SC22/WG20 N938

Title: **Disposition of comments of ballot results on FPDAM-1 to ISO/IEC 14651:2001**

Date: **2002-06-11**

Project: **JTC 1.22.30.02.02**

Source: **Alain LaBonté, Project editor, on behalf of SC22/WG20**

Status: **Information required according to directives by SC22 Secretariat**

References: **SC22 N3352, N3392 (WG20 N936)**

Action: **For national bodies consideration**

SUMMARY OF VOTING AS PER SC22 N 3392

"P" Members supporting approval without comment
8 (Canada, Czech Republic, Denmark, Finland, Japan, Republic of Korea, Netherlands, United Kingdom)

"P" Members supporting approval with comments
1 (Norway)

"P" Members not supporting approval
1 (United States of America)

"P" Members abstaining
1 (Switzerland)

"P" Members not voting
13 (Austria, Belgium, Brazil, China, Egypt, France, Germany, Ireland, DPR of Korea, Romania, Russian Federation, Slovenia, Ukraine)

O-member Sweden approves with comments.

Disposition of comments

1 Norwegian comments accompanying approval vote

NORWAY 1.

NOR.1: The template table needs to be aligned with IS 12199 and CEN/ENV 13710 European Ordering rules. This would mean that all Latin letters be sorted on level 1 as their base letters, and that special characters will be ignored at level 1.

Not accepted. Although the template can not satisfy every requirement at once, it shall be tailored to the users' needs. Furthermore, the template is an international reference on which deltas are based and even though new versions add definitions, the already weighted characters need stability. Norway is invited to provide an example delta for the next amendment for possible inclusion in informative annexes.

NORWAY 2.

NOR.2: Cyrillic letters need to be sorted in a fully deterministic way. Fully composed characters and corresponding character sequences should sort equally on level 3 but differently on level 4.

Not accepted for the same reason as in « Norway 1 ». Norway is invited to provide an example delta for the next amendment for possible inclusion in informative annexes.

NORWAY 3.

NOR.3: Control characters U0000..U001F and U007F..U009F shall have distinct sorting code on level 4 to make the sorting fully deterministic.

Not accepted. Although the result of sorting a file is not fully predictable with the current table in certain edge cases, this is the result of previous consensual committee decisions, due to strong opposition from some countries. Norway is invited to document extra text for a possible future informative annex indicating how to produce stable sorts in those conditions (which is a trivial process if UCS coding equivalent to input coding is used as the last collating key in case of absolute equality otherwise).

NORWAY 4.

NOR.4: The template needs to be specifying a fully deterministic ordering, as also decided in previous ballots.

Not accepted. At some point it was agreed that if one body did not want to produce fully determistic data-wise, it could, and the

result would still be conformant. What was then judged the most important was that ordering be reasonably predictable when visible data was presented to the user. Norway is invited to document extra text for a possible future informative annex indicating how to produce stable sorts in those conditions (which is a trivial process if UCS coding equivalent to input coding is used as the last collating key in case of absolute equality otherwise).

2 Swedish comments

SWEDEN 1.

1. There is a symbol generation bug for parenthesised Hàn in the draft new CTT: e.g. <U3232> "<SFF40><SE709>";...
should be <U3232> "<SFF40><TE709>";...
(Note S --> T) Etc. for other parenthesised Hàn.

Accepted. This will be fixed. This is a bug in the "sifter" program.

SWEDEN 2.

2. None of the five-hexdigit S-symbols in the draft new CTT are declared as they should be.

Accepted. It is also a bug.

SWEDEN 3.

3. The first level weighting split into groups according to SC22/WG20 N898R, String ordering weighting roadmap, so that Brahmic derived scripts and Hangul are collated according to collation clustering. Note that Hangul in addition needs some extra prehandling for this to work properly, but detailing that part has always been seen as out of scope for 14651.

Not accepted. It is not specific enough about which sets of grouping would be where. Furthermore Brahmic-derived scripts do not all behave in the same way. For Hangul see disposition of USA comments.

SWEDEN 4.

4. The weights for Hangul letter cluster jamos should reflect the letter content of each cluster jamo. Similarly for the letter cluster Hangul compatibility letters. The weighting for the circled and parenthesised Hangul should be based on the (incomplete) syllabic content. See SC22/WG20 N891R, Hangul ordering rules, for details.

Not accepted. For sorting old Hangul, the suggestion would lead to correct results. However for modern Korean sorting only (without compatibility characters), the current results seem to be relatively correct (if representation is restricted to one-code-position for each complex letter), according to the Korean expert present at the SC22/WG20 Tromsø meeting. Sorting compatibility Hangul characters requires extra steps which are outside the scope of ISO/IEC 14651. A note will be added in the CTT comments to that effect. (Info: the Korean expert present at the Tromsø meeting indicated that its national body would like to write an explanatory text to that effect too, for a possible informative annex of ISO/IEC 14651).

3 USA comments accompanying disapproval vote

The US votes NO with comments on FPDAM 1 to ISO/IEC 14651 (SC22/WG20 N890; L2/01-470). Its vote will be changed to YES if the following two problems are addressed. (Other than to address these two problems, the US prefers the weights that are present in the table).

USA 1.

1. Numerics

Due to production problems in generating the data tables, the following item TC4 from L2/01-330 was not implemented in the current data table, although it was accepted (see the disposition of comments on the PDAM, SC22/WG20 N882.) It needs to be implemented to prevent formal ordering problems and maintain synchronization with UCA.

TC4. Modify handling of secondaries for Numerics. These are to be weighted consistent with the approach used in other constructed secondaries (not involving an accent), such as in:

<U16AA> <S16A8>;"<BASE><VRNT1>";"<COMPAT><MIN>";<U16AA> % RUNIC LETTER AC A
Thus, the following example for a Mongolian digit

<U1811> <S0031>;<MONGL>;<MIN>;<U1811> % MONGOLIAN DIGIT ONE
will become

<U1811> <S0031>;"<BASE><MONGL>";"<MIN><MIN>";<U1811> % MONGOLIAN DIGIT ONE
The list of numeric script secondary symbols to which this should be applied are the following:

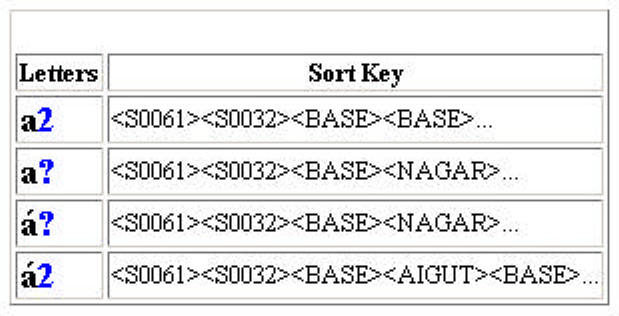
<NEGATIVE>
<SANSSERIF>
<NEGSANSSERIF>
<ARABIC>
<EXTARABIC>
<ETHPC>
<NAGAR>
<BENGL>
<BENGALINUMERATOR>
<GURMU>
<GUJAR>
<ORIYA>

<TAMIL>
<TELGU>
<KNNDA>
<MALAY>
<THAII>
<LAAOO>
<BODKA>
<MYANM>
<KHMER>
<MONGL>
<CJKVS>

Background. Look at the following example, with:

<U0061> <S0061>;<BASE>;<MIN>;<U0061> % LATIN SMALL LETTER A
<U00E1> <S0061>;"<BASE><AIGUT>";"<MIN><MIN>";<U00E1> % LATIN SMALL LETTER A WITH ACUTE
<U0032> <S0032>;<BASE>;<MIN>;<U0032> % DIGIT TWO
<U0968> <S0032>;<NAGAR>;<MIN>;<U0968> % DEVANAGARI DIGIT TWO

The following shows how combinations of the first two and second two sort:



Letters	Sort Key
a2	<S0061><S0032><BASE><BASE>...
a?	<S0061><S0032><BASE><NAGAR>...
á?	<S0061><S0032><BASE><NAGAR>...
á2	<S0061><S0032><BASE><AIGUT><BASE>...

Notice that in the first two cases we get 2, then Devanagari 2; while in the second two cases we get the reverse. This is clearly wrong; the wrong secondary weights are being compared to one another. To prevent these cases, UCA is adding the following invariant:

For all collation elements,

3. All secondaries in non-ignorables must be strictly less than those in primary ignorables.

4. All tertiaries in primary ignorables must be strictly less than those in secondary ignorables.

In general, all Level N weights in Level N-1 ignorables must be strictly less than those in Level N-2 ignorables.

The accent in a-acute is a primary-ignoreable, and must thus have a secondary weight less than the secondary weight in Devanagari digit two. While there are different ways to produce this, the easiest way to do this is to expand the Devanagari weight into:

<U0968> <S0032>;"<BASE><NAGAR>";<MIN>;<U0968> % DEVANAGARI DIGIT TWO

Accepted. The only impact is a change in the table generated automatically by the "sifter".

USA 2.

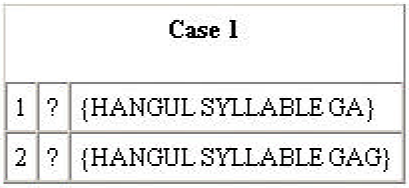
2. Hangul Sorting Problem

To maintain the synchronization between ISO/IEC 14651 and the Unicode Collation Algorithm, the US requests that the primary values for JUNGSEONG and JONGSEONG characters be made higher than any other weights in the Default table. In no case will this result in worse sorting results, and it does preserve synchronization.

<U20000>..<U2A6D6> <S20000>..<S2A6D6>;<BASE>;<MIN>;<U20000>..<U2A6D6> % Han Extension B
<U1160> <S1160>;<BASE>;<MIN>;<U1160> % HANGUL JUNGSEONG FILLER
....
<U11F9> <S11F9>;<BASE>;<MIN>;<U11F9> % HANGUL JONGSEONG YEORINHIEUH
<PLAIN> % Maximal level 4 weight

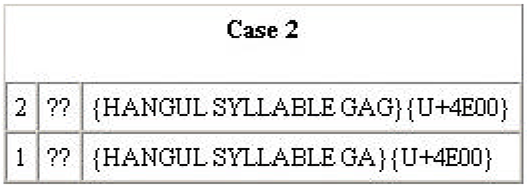
This change does not preclude adding descriptions of possible preprocessing steps with similar objectives, as some other national bodies may request.

Background. *The UCA currently sorts Hangul as follows. ISO/IEC 14651 does the same, whenever NFD (decomposed) data is used, or when archaic Hangul syllables (requiring the use of Jamo) are used.*



Case 1		
1	?	{HANGUL SYLLABLE GA}
2	?	{HANGUL SYLLABLE GAG}

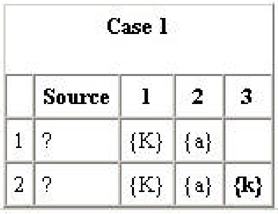
Notice that GAG comes after GA in Case 1. But in Case 2, it comes before. That is, the order of these two Hangul syllables is reversed when each is followed by a CJK character.



Case 2		
2	??	{HANGUL SYLLABLE GAG}{U+4E00}
1	??	{HANGUL SYLLABLE GA}{U+4E00}

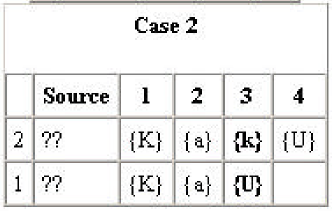
This is not acceptable: when two characters A and B have different primary order, appending another independent primary-weighted character C to each should not affect the ordering. (Independent means that AC and BC do not form contractions, interact in normalization, or are subject to Thai rearrangement).

Why does this happen? All characters are decomposed when sorting in UCA, to preserve canonical equivalence. (This is the logical procedure -- optimizations can be used as long as they have the same order). This results in the following comparisons being made:



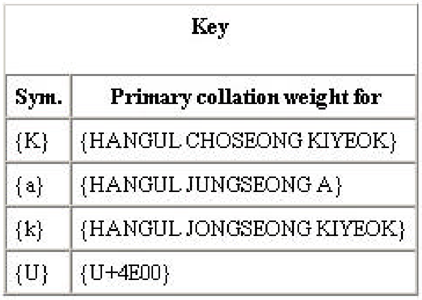
Case 1

	Source	1	2	3
1	?	{K}	{a}	
2	?	{K}	{a}	**{k}**



Case 2

	Source	1	2	3	4
2	??	{K}	{a}	**{k}**	{U}
1	??	{K}	{a}	**{U}**	



Key

Sym.	Primary collation weight for
{K}	{HANGUL CHOSEONG KIYEOK}
{a}	{HANGUL JUNGSEONG A}
{k}	{HANGUL JONGSEONG KIYEOK}
{U}	{U+4E00}

Look at column 3 in Case 1 and 2.

In Case 1, {k} is compared against nothing, and sorts later. (This would also happen if {k} were compared against characters with low primary weight, such as A, or Alpha). This is the correct ordering.
In Case 2, the {k} is compared against {U}. Since all CJK characters sort above all other characters, this has the effect of reversing the ordering of rows 1 and 2, which is incorrect.

The Unicode Technical Committee has considered this issue, and for a number of reasons has approved the following solution. In particular, this solution normally has no performance or sort-length impact on the UCA. Collation implementations are extremely sensitive as to both performance and sort-key length, so this is a very important feature. It also has the advantage of essentially no impact on the standard implementations, since it only changes three constants used in the UCA algorithm. The changes that have been approved for UTR #10: Unicode Collation Algorithm are:

1. In **7..1.3 Implicit Weights**, an area of 1024* high primary weights is reserved, by changing the BASE weights from:

FFC0
 CJK Ideograph

FF80
 CJK Ideograph Extension A/B

FF40
 Any other code point

to

FBC0
 CJK Ideograph

FB80
 CJK Ideograph Extension A/B

FB40
 Any other code point

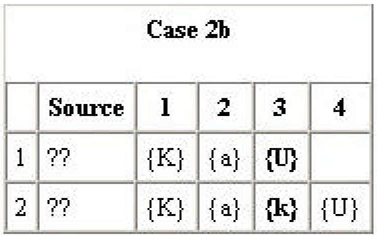
* 1024 is sufficient room, given that multiple primaries can always be used if necessary, as **in 6.2 Large Weight Values**).

2. In the **Default Unicode Collation Element Table**, the trailing Hangul characters are changed to have primary weights in the Fxxx range, e.g. FCE0..FD7E. These include:

1161 ; [.16E0.0020.0002.1161] # HANGUL JUNGSEONG A
....
11F9 ; [.1773.0020.0002.11F9] # HANGUL JONGSEONG YEORINHIEUH
3. Since the assignment of CJK Ideographs has changed, the dependent characters are modified, such as

U+3280 CIRCLED IDEOGRAPH ONE

Because of these changes, the JUNGSEONG and JONGSEONG characters are assigned primary weights in a high range, higher than any other characters. Thus the above Case 2 changes to:



Case 2b

	Source	1	2	3	4
1	??	{K}	{a}	**{U}**	
2	??	{K}	{a}	**{k}**	{U}

Because {a} and {k} now have high weights, higher than anything (e.g., {U}) that might follow them, the right order results. The only further issue is the case of multiple lead characters. The UCA and 14651 have mechanisms that can be called into play in this case, described in Section **3.1.1 Multiple Mappings**. For example, suppose that the Hangul Syllable is of the form LLVT instead of LVT (this happens with archaic

Hangul). If the LL is to be sorted as a unit, then it would require the addition of a contraction, so that the LL mapped to a single primary. If the second L is to be sorted as if it were trailing, then this would require a contraction-expansion, as described in 3.1.1. There are a small number of LL cases -- these can be easily tailored for environments requiring the sorting of archaic Hangul.

Note: Such a strategy can also be used for other languages. For any case where trailing characters in a sequence (grapheme cluster, conjunct, etc) are given primary weights above any other characters, tailoring to high weights can produce the right results.

Not accepted. This would solve, in a compatible way for the users of pure Korean, the problem of intermixing high weight characters such as Chinese characters by putting Korean weights higher than those other scripts'. However the Korean expert present in Tromø believes that further study is necessary to see if there are not other serious side effects.

Preprocessing would still be required, in any way, to make sure that syllables are always systematically sorted together. This remains a requirement in all cases for Korean.

------------- End of this disposition of comments -------