L2/17-344R2

Clarify handling of ill-formed UTF-8

Markus Scherer 2017-oct-25

The Unicode Standard “recommends” in chapter 3 (Unicode 10: page 128) certain “Best Practices” for replacing byte subsequences that are not well-formed UTF-8 with one or more U+FFFD.

Proposal

a) Drop the terms “recommends” and “best practice”.

> These terms in this part of the standard have been interpreted as indicating requirements. They have no particular meaning in the standard beyond their customary meanings.

b) Refer to the W3C Encoding Standard for justification.

> This provides both a way of reference by name and a justification for the practice.
>
> Historically, this is circular: Unicode developed this text in response to the W3C’s desire to standardize such behavior, and then W3C did adopt this behavior with reference to the Unicode Standard. The main argument for Unicode to keep describing this behavior is now compatibility with the W3C Encoding Standard.

c) Clean up, simplify, and clarify the description and examples.

> The current description is convoluted and repetitive. I would like to work with the editorial committee on a new and improved version.

Chapter 3 proposed changes

Proposed changes indicated with strike-throughs, underlines, and [bracketed comments].
`Changes and additions indicated in typewriter style are from discussion at the UTC meeting 2017-oct-25.`

[Previous text defines well-formed UTF-8 byte sequences, forbids interpreting ill-formed subsequences as characters, forbids consuming valid successor bytes as part of ill-formed subsequences.]

Although a UTF-8 conversion process is required to never consume well-formed subsequences as part of its error handling for ill-formed subsequences, such a process is not otherwise constrained in how it deals with any ill-formed subsequence itself. An ill-formed subsequence consisting of more than one code unit could be treated as a single error or as multiple errors.

[new paragraph break]

For example, in processing the UTF-8 code unit sequence <F0 80 80 41>, the only formal requirement mandated by Unicode conformance for a converter is that the <41> be processed and correctly interpreted as <U+0041>. The converter could return <U+FFFD, U+0041>, handling <F0 80 80> as a single error, or <U+FFFD, U+FFFD, U+FFFD, U+0041>, handling each byte of <F0 80 80> as a separate error, or could take other approaches to signalling <F0 80 80> as an ill-formed code unit subsequence.

~~***Best Practices for Using U+FFFD***. When using U+FFFD to replace ill-formed subsequences encountered during conversion, there are various logically possible approaches to associate U+FFFD with all or part of an ill-formed subsequence. To promote interoperability in the implementation of conversion processes, the Unicode Standard recommends a particular best practice. The following definitions simplify the discussion of this best practice:~~

→ [change to]

An increasing number of implementations `are` adopting the handling of ill-formed subsequences as specified in the **W3C Encoding Standard** [link in footnote or similar], in order to achieve consistent U+FFFD replacements. `The Unicode Standard does not require this practice for conformance.` The following definitions simplify the discussion of this practice; they are followed by a detailed example.

D93a Unconvertible offset: An offset in a code unit sequence for which no code unit subsequence starting at that offset is well-formed.

D93b Maximal subpart of an ill-formed subsequence: The longest code unit subsequence starting at an unconvertible offset that is either:

a. the initial subsequence of a well-formed code unit sequence, or
b. a subsequence of length one.

[Following text discusses details and examples – to be editorially simplified and clarified.]

References

http://www.unicode.org/versions/Unicode10.0.0/ch03.pdf

August 2017 UTC meeting:

B.16.2 Request to Retract Consensus 151-C19 and Action Item 151-A134 (see L2/17-168) [Sivonen, L2/17-197]

[152-C21] Consensus: Retract Consensus 151-C19.

W3C Encoding Standard, section 8.1.1 UTF-8 decoder, which refers to the Unicode “best practices”.

(= https://www.w3.org/TR/encoding/#utf-8-decoder)