L2/17-168

Illegal UTF-8: Recommend maximal subsequences according to extended lead bytes

Markus Scherer 2017-may-11

Proposal

For definition and discussion of *D93b Maximal subpart of an ill-formed subsequence*, create two options:

a) For all but UTF-8, use the existing definition of D93b, equivalent to a state machine that walks strictly valid sequences.

b) For UTF-8, recommend evaluating maximal subsequences based on the original structural definition of UTF-8, without ever restricting trail bytes to less than 80..BF. For example: <C0 AF> is a single maximal subsequence because C0 was originally a lead byte for two-byte sequences. Similarly, <E0 9F 80>, <F4 90 80 80> and <FD 81 82 83 84 85> are single maximal subsequences.

Note: If at an Unconvertible offset there is a byte 80..BF or FE or FF, it is an illegal “subsequence of length one” in either case.

Background

Several versions ago, the Unicode Standard added recommendations for which subsequences of illegal code unit sequences to report as units of error handling. *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.

This makes sense for most charsets, but is not natural for UTF-8. In particular, the originally documented structure for UTF-8 defines more structurally well-formed byte sequences than those that are valid Unicode UTF-8 sequences. The Unicode Standard text includes examples where the “maximal subpart” is different depending on which definition is used.

References & details

Unicode Standard

http://www.unicode.org/versions/Unicode9.0.0/ch03.pdf p.127 “Best Practices for Using U+FFFD”

The origin of the text there about best practices for using U+FFFD was the discussion and resolution of PRI #121 in August, 2008: http://www.unicode.org/review/pr-121.html

Unicode feedback

Date/Time: Sat Jan 21 17:12:39 CST 2017
Name: Karl Williamson
Report Type: Other Question, Problem, or Feedback
Opt Subject: Best practices for replacing UTF-8 overlongs

> A little over a month ago, I wrote a question to the unicode mailing list concerning the current rules in TUS for handling overlongs. Its message id was <20083c6b-c861-b197-5fdb-d091daaeb517@khwilliamson.com>. In short, I believe the best practices are wrong. This started a thread of comments, but no official explanation from any one in Unicode as to the rationale of why it is the way it is. Ken Whistler explicitly declined to defend the current text. And I found out that implementations, like ICU, do it the way I think it should be done.
>
> I would like the text changed to promote the ICU implementation as the best practice.

Markus 2016-dec mailing list response

... some of the discussion in this thread is due to details that were not spelled out in the PRI. There is basically a 2a and a 2b, while the examples in PRI #121 work the same in both variants.

2a. As Richard said, "The natural logic is to read the requisite number of continuation bytes, converting the whole to a codepoint value, and then check that the codepoint value is allowed in UTF-8. Obviously one also has to check that the requisite continuation bytes are present."

This naturally treats overlong sequences, surrogate-code-point sequences, and 5/6-byte sequences (and prefixes thereof) as single errors.
(I suppose that lead bytes above F4 could be somewhat debatable.)

(This is what ICU does for UTF-8.)

2b. The text in the standard represents the workings of a state machine that walks strictly valid sequences. Overlong/surrogate/etc. sequences become multiple errors.

(This is what ICU converters do for multi-byte charsets like Shift-JIS and GB 18030.)

In my opinion, 2a. "feels right" for UTF-8, because of the history and mechanics of the encoding, and 2b. is a good fit for MBCS where concepts like overlong sequences don't exist. (And for GB 18030 you do have to walk a validity state machine, you can't just look at the lead byte.)

Unicode action item

150-A95 Markus Scherer: Review the feedback from Karl Williamson in L2/17-018 on best practices for UTF-8 overlongs, and make a recommendation to UTC.