Proposal for C23
WG14 N 3106

Title:	Six versus eight-digit short identifiers
Author, affiliation:	Robert C. Seacord, Woven Planet
Date:	2023-2-13
Proposal category:	Defect
Target audience:	Implementers
Abstract:	Six versus eight-digit short identifiers for universal character names
Prior art:	C

Six versus eight-digit short identifiers

Reply-to: Robert C. Seacord (rcseacord@gmail.com)
Document No: **N 3106**
Reference Document: **N 3019**
Date: 2023-9-2

Change Log

2023-2-13:

- Initial version

1.0 Introduction and Rationale

NB comment GB-012 from [n3019] identifies the issue that the 2011 edition of ISO/IEC 10646 removed eight-digit short identifiers that were present in the 2003 edition (and this removal still applies as of the 2020 edition) but the current C23 draft supports eight-digit short identifiers but not six-digit short identifiers.

SC 22 N 5777, Subclause 6.4.3, “Universal character names” paragraph 4 states that:

> *The universal character name \Unnnnnnnn designates the character whose eight-digit short identifier (as specified by ISO/IEC 10646) is nnnnnnnn.80) Similarly, the universal character name \unnnn designates the character whose four-digit short identifier is nnnn (and whose eight-digit short identifier is 0000nnnn).*

Ideally, the C standard would only use short identifiers with no more than six digits. However, this would break backwards compatibility.

2.0 Proposed Solution

[n2785] proposed a new syntax `\u{}` usable in places where `\u` currently is. `\u{}` accepts an arbitrary number of hexadecimal digits. The values represented by this new syntax has the same requirements as the existing escape sequence, for example: `\u{nnnn}` must represent a valid Unicode scalar value. [n2785] is based on [P2290R3] which was adopted into C++23 https://github.com/cplusplus/papers/issues/983

The question should be "Can WG14 live with curly braces?" was polled at the 27 and 30 – 31 August, 1 – 3 September 2021 meeting [n2874].

Opinion Poll: Would WG14 be willing to accept using curly braces to delimit escape sequences as described in N2785?
17-0-5 Clear direction

Opinion Poll: Would WG14 want to adopt something along the lines of N2785 to be adopted into C23? 16-2-3 Clear direction

3.0 Wording

3.1 Wording Proposal #1

Replace 6.4.3, “Universal character names”, paragraph 4:

The universal character name \Unnnnnnnn designates the character whose eight-digit short identifier (as specified by ISO/IEC 10646) is nnnnnnnn. Similarly, the universal character name \unnnn designates the character whose four-digit short identifier is nnnn (and whose eight-digit short identifier is 0000nnnn).

with

A universal character name designates the character in ISO/IEC 10646 whose code point is the hexadecimal number represented by the sequence of hexadecimal digits in the universal character name.

[Editor’s note: Remove footnote 80]

Poll #1: Do we want to resolve GB-012 by applying the changes from n 3106 section 3.1?

3.2 Wording Proposal #2

Replace 6.4.3, “Universal character names”, paragraph 4:

universal-character-name:
\u *hex-quad*
\U *`0` `0` hexadecimal-digit hexadecimal-digit hex-quad hex-quad*

hex-quad:
hexadecimal-digit hexadecimal-digit hexadecimal-digit hexadecimal-digit

Modify Subclause 6.4.3, “Universal character names”, paragraph 2:

The universal character name \U00nnnnnnnn designates the character whose eight six-digit short identifier (as specified by ISO/IEC 10646) is nnnnnnnn. Similarly, the universal character name \u*nnnn* designates the character whose four-digit short identifier is nnnn (and whose eight six-digit short identifier is 00 00nnnn).

[Editor’s note: Remove footnote 80]

Poll #2: Do we want to resolve GB-012 by applying the changes from n 3106 section 3.2?

3.3 new syntax **\u{ }**

Modify Subclause 6.4.3, “Universal character names”, paragraph 2:

Syntax

hex-quad:

hexadecimal-digit hexadecimal-digit hexadecimal-digit hexadecimal-digit

simple-hexadecimal-digit-sequence:

hexadecimal-digit

simple-hexadecimal-digit-sequence hexadecimal-digit

universal-character-name:

\u *hex-quad*

\U *hex-quad hex-quad*

\u{ *simple-hexadecimal-digit-sequence* **}**

Poll #3: : Do we want to resolve GB-012 by applying the changes from n 3106 section 3.3?

(Requires wording proposal #1 be adopted).

4.0 Acknowledgements

I would like to recognize the following people for their help with this work: Corentin Jabot, Aaron Ballman, Steve Downey, Peter Bindels, Jens Gustedt, and Joseph Myers.

5.0 References

[n2785] Corentin Jabot, Aaron Ballman. Delimited escapes sequences. https://www.open-std.org/jtc1/sc22/WG14/www/docs/n2785.pdf

[n3019] Keaton, David. CD1 9899 ballot comments with progress from first week of ballot resolution.

[P2071R0] Tom Honermann and Peter Bindels. P2071R0: Named universal character escapes. https://wg21.link/p2071r0, 1 2020.

[P2290R3] Corentin Jabot. P2290R3: Delimited escape sequences. https://wg21.link/p2290r1, 6 2021.