L2/18-296

Freeze Bidi maximum explicit level at 125

Markus Scherer, 2018-sep-14

Proposal

In the Unicode Bidirectional Algorithm, the maximum explicit embedding level (“max_depth”) is currently 125. This is deemed “far more than sufficient for ordering, even with mechanically generated formatting”.

I propose that we guarantee that this value will not change.

I suggest that this could be done in the UBA spec and/or as a Stability Policy.

Details

Bidi implementations track nested sections enclosed in Bidi embedding and override controls via a stack where the depth corresponds to the current embedding level. It is useful to specify a maximum embedding level, a max_depth, so that implementations can use fixed-size arrays, and so that the behavior of different implementations is consistent.

The max_depth was originally 15 which is a stretch for human readers.
It was raised in 1999 to 61, to account for “mechanically generated formatting”.
It was raised in 2013 (Unicode 6.3) to 125, apparently because the introduction of isolates may cause increased nesting.

The original UTC decision for Unicode 6.3 was actually to raise it to 251, but I asked for a max_depth no higher than 125. A higher value would not have worked in the ICU API and would have required more extensive changes in the Java reference implementation.

It appears that there was no hard and fast requirement for a specific max_depth value.

Changing the value again would again create an inconsistency among implementations. Raising it above 125 would break some long-standing implementation practice.

References

http://www.unicode.org/reports/tr9/tr9-2.html (1998-10-30)

> The minimum embedding level of text is zero, and the maximum depth is level 15. (The reason for having a limitation is to provide a precise stack limit for implementations to guarantee the same results. Fifteen levels is far more than sufficient for ordering; the display becomes rather muddied with more than a small number of embeddings!)

http://www.unicode.org/reports/tr9/tr9-3.html (1999-02-08)

> The minimum embedding level of text is zero, and the maximum explicit depth is level 61. (The reason for having a limitation is to provide a precise stack limit for implementations to guarantee the same results. Sixty-one levels is far more than sufficient for ordering, even with mechanically generated formatting; the display becomes rather muddied with more than a small number of embeddings!)

http://www.unicode.org/reports/tr9/tr9-29.html#BD2 (Unicode 6.3, 2013-09-24)

> The minimum embedding level of text is zero, and the maximum explicit depth is 125, a value referred to as *max_depth* in the rest of this document. [...] The reason for having a limitation is to provide a precise stack limit for implementations to guarantee the same results. A total of 125 levels is far more than sufficient for ordering, even with mechanically generated formatting; the display becomes rather muddied with more than a small number of embeddings.

http://www.unicode.org/reports/tr9/#BD2 (current Unicode 11, 2018-05-09)

> The minimum embedding level of text is zero, and the maximum explicit depth is 125, a value referred to as *max_depth* in the rest of this document. [...] The reason for having a limitation is to provide a precise stack limit for implementations to guarantee the same results. A maximum explicit level of 125 is far more than sufficient for ordering, even with mechanically generated formatting; the display becomes rather muddied with more than a small number of embeddings.

https://www.unicode.org/policies/stability_policy.html Unicode Character Encoding Stability Policies

https://unicode-org.atlassian.net/browse/ICU-20125
ICU bug report about the corresponding API constant which is problematic if the value were to change again. Includes a comment about the 2013 reasoning for not exceeding 125.