of keyboards that has accompanied the explosive growth in information technology has prompted a simplification in the Malayalam script. There were formerly numerous irregularities in the forms indicating the addition of *-u* or *-ū* to a base character. While several characters added a "tail", *e.g.* "ഗു" *gu* from "ഗ" *ga* or "തു" *tu* from "ത" *ta*, this was not always the case (*e.g.* "ണു" *ṇu* from "ണ" *ṇa*). In modern printing all these irregular forms have been abolished and *-u*/*-ū* additions are now indicated by the postposed characters ു [U+0D41 MALAYALAM VOWEL SIGN U] and ൂ [U+0D42 MALAYALAM VOWEL SIGN UU]. There may, however, be some argument for providing additional code points for the obsolete forms on historical grounds (cf. the remarks above concerning Coptic, Old Church Slavonic and Georgian).

* In the HTML version of the Roadmap to the BMP a block is provisionally allocated to Coptic at U+2C00 ff. However, the link goes to the proposal for Glagolitic, and the website provides no further information.