Proposal for C23

WG14 N 2939

Title: Identifier Syntax Fixes

Author, affiliation: Robert C. Seacord, Woven Planet
rcseacord@gmail.com

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

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

Corentin Jabot, France, <coretin.jabot@gmail.com>

Date: 2022-3-2

Proposal category: Defect

Target audience: Implementers

Abstract: Adopt Unicode Annex 31 as part of C23

Prior art: C23
Identifier Syntax Fixes

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

Document No: N 2939


Date: 2022-3-02

This paper is to repair a defect inadvertently introduced by voting N2836 Identifier Syntax using Unicode Standard Annex 31 into C23.

Change Log

2022-3-02:

• Initial version

1.0 PROBLEM DESCRIPTION

There are two known defects in N2836, which was voted into C23. The first more serious problem is that WG14 N2836 has changed the grammar of the "identifier" production in to read

identifier:

identifier-start

identifier identifier-continue

identifier-start:
	nondigit

universal-character-name of class XID_Start

identifier-continue:
	digit

nondigit
The previous grammar built identifiers from identifier-nondigit, which was:

identifier-nondigit:

nondigit
universal-character-name
other implementation-defined characters

The last option allowed an implementation to consider extended characters (e.g., Unicode characters other than ASCII) as valid in identifiers; the new grammar does not allow such freedom anymore. This change invalidates currently-valid C programs compiled with (popular) conforming extensions that accept e.g. UTF-8 encoded input employing identifiers in non-Western languages.

The second defect is that N2836 failed to update the text in Subclause 6.4.2.1. paragraph 2 and the paragraph is now inconsistent with the revised C23 text. N2731 working draft — October 18, 2021 has not yet been updated by N 2836, but the unchanged second paragraph reads as follows:

2 An identifier is a sequence of nondigit characters (including the underscore _, the lowercase and uppercase Latin letters, and other characters) and digits, which designates one or more entities as described in 6.2.1. Lowercase and uppercase letters are distinct. There is no specific limit on the maximum length of an identifier.

2.0 PROPOSED WORDING

Change in §6.4.2.1 paragraph 1:

identifier-start:

nondigit

XID_Start character

universal-character-name of class XID_Start

identifier-continue:

digit
nondigit

XID_Continue character
Replace §6.4.2.1 paragraph 2 with:

An XID_Start character is an implementation-defined character whose corresponding codepoint in ISO/IEC 10646 has the XID_Start property. An XID_Continue character is an implementation-defined character whose corresponding codepoint 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. 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, Corentin Jabot, Zach Laine, Tom Honermann, Jens Maurer, and Aaron Ballman.

5.0 References

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

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

https://wg21.link/n3146

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

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

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


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


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