L2/19-114

Proposal to focus line break prevention design on end-user input

For consideration by Unicode Technical Committee

2019-03-21
Marcel Schneider (charupdate@orange.fr)

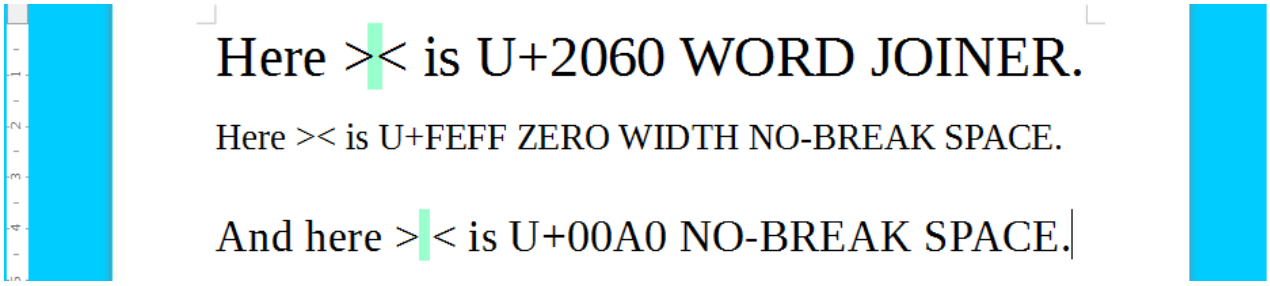
“We must always say what we see.
Above all we must always
— which is more difficult —
see what we see.”

Alain Finkielkraut quoting Charles Péguy

Problem

Still today, the presence in Unicode since 2002 of the character called WORD JOINER is a nightmare in terms of user experience and UI design, to such an extent that this is likely to impact the overall reputation of the Standard. Some environments strongly favor the new U+2060, while others are not eager to give it the expected breakthrough after a decade of U+FEFF ZERO WIDTH NO-BREAK SPACE both for glyphless line break prevention and as an encoding scheme signature. At font level, some important families are supporting the one, and some others the other, but scarcely any both (example below). That brings the need to design keyboard layouts featuring both WJ and ZWNBSP, in isolation as well as to emulate a justifying no-break space for use in word processors (where U+00A0 NO-BREAK SPACE is tailored as fixed-width; please see below and refer to the upcoming *proposal to clarify the purpose of U+202F NARROW NO-BREAK SPACE*), or even a non-breaking thin space (see the upcoming *proposal to enlarge the availability of a non-breaking thin space*). And it almost mechanically inflates the number of layout variants designed in an attempt to broaden font support for automated spacing of big punctuations in French. Writing up end-user documentation under such circumstances is a constant challenge, as it is desirable to spare users the ordeal of getting an unpleasant idea of so widely used a standard.

A good example of software urging end-users to abandon ZERO WIDTH NO-BREAK SPACE and to adopt WORD JOINER is LibreOffice Writer, as it displays visual feedback where WORD JOINER is used, but not where ZERO WIDTH NO-BREAK SPACE is present. This screenshot has been taken on Windows 7 with customized field background color (gray by default). The field background is used in OpenOffice and LibreOffice word processors to help users of the software check the presence of U+00A0 or U+2060:



Exclusion of the ZWNBSP from support by this feature is fueled by ***The Unicode Standard;*** please see the Background section below.

Additionally, “WORD JOINER” may be considered a misnomer needing special attention when describing it in end-user documentation, as this term does not accurately express the identity of this character. The function of U+2060 is not to “join words” like HYPHEN does, but to keep characters together for the purpose of line breaking, so that it is used in Korean (TUS § 18.6 Hangul, p. 736), and more generally to cancel any break opportunity that it is placed before or after, empowering authors to convert any breaking character into a non-breaking one. This use case is described in TUS § 23.2, p. 871 *Word Joiner*:

> U+2060 WORD JOINER behaves like U+00A0 NO-BREAK SPACE in that it indicates the absence of line breaks; however, the *word joiner* has no width. The function of the character is to indicate that line breaks are not allowed between the adjoining characters, except next to hard line breaks. For example, the *word joiner* can be inserted after the fourth character in the text “base+delta” to indicate that there should be no line break between the “e” and the “+”. The *word joiner* can be used to prevent line breaking with other characters that do not have nonbreaking variants, such as U+2009 THIN SPACE or U+2015 HORIZONTAL BAR, by bracketing the character.

Actual user experience is in contradiction to Unicode recommendations with respect to both the character and the number. Using U+FEFF instead, and once only, has enabled Word user Jakub Stachu to post a solution [please scroll down to the second post of the author; I can’t get message IDs any more] on Microsoft Community forum for Non-breakable space justification in Word 2016, proposing to compose a justifying no-break space out of U+0020 SPACE and U+FEFF ZERO-WIDTH NO-BREAK SPACE, by inserting SPACE followed by ZWNBSP. (The idea of an **emulation,** applied here to a justifying no-break space in an environment where the actual NO-BREAK SPACE is tailored to be fixed-width, is mentioned in TUS § 24.1, ¶ *Decompositions*, page 913, with respect to the WORD JOINER.) This solution using ZWNBSP instead of WJ in 2017 (15 years after encoding of WJ) is an application of the principle that Jukka K. Korpela expressed on Unicode Public on August 4 and August 5, 2011:

> […] I surely will use non-conformant things if they help to get things done and conformant things won’t.
>
> […] ZWNBSP vs. WJ. These control characters do the same job in text, as per the standard, so the practical question is simply which one is better supported.

When commenting the abovementioned post, I didn’t notice that the unsupport of WORD JOINER in Word simply boils down to a font issue, that Doug Ewell identified in reply:

> […] opening Word 2007 and pasting a snippet of text with embedded ZWNBSP does display correctly, while the same experiment with embedded WJ shows a .notdef box. This seems to be a font-coverage problem, amplified by Word's silent overriding of user font choices—changing the font from the default Calibri to DejaVu Sans (and optionally back to Calibri) makes the display problem go away, but of course no user could reasonably be expected to go through that.

There is probably a need to hint that this font issue is due to a flaw in the Unicode Standard not specifying that conformant rendering engines should be able to avoid calling for font-provided information whenever the character is only a format control, or even a whitespace, as every space has a default width that can be computed based on current font metrics, Philippe Verdy argued on Unicode Public by several occasions.

The encoding proposal for U+2060 (please see next section for more details) was optimistic about a quick move “over the next few years” of “implementations” towards using the new character. In reality, that move

was a matter of getting probably millions of end-users change their habits, autocorrect entries, keyboard layouts and other input methods, while end-users can see zero benefit from the change, as “carelessly” concatenating files was a mistake that only the relatively narrow community of programmers and extensive command-line users could commit. Therefore, big companies have no real incentive to implement the change and to spread the word, the less as even the expected benefit of being able to “freely delete” U+FEFF is compromised by the mass of data that was authored before WORD JOINER was encoded, and that would never be upgraded.

Consistently, by fall 2017, the fonts shipped with Windows 10 version 1607 mostly did not support WORD JOINER; only Times New Roman, Segoe UI Symbol, Microsoft Sans Serif, Javanese Text, and Myanmar Text families did. But nearly none of these fonts supporting WORD JOINER has a mapping for ZERO WIDTH NO-BREAK SPACE; only the Microsoft Sans Serif family does have both. On the other hand, six families still had only ZWNBSP: Calibri (including Calibri Light), Consolas, Lucida Sans Unicode, Malgun Gothic Semilight, Segoe UI (and the italic fonts of its light, semilight and semibold variants), and Tahoma. Hence, as long as fonts are supposed to contain all characters needed in running text, and software does not switch back from a fallback font, end-users choosing Times New Roman for Serif, and Calibri (default in Microsoft Word) for Sans Serif, need keyboard layouts supporting both ZWNBSP and WJ, and appropriate documentation.

Asmus Freytag brought the topic to the point in the above-cited thread on August 5, 2011:

> The ambiguity of an initial FEFF was not desirable, but this discussion shows that certain things can't be so easily "fixed" by adding characters at a later stage.
>
> The more time elapsed between encoding of the ambiguous character and the later "fix" the more software, the more data, and the more protocols exist that support the original character, creating backwards compatibility issues.
>
> Incidentally, this is totally what I expected when the WJ was proposed, but sentiment in favor of its addition ran high at the time...
>
> The ZWNBSP was present in Unicode 1.0 (1991) while the WJ was added in 3.2 (2002), that is about 10 years later. We are now an additional 10 years down the road, and instead of clarifying the issue, the interim result is that WJ has muddied the waters instead.
>
> Somewhere here are lessons to be learned.

This proposal suggests to revert the functional deprecation of ZERO WIDTH NO-BREAK SPACE, and also to add a conformance requirement of being able to handle format controls (and spaces; please see subsequent proposals) by default without calling for font support (while fonts would keep the status of overriding defaults). The purpose is to clear the ground for the upcoming proposals of this series, and to allow to keep the focus of the one about NARROW NO-BREAK SPACE on that particular character instead of making it an omnibus proposal.

Background

This issue may be considered an unfortunate adverse effect of the elsewhere successful action aiming at palliating the issues reportedly introduced at merger with ISO/IEC 10646. The story goes that Unicode

intended the code point U+FEFF only as byte order mark, whereas ISO/IEC JTC1 SC2 WG2 insisted upon overloading it with an additional break preventing semantics. A decade later, Unicode brought up several improvements, among which namely the combining grapheme joiner (a character able to protect combining marks against canonical reordering; L2/00-156-R1), and the word joiner (a character able to prevent line breaks; L2/00-258).

The emotionally charged tone of the zero-width word joiner proposal — expressions like “innumerable” and rhetoric stress have been redacted when the rationale was mirrored in TUS — suggests a context of frequent and passionate complaints about not being able to keep using legacy file handling processes, and a strong desire on Unicode’s side to stop a semantic interference that was continuously pointed as a nasty design flaw. Hence the proposal was among the top priorities for UTC meeting #84 (agenda: L2/00-185), and was later approved on October 6, 2000 by WG2 (L2/00-369).

But while the COMBINING GRAPHEME JOINER was approved at the UTC meeting — where also the file *BidiMirroring.txt,* cited in the first proposal of this series, was officialized — (L2/00-186 and L2/00-187), the WORD JOINER does not appear anywhere in the meeting minutes. The Open Action Items list from October 27, 2000 (L2/SD2) mentions “word joiner” among “the Unicode accepted characters which have not yet been proposed to WG2.” And motion #7 of UTC meeting #85 (L2/00-323 and L2/00-324), almost unanimously approved (Microsoft voted yes, only Apple abstained) recorded the word joiner among the additions approved by WG2.

The Unicode Standard does not list ZERO WIDTH NO-BREAK SPACE among the line break controls in table 4-10, page 190 of TUS 12.0, and does not mention it when describing *Characters Ignored for Display* (§ 5.21, p. 250) — yet does include it when enumerating *Characters Ignored in Cursive Joining*, one page above — and when it describes *Zero Width No-Break Space* in § 23.2 Layout Controls, p. 871:

> In addition to its primary meaning of *byte order mark* (see “Byte Order Mark” in Section 23.8, Specials), the code point U+FEFF possesses the semantics of ZERO WIDTH NO-BREAK SPACE, which matches that of *word joiner*. Until Unicode 3.2, U+FEFF was the only code point with word joining semantics, but because it is more commonly used as *byte order mark,* the use of U+2060 WORD JOINER to indicate word joining is strongly preferred for any new text.

Adding however:

> Implementations should continue to support the word joining semantics of U+FEFF for backward compatibility.

The need for continuous support of the break-preventing semantics of U+FEFF is underscored detailedly in § 23.8 Specials, ¶ *Byte Order Mark (BOM): U+FEFF,* p. 894:

> For compatibility with versions of the Unicode Standard prior to Version 3.2, the code point U+FEFF has the word-joining semantics of *zero width no-break space* when it is not used as a BOM. In new text, these semantics should be encoded by U+2060 WORD JOINER. See “Line and Word Breaking” in *Section 23.2, Layout Controls,* for more information.

Consistently, since the WORD JOINER has been encoded, the ***Code Charts*** show a comment at U+FEFF:

- use as an indication of non-breaking is deprecated; see 2060 instead

Much understandably any software is deterred from providing the end-user with visual feedback where ZERO WIDTH NO-BREAK SPACE is present, even when a corresponding feature is implemented for WORD JOINER, since doing otherwise may be feared to be considered a sign of non-respect towards the Standard.

The **Unicode Standard Annex (UAX) #14: *Unicode Line Breaking Algorithm*** notes in section 5.1 Description of Line Breaking Properties, ¶ WJ: Word Joiner, after introducing ("These characters glue together left and right neighbor characters such that they are kept on the same line.") and showing a table listing U+2060 and U+FEFF:

> The word joiner character is the preferred choice for an invisible character to keep other characters together that would otherwise be split across the line at a direct break. The character FEFF has the same effect, but because it is also used in an unrelated way as a *byte order mark,* the use of the WJ as the preferred interword glue simplifies the handling of FEFF.

The simplification induced by using U+2060 WORD JOINER instead of U+FEFF ZERO WIDTH NO-BREAK SPACE seems to be widely overstated, as according to the thread Using awk to remove the Byte-order mark on *Stack Overflow,* properly handling BYTE ORDER MARK (formal alias of U+FEFF; for instance, deleting it prior to concatenating files), while that character is also present in file bodies with break prevention semantics, is only a matter of one short line in code or terminal.

Proposed changes

Simplicity for end-users should be prioritized over simplicity for programmers. Now as the break preventer has been duplicated, the adverse effects of the resulting pervasive dichotomy between pre-Unicode-3.2 support and Unicode-3.2-upwards support should be mitigated in order to prevent end-user experience from being deteriorated by problems related to keyboarding or other input methods, by font problems, and by display issues.

The first step would probably be to equalize the status of WORD JOINER and ZERO WIDTH NO-BREAK SPACE with respect to line break prevention, by removing the functional deprecation of the latter, and by urging implementers to treat both on an equal footing. The expected result is that new font versions support both characters, and that word processing software applies the same syntactic highlighting to both WJ and ZWNBSP.

For UAX #14, the following rewording of the above-quoted paragraph is proposed:

> To keep characters together that would otherwise be split across lines at a direct break, 2060 and FEFF may be used interchangeably. Given that FEFF is also used in an unrelated way as a *byte order mark* (BOM), the glue semantics may be represented uniformly by 2060 in those environments that need to be able to delete all FEFF in order to repair a file output by a process improperly concatenating files with BOM signature.

A thorough solution would include that the Unicode Standard specifies font-independent default handling of format control characters at rendering as a conformance requirement. It's a bit late for imposing that new constraint, given that virtually all text-processing software on the marketplace would need to be thoroughly updated. A more lenient option would be to issue it as a recommendation, but the short-term efficiency on user experience of that option is uncertain.

References

Please refer to the references cited in the body, most with hyperlinks.

Acknowledgments

Thanks to everyone who directly or indirectly helped put this paper together.

Thanks to Google for Google Search.

Thanks to Microsoft for Word Online and OneDrive.