Proposal for C23

WG14 N 2940

Title:	Removing trigraphs??!
Author, affiliation:	Robert C. Seacord, Woven Planet rcseacord@gmail.com
Date:	2022-3-2
Proposal category:	Defect
Target audience:	Implementers
Abstract:	Eliminate trigraphs
Prior art:	C++

Removing trigraphs??!

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

Document No: **N 2940**

Reference Document: WG14 N2701**,** WG21 N4086

Date: 2022-3-08

Change Log

2022-3-08:

· Initial version

1.0 PROBLEM DESCRIPTION

The adoption of *@ and $ in source and execution character set* [N2701] into C23 introduced a defect by introducing characters that are not defined in the Invariant Code Set as described in ISO/IEC 646 that lack trigraph sequences. There are two ways to repair this defect. The first is to add the missing trigraph sequences. The second is to remove trigraphs altogether. This paper proposes that trigraphs be removed entirely for the following reasons:

1. For C++ compatibility (C++ has eliminated trigraphs [WG21 N4086])
2. Trigraphs have limited utility.
3. Trigraphs, more often or not, are accidentally introduced and can result in defects and vulnerabilities [CERT].

1.1 Case study

The uses of trigraph-like constructs using https://searchcode.com/?q=%3F%3F%3D . We discovered:

- **22** instances of trigraphs being used deliberately in compiler test code.
- **3** uses were accidental
- **0** instances of trigraphs being deliberately used in production code.

Trigraphs continue to pose a burden on users of C.

1.3 Proposal

Trigraphs are handled in the first phase of translation:

> 1. Physical source file multibyte characters are mapped, in an implementation-defined manner, to the source character set (introducing new-line characters for end-of-line indicators) if necessary. Trigraph sequences are replaced by corresponding single-character internal representations.

Note that the mapping from physical source file characters to the basic source character set is implementation-defined. If trigraphs are removed from the language entirely, an implementation that wishes to support them can continue to do so: its implementation-defined mapping from physical source file characters to the basic source character set can include trigraph translation (and can even avoid doing so within raw string literals). **Trigraphs are not needed in the standard for backwards compatibility.**

2.0 PROPOSED WORDING

Modify Subclause 5.1.1.2, “Translation phases”, paragraph 1:

> 1. Physical source file multibyte characters are mapped, in an implementation-defined manner, to the source character set (introducing new-line characters for end-of-line indicators) if necessary. ~~Trigraph sequences are replaced by corresponding single-character internal representations.~~
>
> Forward references: universal character names (6.4.3), lexical elements (6.4), preprocessing directives (6.10), ~~trigraph sequences (5.2.1.1),~~ external definitions (6.9).

Delete Subclause 5.2.1.1, “Trigraph sequences”

4.0 Acknowledgements

We would like to recognize the following people for their help with this work: JF Bastien, Richard Smith, and Aaron Ballman.

5.0 References

[WG21 N4086] Richard Smith. Removing trigraphs??! 2014-06-18

[CERT] The CERT C Secure Coding Standard, PRE07-C. Avoid using repeated question marks