L2/01-382
ISO/IEC JTC1/SC2/WG2 N2385
2001-10-11

Universal Multiple-Octet Coded Character Set
International Organization for Standardization
Organisation internationale de normalisation
Международная организация по стандартизации

Doc Type: **Working Group Document**
Title: **Response to Cambodian official objection to Khmer block (N2380)**
Source: **Maurice Bauhahn and Michael Everson**
Status: **For discussion at the 41st meeting of WG2**
Action ID: **ACT**
Date: **2001-10-11**
Distribution: **WG2 and UTC**

Background comments on the existing and the proposed Khmer block:

The complexity of the Khmer Script

Happily, the details of official concerns regarding issues surrounding Khmer encoding have come out in document N2380. The Khmer script is one of the world's most complex, as are all the Brahmic scripts. It has been frustrating for all those who love Khmer to run up against technological limitations. Crude workarounds that violate the principles of the language are in widespread use. And the use of Khmer on computer has been all too often a far step from traditional and pure methods. Fortunately, technology is changing to make it possible for computer practice to move closer to the traditional methods.

A parallel path: the Khmer government and Khmer standardization

Maurice Bauhahn apparently took the first microcomputer into Cambodia in 1982 (an Osborne) and subsequently developed the first Khmer PostScript fonts. When issues of encoding Khmer in Unicode were first broached in Cambodia there appeared to be only one government agency (the Ministry of Agriculture with assistance from IRRI) involved in international standardization bodies – and they were not interested in character encoding. UNESCO organized a conference in Phnom Penh on Khmer Unicode immediately after UNTAC concluded their transitional role in Cambodia. Peter Lofting (who was encouraged by members of the Unicode Consortium to stimulate additions to Unicode of living scripts from developing countries) was invited to lead the discussions. Many government officials took part. Maurice Bauhahn presented the major paper of that conference, suggesting an encoding with explicit subscripts. Shortly afterwards UNESCO organized a similar conference in Bangkok at the Asian Institute of Technology, to which Maurice Bauhahn was also invited. In both conferences, Peter Lofting challenged as being inappropriate the explicit subscript model. He did not find the strong opposition expressed in the N2380 document. Subsequently, there was extensive liaison with Olivier de Bernon of L'École française d'Extrême Orient; Olivier was engaged in transcribing many historical documents from his office near the Royal Palace. His contributions were important in locating unusual and rare forms and eliminating glyph variants of existing characters. In addition, a team of Khmer typists was hired to type the entire Chhuan Nath Khmer-Khmer dictionary – partly with the aim of discovering exceptional characters and constructs that might be needed in computer implementations. Furthermore, Maurice Bauhahn was hired by UNICEF to instruct staff of the Ministry of Education, Youth and Sport on methods to desktop publish the textbooks of the Ministry (partly based on his experience editing a 1600-page medical dictionary in Khmer, French, English, and Vietnamese).

In 1997 the government of Cambodia was asked to convene an advisory panel to discuss matters of Khmer Unicode encoding. The ad hoc committee under the Ministry of Education, Youth and Sport consisted of leading Khmer linguists in Phnom Penh (and one Khmer linguist studying for his Masters degree in Australia, who was visiting at that time). After several days of discussions the committee

drafted a document elaborating what should be included in and excluded from a Unicode standard encoding of Khmer. Although there was no explicit written statement regarding the use of the virama model issued, verbal discussions confirmed that either option (virama model or explicit subscript model) was acceptable to them, so long as all subscripts could be entered. Subsequently the Unicode Consortium encouraged discussions which started on the Unicode mailing list and moved to the Khmer mailing list organized by the Consortium. Michael Everson, Irish representative to JTC1/SC2/WG2, took part in these discussions. Every attempt was made to include all interested parties in the discussions – and sometimes the discussions were heated! Certainly the virama model was one of the issues of controversy. But if there has not been an appropriate official Cambodian representative, it has not been for lack of trying!

The Khmer Philology Project

Subsequently the Khmer Philology Project, together with other bodies, has been organizing conferences demonstrating a Khmer word-processing application. The impression we have is that this KPP project has been driving the current challenge to Khmer as presently encoded in Unicode. Unfortunately, despite many requests, it appears that no alternative Khmer encoding proposals have been made publicly-available from those sessions. In other words, the KPP has not been building consensus or modelling transparency.

It is unfortunate that the Cambodian government has recently been so heavily influenced, in this case, by such short term interests. An understanding of the principles underlying this counter proposal will, we hope, make this evident. There needs to be a balance between the encoding principles of Unicode and ISO/IEC 10646 on the one hand, and the pressures for immediate implementation on the other. Khmer issues have not been brought to the Unicode mailing list by the persons offering the proposal contained in N2380. And although there have been some criticisms on the Cambodian-based Inetlist mailing list of the Khmer Unicode encoding, alternatives have not been presented, and certainly not under the spotlight of Unicode principles. Quite possibly, the push for immediate results has overlooked more substantive issues which would plague Khmer script implementors for generations if the subscript-model were to be adopted. It bodes ill that further proposals regarding Khmer sorting, Khmer keyboarding, and the Khmer locale have been routinely ignored on the Inetlist.

Foundations of the existing Khmer Unicode encoding

The existing Khmer Unicode encoding, in contrast, was freely debated and discussed. It was the result of many years of investigation and analysis and benefited from the contributions of many different experts, including the top linguists of Cambodia. No vested interests influenced the decisions, nor was it imposed from the top down.

The fundamental principles of the existing Khmer Unicode encoding were several:

- The Khmer script encoding must be open-ended enough to accommodate all languages regularly transcribed in the script (including Pali, Sanskrit, and minority languages)
- The encoding must be character-based and phonetically ordered (just as is the case with spelling and handwriting)
- Ambiguity must be avoided: there must not be two different ways to accomplish the same thing
- The encoding must facilitate sorting, searching, and text-to-speech, and must not be limited to graphic elements
- The most important efficiency to be achieved is that of typing speed.

Character-based and phonetically ordered

The second principle above (encoding must be character-based and phonetically ordered) is the single greatest obstacle to mutual agreement on the implementation of Khmer. All new encodings in Unicode must be based on this principle (some legacy standards, such as those for Thai and Lao, contained departures from those principles). It is a fact that adherence to this principle takes a Khmer Unicode implementation out of the reach of legacy applications. The best that implementers in Cambodia can do for legacy applications is to have an unambiguous glyph-based encoding that could be relatively easily converted into Khmer Unicode in the future (which is what the KPP seemed to be working on).

There are a number of reasons why the virama model was chosen for Khmer. In the first place, Khmer is historically a Brahmi-derived script. The virama model does derive from ISCII. The advantage of using this encoding model for Khmer is that it is well-understood and that it works for almost all Brahmic scripts. This makes it *far* easier for industry to implement for Khmer. (Thai and Lao followed the long-entrenched glyph-based Thai model, and Tibetan has unique stacking behaviours – for long vertical stacks – which the virama model could not handle well.) For Khmer, allowing system software developers and font developers to use a familiar model will be of great benefit to Cambodians. Sorting and searching algorithms can also use a familiar model for Khmer implementation. Transliteration between Brahmic scripts – is likewise made far simpler. Pali and Sanskrit electronic texts produced in Myanmar or Sri Lanka can be made available to Cambodian readers far easier if the same encoding model is used for Khmer, Myanmar, and Sinhala.

It would appear that the fundamental question is: Should the Khmer encoding have been a kludge to fit with a glyph-based encoding model, or should it have been a truly phonetically-ordered character-based model? The existing Khmer Unicode encoding follows a phonetically-ordered character-based model.

How to handle this challenge?

It is disappointing that this delegation from Cambodia has proposed to take out of Unicode a well thought out encoding of Khmer to replace it with something not yet thought out and as-yet unpublished. Such an action can only lead to further delays and perpetuate conflicting implementations into the future. It is generally agreed that an earlier attempt with Korean to do something similar was a mistake. Even more so it would be a mistake to do something similar now with Khmer. The existing Khmer Unicode encoding has long predated the competing encoding which, as of this writing, still has not taken shape.

Furthermore, combining explicit subscripts with the VIRAMA/COENG model would lead to two paths to each and every subscript – an utterly chaotic option which would make effective searching and spell checking impractical for years to come. An explicit encoding of subscripts by itself would probably prevent the encoding of dictionaries such as that of Chhuan Nath and many historical documents in Khmer, Pali, and Sanskrit. See the separate section below on this specific issue.

On the other hand, we would encourage the Cambodian authorities to propose additional characters still lacking which are needed to encode the wealth of Khmer history, literature, minority scripts, and scholarship. We would be happy to work with those authorities on the addition of consonants needed by minority scripts and signs needed to transcribe Indic languages.

Responses to the issues raised in the Appendix of N2380

1 The characters អឝ U+17A3 and អឞ U+17A4 are discrete sortable entities in Pali dictionaries written in the Khmer script, each taking up separate chapters at the start of those dictionaries (in Khmer អំ U+17A2 sorts at the end). It was at the insistence of the Khmer linguists committee that both of these characters are in the encoding. They distinguished the consonant U+17A2 from the independent vowel U+17A3. At some time in the future it might be advisable to revisit the distinction between U+17A2 and U+17A3 – but for the time being the jury is still out on that issue. If necessary, the second of these two characters (U+17A3) could be deprecated, but it would not be a good idea to make an abrupt decision that would later have to be reversed – again.

2 Character ឨ U+17A8 (KHMER INDEPENDENT VOWEL QUK) does indeed have its roots in a combining of ឧ U+17A7 (KHMER INDEPENDENT VOWEL QU) and ក U+1780 (KHMER LETTER KA). Since it is a very rare character, however, it seemed appropriate to avoid breaking the much more numerous combination of U+17A7 (KHMER INDEPENDENT VOWEL QU) and U+1780 (KHMER LETTER KA) which never ligate in current usage. There is another independent vowel ឩ U+17AA (KHMER INDEPENDENT VOWEL QUUV) which has a ligature-like relationship (with the consonant វ U+179C, KHMER LETTER VO), but it also is not a true ligature for the VO component is likewise ignored in sorting.

3 It is true that the independent vowels ឪ U+17B1 (KHMER INDEPENDENT VOWEL QOO TYPE ONE) and ឫ U+17B2 (KHMER INDEPENDENT VOWEL QOO TYPE TWO) are variants one of the other, as their names imply. On the other hand they are regularly both used even in the same sentence in different forms; for instance, one word, ឫស, can contrast with ឪស). There is apparantly another word which requires the QOO TYPE TWO form, but we don't have it to hand as of this writing. in that instance, the two are not simply glyph variants, despite their similar sound.

4 The two inherent vowels ឣ U+17B4 (KHMER VOWEL INHERENT AQ) and ឤ U+17B5 (KHMER VOWEL INHERENT AA) by definition lack glyph forms. They are represented in the Unicode standard with pseudo-glyphs to help distinguish them. It was important to encode them, however, in that they are referenced in verbal spelling and some dictionary orderings. Furthermore they would be important in text-to-speech applications. The shorter form is inherited from other Indic languages whereas the long form is native Khmer. However they would only have a specialist use and are not expected to be used in ordinary text.

5 The decision to not encode the ligatures ◌ុ U+17C6 (KHMER VOWEL SIGN U) + U+17C6 (KHMER SIGN NIKAHIT) and ◌ាំ U+17B6 (KHMER VOWEL SIGN AA) + U+17C6 (KHMER SIGN NIKAHIT) was a bold move by the Khmer linguists committee. Indeed it appears to fly in the face of Khmer textbooks and the Chhuan Nath dictionary introduction. On closer inspection, however, it is obvious that it was the right decision. First of all, note that the name NIKAHIT distinguishes this character from pure vowels whose names are purely phonetic (their names mimic their sound). Second, note that this is not the only combination which creates a unique vowel. The signs ◌ុ U+17C6 (KHMER SIGN NIKAHIT), ◌ះ U+17C7 (KHMER SIGN REAHMUK) and ◌ៈ U+17C8 (KHMER SIGN YUUKALEAPINTU) combine with various dependent vowels to create about 18 additional vowels. If one would accept the two suggested ligatures, it would also be necessary to add the 16 other vowels, thoroughly complicating the life of already overworked typists! Third, the ambiguity of

whether to type a combined ligature of a dependent vowel and NIKAHIT (or REAHMUK or YUUKALEAPINTU) or to use separate parts would violate the principle of "ambiguity must be avoided". Note that sometimes those signs are stand alone (used only with the inherent vowel). Fourth, Khmer sorting would not be simplified by encoding combined characters. Khmer requires a syllabic-based sort which is *much* more complicated than default Latin-based algorithms allow. An additional conversion of dependent vowel plus sign to create a separately sorted vowel would place a minor load on such a sort...which would probably be key-based rather than live in any case. Fifth, it is good to learn from problems a similar encoding has caused. Compatibility decompositions defined in The Unicode Standard 3.1 make it very difficult to classify the constituent parts of such a ligature (note U+0E33 THAI CHARACTER SARA AM and U+0EB3 LAO VOWEL SIGN AM), because these both decompose into a combining mark followed by a base character, with the former combining with some preceding character.

6 The discussion concerning the virama model is given above. Note that the virama model is also extensible to the Khmer practice of subscripting some independent vowels: ឣ

ហ្ស្ទ្យ can be represented by ហ HA + ◌្ COENG + ្ស RY + ្ទ TO + ◌ SAMYOK SANNYA + ្យ YO;

ប្ង្ផ្ន can be represented by ប BA + ង NGO + ◌្ COENG + ្ផ QU + ផ PHA + ន NO;

ផ្ធ can be represented by ផ PHA + ◌្ COENG + ធ THO.

The one consonant ឡ LA which appears to have no subscripted form is easily handled, as a spelling error, by the virama model: if someone were to type ក KA + ◌្ COENG + ឡ LA, the result would simply be ក្ឡ (see the note at point 8 below).

7 It is a welcome observation that the Khmer Unicode encoding is lacking the TUTEYASAT. We strongly encourage the delegation to propose the addition of that sign, and will work with them to facilitate this .

8 The rationale for the code point ៗ U+17D8 is in fact brought out in this argument against it! The function of all these various glyph forms resembles that of the Latin clause: *et cetera*. Hence they should be encoded with one code point and rendered using the end user's glyph variant preference.

9 There is a misunderstanding here. As pointed out earlier the two so-called ligatures (ឥ and ឦ) are not true ligatures. The two independent vowel entities have accumulated unique identities. ក៓ should be represented by KA + WIRIAM. ក៓ could be represented by KA + WIRIAM + VARIATION SELECTOR-1.

It is true that the glyph used for COENG in the standard is misleading in this regard. ◌្ should not have been used, and we propose that the glyph be changed to ◌្, as some implementers such as Microsoft are currently using in their development. This will have the advantage to the Khmer typist in indicating that the following character entered will be a subscript, so: ក KA + ◌្ COENG will yield ក +, and then the next K entered will appear correctly in a conjunct as ក្ក. The other presentation form variants (ligatures) discussed in the paragraph with this number (showing special context-dependent forms of, ត្ត and ន្ត should be handled by a separate protocol such as a formatting language for a "feature" of an intelligent font. *Intelligent fonts are required for Khmer.*

10 Unfortunately this paragraph does not adequately consider the complexities of Khmer sorting (syllabic sorting). Some preprocessing will be necessary whatever encoding is used. A preliminary study is already available.

Specific discussion of explicit subscript vs. virama encoding

Maurice Bauhahn discussed with the official group of Khmer linguists in Phnom Penh (who formed the basis of Khmer Unicode) the issue of subscripts. Those linguists were not opposed to the method we arrived at of handling subscripts using "coeng" (so long as we had appropriate subscripts in the final display). Originally Maurice, too, felt that subscripts should be encoded explicitly. However there are several reasons why he (and others who have thoughtfully considered the matter) backed away from that. Here are seven of those reasons:

1 We keep finding subscript forms that had not been anticipated. We should avoid creating a standard with explicit subscripts that do not exist (!) but also avoid inserting subscripts which never occur. In one stroke the COENG character sidesteps the dilemma. To date five subscript independent vowels in words (◌ឝ U+179D, ◌ឣ U+17A3, ◌ឨ U+17A7, ◌ឫ U+17AB, ◌ឯ U+17AF) and all ten digits in subscript form for lunar dates have been documented. Also, Khmer digits function in a Khmer language Sanskrit grammar as stand-ins for grammatical tagging (although these should probably be handled with another protocol). We have to remember that the encoding is not just for the Khmer language, but also for any and all languages which are written with the Khmer script. We have barely scratched the surface of what forms might be needed for rarer Sanskrit forms (for example, we have only recently become aware of three additional signs which probably need to be encoded in Khmer Unicode in connection with Sanskrit transliteration).
2 Nomenclature is a very important part of intent. None of the Khmer subscripts have a unique name: they are always called "coeng" + the name of the base form from which they are formed. Their phonetic value is the same as their equivalent base form (and either base or subscript form can lose its inherent vowel). If you would listen to a Khmer language class reciting spelling, you would always hear that "control" word "coeng" preceding *every* subscript. And the subscript name would otherwise be identical to the base form name. Furthermore, within their rank of sorting subscripts sort in the same order as their corresponding base characters.
3 Encoding subscripts separately will not allow the Khmer script to take advantage of legacy applications and avoid the technological leap to intelligent rendering. The complications of the Khmer script require an intelligent display mechanism (with a many to many relationship between glyph and character) if a one-character, one-code rule is applied. A glyph-based encoding limited to one code per character is doomed to failure. The following requirements of the Khmer script make a glyph-based encoding irrelevant:
    i subscripts cannot be reliably encoded as characters at any particular vertical position [at least eight of the consonantal subscripts (◌ឆ U+1785, ◌ឍ U+178E, ◌រ U+179A, ◌ំ U+179C, ◌ស U+179F, ◌ហ U+17A0) and one of the independent vowel subscripts (◌ឫ U+17AB) occur in a second subscript position as well as a first subscript position]
    ii any phonetic ordering of Khmer will have to handle at display-time the large number of pre-consonantal forms (six different minimalist vowels which share a pre-consonantal glyph identical to the glyph for vowel ◌ែ U+17C1 [◌ើ U+17BE, ◌ឿ U+17BF, ◌ៀ U+17C0, ◌ោ U+17C4, ◌ៅ U+17C5, as well as ◌ែ U+17C1 itself], two other pre-consonantal vowels [◌ុ U+17C2, ◌ូ U+17C3] and two glyph variants of the subscript form of ◌រ U+179A [KHMER LETTER RO] which are also pre-consonantal)

iii two-glyph vowels (ᩤ U+17BE, ᩥ U+17BF, ᩦ U+17C0, ᩧ U+17C4, and ᩨ U+17C5) of which one part precedes the consonant and the other has multiple positions/ligation

iv obligatory ligation forms would need at least three redundant forms of ᨷ U+1794 (KHMER LETTER BA): ᨣ U+1794 + U+17B5, ᨤ U+1794 + U+17B5 and stand-alone ᨷ U+1794

v two obligatory subscript variant forms of ᨬ U+1789 [KHMER LETTER NYO]

vi two obligatory variant forms for both of the register shifter characters ◌ U+17C9 and ◌ U+17CA [KHMER SIGN MUUSIKATOAN and KHMER SIGN TRIISAP] – which incidentally are identical in shape to the glyph for ◌ U+17BB [KHMER VOWEL U]

vii dozens of optional base consonant/vowel or subscript consonant/vowel ligatures which are the norm for written Khmer.

4 Explicit subscripts would lead to a substantial expansion of the Khmer Unicode keyboard that would be increasing hard to type and complicated to process. For example, there are 34-35 key positions (in the first three rows) which are relatively easy to type (plus 13 in the top row)...not enough space (without time-consuming helper key input) to put all the base consonants (35), dependent vowels (16), and common independent vowels (14), other Khmer signs and punctuation (33-35), let alone 40+ consonant/independent vowel subscripts! In addition there are up to 50 non-Khmer characters which also need to be accessed, some extremely frequently (word break ZWSP, phrase break SPACE, arabic numbers, currency signs, lower ASCII signs/punctuation/parentheses/quotation signs). In total with explicit subscripts one would have to squeeze about 190 characters on to the keyboard with 49 character keys. I.e., you will have to hit at least two keys to type subscripts in any case. The options for expanding the keyboard to include subscripts are: (1) COENG + base form which gives visual feedback with the COENG (and takes twice as long to type as an unaided key), (2) a dead key + SUBSCRIPT which does not give visual feedback with the first touch (which also takes twice as long), or (3) one or two helper keys + SUBSCRIPT which takes 3-4 times as long to type than an unaided key.

5 There should not be two different combinations to express the same word. This would result in chaos for searching/grep-like algorithms in the future. If subscripts were added, COENG and all its advantages would have to be deleted.

6 In the past, Khmers concerned about computerisation have regarded only the visual aspect of their language, but computers have to also understand the characters behind that encoding. For example, simplistic entry of the subscript ᩆ U+178F KHMER LETTER TA (◌᩠◌) when it should be subscript ᩈ U+178A KHMER LETTER DA (which has an identical glyph) could change the sorting/searching properties of the word typed. By using a coeng + base form of data entry, the native typist is drawn to enter the proper form because the base forms are distinct. There are an additional three characters which optionally have one glyph form (◌ U+17BB, ◌ U+17C9, ◌ U+17CA) – another reason to avoid glyph-based data entry.

7 The discussions with the linguist committee focused on how to bring technology into confirmity with the Khmer language, not maintaining legacy technology ways of working with the Khmer language! The KPP seems to have unwittingly turned that around. Computer keyboards of the last ten years were evidences of legacy technology (and they were completely different from typewriter keyboards of the preceding decades which represented another legacy technology where individual glyphs had to be composed of constituent parts).

Least is best! We have not seen the vowel encoding that is proposed, but fear a very problematic (two valid-paths to one vowel) encoding has been suggested. The existing encoding of vowels has only one valid path to each vowel. Maurice Bauhahn's recent Khmer sorting research paper (KhmerSortingUnicodebeta.pdf) includes a list of 44 explicit dependent vowels which have examples that affect sorting in the Chhuan Nath dictionary, which integrate well with Khmer Unicode that was based on a "least is best" principle. Hence, the encoding based on fundamental linguistic principles yields a result that anticipates future needs. The linguist committee in Phnom Penh was aware of some of those – only recently have we been able to flesh out the whole (thanks to having the entire Chhuan Nath dictionary in digital form).