To: Unicode Technical Committee
From: Richard Ishida, summarising arguments from Greg Eck, Andrew West, et al.
Subject: U+202F NNBSP Impact on Mongolian Options
Date: 29 July 2015

NNBSP Impact

[Richard Ishida produced this summary for the UTC based on discussions led by Greg Eck and in response to a call for discussion at the UTC by Addison Phillips, on behalf of the W3C i18n WG.]

A group of Mongolian experts representing the major Mongolian font developers and others keen to implement standardized solutions to handling of Mongolian script text has been discussing NNBSP on a W3C list (https://lists.w3.org/Archives/Public/public-i18n-mongolian/).

NNBSP is used in Mongolian to connect suffixes with preceding words or suffixes. It produces various effects on shaping of the adjacent characters plus a small gap, but the word base plus all suffixes should be handled in implementations as a single word.

The fact that NNBSP is a space currently creates problems in a variety of implementations, including Word, for proper handling of Mongolian for word counting and selection. There has been talk of impact in the area of parsing but this as yet unconfirmed.

Some of the community feels that it would be best to create a new, Mongolian character (at U+180F) to replace NNBSP; others argue that it would be sufficient to change the properties of NNBSP. One concern is the impact that changing properties would have on uses of NNBSP in other languages, eg. French.

The following edited excerpts are attempts to summarise the discussion so far.

The first is an edited version of a summary by Greg Eck, who would prefer a solution that involved keeping NNBSP and just changing its properties, if such a solution is viable, but tries to reflect the concerns of those arguing for a new character.

==========

Implementations (both fonts and utilities) dealing with the Mongolian script and specifically U+1800-18AA have used the following control characters in the following manners:

1.) NNBSP - sole purpose is to separate the Stem+Suffix and the Suffix+Suffix context (with space) while at the same time keeping the given contexts connected as a word.

2.) MVS - sole purpose is to separate the Stem+Orkhitz_A/E context (with space) while at the same time keeping the given contexts connected as a word.

3.) FVS1/FVS2/FVS3 - designed to tag the previous character in such a way that the OT rulings can modify the preceding character.

4.) ZWJ/ZWNJ - provide simulated environments for stand-alone isolate/initial/medial/final contexts.

5.) ZWJ - prevent/allow OT ligaturing; break otherwise expected OT rulings.

Observations from our current discussions:

1.) NNBSP gives the following problems in the current Mongolian script Utilities functionality

- Considered to be a space in the case of most programming languages and embedded routines and therefore gives undesired results in parsing processes

- Breaks the word as seen in word counting, word jumping, sorting, parsing

2.) The character properties of the MVS are probably identical in all ways to the "desired_NNBSP". However, the idea of adding NNBSP functionality into the MVS is infeasible as there are identical contexts that the MVS and the "desired_NNBSP" need to distinguish.

3.) The current functionality of the ZWNJ is not compatible with the desired functionality of the NNBSP in Mongolian , as the ZWNJ affects the joining behaviour of preceding and following Mongolian letters in one particular way (selects non-joining forms), but NNBSP affects the joining behaviour of preceding and following Mongolian letters in a different way (selects non-joining form for preceding letter but may select an initial, medial or final form of the following letter depending on the suffix) - Andrew West.

4.) NNBSP problem of breaking words might be fixed by defining a new Word_Break Property Value "Mongolian" similar to the value "Katakana"

5.) NNBSP problem of being classified as a space/white_space cannot be solved. It seems apparent that the use of NNBSP in other languages as a bona fide space makes it unreasonable to request that it be reclassified as a non_space.

6.) The MVS went through several iterations of design/re-design before the current set of character properties were stabilized. As there were unknowns in the stabilization of the MVS, it is not known now whether more problems will creep in with future upper-level processing using the "a_modified_NNBSP". With the history of the MVS refinement in mind, the idea of modifying the NNBSP over a possible lengthy period of testing and refinement is problematic.

7.) The current state of documents circulating which use the NNBSP now is unknown. My guess is that it is a low figure.

8.) Both Badral and Jirumutu have mentioned a code base dealing with the current problematic NNBSP implementation. I have a small code base dealing with the NNBSP now myself. To update my code base to a new character is not a problem. To leave my code base as it is with the NNBSP does not seem to be a present problem. But then, my current set of utilities is small also. Could I ask Jirimutu and Badral to give more detail as to how specifically the new_NNBSP character would help in their upper-level processing? How specifically, the old_NNBSP has broken their systems? Others?

9.) Even considering the possible implementation of a new Word Break property as mentioned above - that it would indeed keep a suffix attached word intact - there is still the unsolved problem of the NNBSP being a bona fide space - a property that is unchangeable.

==========

Andrew West provided some additional thoughts about the options:

==========

If you are going to make a proposal for a new character you will need to give specific examples of incorrect behaviour, and explain why this incorrect behaviour cannot be remedied by tweaking Unicode properties or the UCA. On the Unicode internal (Unicore) mailing list Asmus Freytag suggested that the word break property of NNBSP could be changed so that by default there would be no word break when the character before and after it belonged to the same category (e.g. both letters, as is the case for Mongolian). Making this change should solve the word boundary issue, as early as Unicode 9.0 next June if someone makes a proposal to the UTC soon, but encoding a new character will take at least two years, possibly much longer if there is opposition from ISO national bodies.

It may take a while before Word catches up with changes to the word break property, but it would take even longer for Word to support a new character. In my opinion, the main advantage of property change over encoding a new character is that the property change will fix existing Mongolian text, whereas the new character will have no effect on existing Mongolian text, and users will still complain that word selection etc. does not work for pre-new-character Mongolian text (and users will not even start to use the new character until it is not displayed as an empty box on their system, and it produces the expected shaping behaviour, which will probably be several years after the several years to get it encoded).

A further problem with encoding a new character is that when it is eventually supported by fonts and rendering systems, Mongolian text with NNBSP and Mongolian text with the new character will look the same to end users, with the result that users will start to complain that internet searches and local find/replace functions do not work correctly for Mongolian because searching for a Mongolian word with the new character will not match the same word with NNBSP and vice versa. And this problem will never go away, because no-one is going to magically change existing Mongolian data, and input methods and users will continue to use NNBSP in place of the new character for years to come -- why not? they both look the same and produce the same visual result.

All in all, I firmly believe that encoding a new character will create more and worse problems than it solves.

==========

The W3C i18n WG would appreciate a recommendation from the UTC regarding how best to proceed.