ISO/IEC JTC1/SC2/WG2 N 1497

The United Nations University

L2/97-028

UNU/IIST

International Institute for Software Technology

Telephone +853 712 930 Fax +853 712 940

Mailing Address:
UNU/IIST
P.O. Box 3058
Macau

Visiting Address:
18/F Ed. Banco Luso Intl.
1-3, Rua Dr. Pedro Jose Lobo
Macau

Date	E-mail	Our ref	Your ref
January 10, 1997	ny@iist.unu.edu		

Dear all,

We have been studying the Report of the 3rd International Mongolian Encoding Meeting (Beijing, China, August 1996) and know that this Draft Proposal will be discussed at the WG2 Meeting #32 in Singapore in January 1997.

We believe that the part of the proposal dealing with using control symbols to generate variants of characters is flawed, specifically that it can lead to ambiguities which are impossible to resolve.

We would like to ask relevant bodies to give some consideration to the following analyses and suggestions for using control codes in the Mongolian Script Encoding System.

Yours sincerely,

Namsrai Yumhayar, Richard Moore, Myatav Erdenechimeg

UNU/IIST Macau

cc:

encl.: [1] On the Use of Control Symbols in the Mongolian Script Encoding

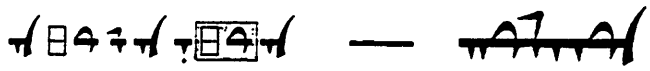
On the Use of Control Symbols in the Mongolian Script Encoding

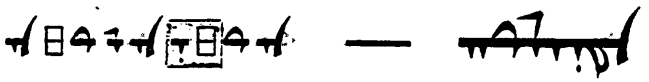
Yumbayar Namsrai, Richard Moore and Myatav Erdenechimeg
UNU/IIST
Macau
10 January, 1997

We have been studying the Report of the 3rd International Mongolian Encoding Meeting (Beijing, China, August 1996) including the draft proposal for the encoding of the Mongolian character set. We believe that the part of the proposal dealing with using control symbols to generate variants of characters is flawed, specifically that it leads to ambiguities which are impossible to resolve and that it means that certain character combinations cannot be generated.

The first problem stems from the fact that there exist variants of characters which are generated by writing some control symbol (#172, #173 and #174) after the basic character, and other variants, both of different characters and of the same character, which are generated by writing the same control character before the basic character. This means that it is in some cases impossible to resolve the meaning of (sub)strings containing control characters. One place where this can occur is in the (sub)string "**character1 control character2**", in particular when the variant forms "**character1 control**" and "**control character2**" are both possible. Similarly, combinations such as "**control1 character1 control2 character2**" can be ambiguous when "**control1 character1**" and "**control1 character1 control2**" are both possible variants of character1 and "**control2 character2**" is a possible variant of character2. And there are of course other situations when the same thing occurs.

One might try to argue that the rules for the language should always allow these to be unambiguously resolved, but we do not believe that this is in fact the case. Specifically, we would cite as an example the name "Atlanta" (i.e. the American city). If this is written in Mongolian, it contains just such an unresolvable ambiguity: it is not clear whether the character belongs with the character preceding it or with the character following, so there are two possible presentations of this sequence of codes as shown in the diagram below.

a) 

b) 

And even if it were possible to generate a set of rules which do ensure that all Mongolian texts can be correctly resolved, we would point out two things:

cannot be coded: the combinations of the Mongolian Nirugu with a basic character which are listed there always produce the variant form of the basic character because the coding says that the Nirugu is to be interpreted as a control character in these positions. For example in the first entry in this column the input method for the letter ᠨ is given as Mongolian Nirugu (-) followed by the basic form of the Mongolian letter A (ᠠ). The coding for this combination would be #12 #32 which according to the table represents the third medial form of the Mongolian letter A and which would therefore be printed as ᠬ.

The same is true of the combinations involving the Mongolian space which are given on page 1 of the same document, though the result in this case is perhaps not quite so obvious visually.

We therefore think that the draft proposal needs to be altered, but before giving our proposal we would also point out that the fact that the three additional control symbols appear after the alphanumeric (printing) characters in the coding list is inconsistent with most other cases in which all control codes appear before the alphanumeric characters. We believe that this may make it difficult (maybe even impossible) to use some standard software packages, particularly those related to the automatic processing of Mongolian texts (for example, sorting Mongolian text).

Based on the above, we would propose that the following changes be made to the current version of the Mongolian Encoding System:

1. That the Mongolian space (#0) and the Mongolian Nirugu (#12) should not be used to generate variants of characters.

2. That in the codes for generating variants of characters using control symbols the control symbols should only appear after the character with which they are to be associated.

3. That, for a given letter, the set of all possible variants of that letter (i.e. not taking the position of the letter within a "word" into account) should be considered as being ordered.

4. That the control symbol(s) should be used simply to identify the position of a particular variant within that ordering. This of course also identifies the particular variant uniquely.

5. That all control codes should appear in the list of characters in the same place as all similar non-printing symbols, i.e. before all the printable characters.

This ensures not only that all encoded strings can be unambiguously interpreted but also that all possible combinations of characters can be coded.

The actual choice of the ordering of the variants of each character is theoretically arbitrary: any one is just as good as any other as far as satisfying the above requirements is concerned.

As an example of how this scheme would work, the variant forms of the Mongolian letter A (see the Mongolian Reference Table) would be coded as follows:

glyph	way of input	its code
		#32
	*	#32 #X1
	**	#32 #X2
	***	#32 #X3
	****	#32 #X4
	...	#32 #X5

Another possible way of avoiding the problem but using fewer (in fact four) control codes is to represent the position of a particular variant in the list of variants as a binary number and to have control codes representing moving through the list by one position, by two positions, by four positions, and by eight positions. If we denote these four control symbols by Y1, Y2, Y4 and Y8, then the third variant form of the letter A would be represented as A-Y2-Y1, and in general the position in the ordered list of variants is obtained by adding together the "values" of the control characters. This is a sort of compromise position between the first two: it has an intermediate number of control codes on the one hand, but on the other some variants can only be represented using 2 or even 3 (in the case of a seventh variant) control characters. Note however that again only one physical control key would be needed to input these symbols: each of them could be generated automatically by software simply by counting the number of times that single control key is pressed and then converting that number to a binary representation. In this case the table shown above looks like:

glyph	way of input	its code
		#32
	*	#32 #Y1
	**	#32 #Y2
	***	#32 #Y2#Y1
	****	#32 #Y4
	...	#32 #Y4#Y1

It is not clear to us which of these schemes of control characters is the best.