Proposal for C2X

WG14 N 3145

Title: $ in Identifiers v2

Author, affiliation:

Robert C. Seacord, Woven Planet
rcseacord@gmail.com

Corentin Jabot
corentin.jabot@gmail.com

Steve Downey, Bloomberg, USA
<sdowney@gmail.com, sdowney2@bloomberg.net>

Peter Bindels, TomTom, Netherlands,
<dascandy@gmail.com>

Date: 2023-6-22

Proposal category: Defect

Target audience: Implementers

Abstract: Allow `$` as an implementation extension in identifiers v2

Prior art: C23

$ in Identifiers v2

Reply-to: Robert C. Seacord (rcseacord@gmail.com)

Document No: **N 3145**

Reference Document: N 3046, N2939, N2836**,** P1949R7 (http://wg21.link/p1949)

Date: 2022-3-02

This paper is to repair a potential defect introduced by accepting Alternative 1 of Proposal for C2X WG14 N 3046 which itself repaired a defect introduced by voting in N2836 Identifier Syntax using Unicode Standard Annex 31 into C23.

Change Log

2023-6-22
- Rebase on top of N3096 and remove options 2 and 3

2022-7-26:

- Initial version

1.0 PROBLEM DESCRIPTION

A question was raised at the July 2022 WG14 meeting concerning going back to the original identifier rules. The following straw poll was taken:

Straw poll: Does WG14 want to bring back the original identifier rules (e.g., allow `$` in identifiers as an extension, but not required to allow it)?

The results had clear consensus:

Results: 10 yes 2 no 8 abstain

Further discussion showed that the actual direction was less clear with the following opinions being noted:

- Each programming language can define its identifier syntax as relative to the Unicode identifier syntax, such as saying that identifiers are defined by the Unicode properties, with the addition of `$`.
- The original text allowed any implementation-defined characters, not just $

- I am strongly against what I'm suggesting but the "best" solution is to revert the "other implementation-defined characters" that got removed
- I would be much strongly opposed to something that would mention $ or any other specific character explicitly
- Allowing $ in identifiers would be a massive and unjustifiable land grab for both C and C++
- Would the following change suffice?

  6.4.2.1#1 add to `identifier-nondigit:`

  other implementation-defined characters

- Probably adding that sentence to both `identifier-start` and `identifier-continue`

As can be seen, opinions ranged from reverting to implementation-defined characters to keeping the current wording.

A quick survey of existing practice shows that current versions of gcc, clang, and icc all allow the `$` character anywhere in an identifier by default:

https://godbolt.org/z/frGzcTWoK

Only clang will diagnose the use of a `$` in an identifier, but only in `-pedantic` mode.

In both GCC and Clang, this is controlled by the `-f[no-]dollars-in-identifiers` flag which defaults to allow.

This paper proposes allowing `$` anywhere in identifiers as an implementation extension.

2.0 PROPOSED WORDING

Wording

Add the text in green in the N3096 working draft:

Subclause 6.4.2.1 paragraph 1

nondigit: one of

_ a b c d e f g h i j k l m

n o p q r s t u v w x y z

A B C D E F G H I J K L M

N O P Q R S T U V W X Y Z

Subclause 6.4.2.1 paragraph 2

An XID_Start character is an implementation-defined character whose corresponding code point ISO/IEC 10646 has the XID_Start property. An XID_Continue character is an implementation-defined character whose corresponding code point in ISO/IEC 10646 has the XID_Continue property. An identifier is a sequence of one identifier start character followed by 0 or more identifier continue characters, which designates one or more entities as described in 6.2.1. It is implementation-defined if a dollar sign $ may be used as a nondigit character. Lowercase and uppercase letters are distinct. There is no specific limit on the maximum length of an identifier

4.0 Acknowledgements

We would like to recognize the following people for their help with this work: Jens Maurer, Zach Laine, Tom Honermann, Corentin Jabot, and Aaron Ballman.

5.0 References

[AltId] Unicode Standard Annex.

http://www.unicode.org/reports/tr31/tr31-11.html#Alternative_Identifier_Syntax

[DefId] Unicode Standard Annex.

http://www.unicode.org/reports/tr31/tr31-11.html#Default_Identifier_Syntax

[N3146] Clark Nelson. 2010. Recommendations for extended identifier characters for C and C++.

https://wg21.link/n3146

[UAX15] Ken Whistler. Unicode Normalization Forms.

http://www.unicode.org/reports/tr15

[UAX31] Mark Davis. Unicode Identifier and Pattern Syntax.

http://www.unicode.org/reports/tr31

[UAX36] Mark Davis and Michel Suignard. Unicode Security Considerations.

http://www.unicode.org/reports/tr36

[UAX44] Ken Whistler and Laurențiu Iancu. Unicode Character Database.

http://www.unicode.org/reports/tr44

[UTS51] Mark Davis and Peter Edberg. Unicode Emoji.

http://www.unicode.org/reports/tr51