

Mixed Wide String Literal Concatenation

JeanHeyd Meneide <phdofthehouse@gmail.com>

October 26th, 2020

- [1 Introduction & Motivation](#)
- [2 Wording](#)

Document: n2594

Previous Revisions: None

Audience: WG14

Proposal Category: New Features

Target Audience: General Developers, Compiler/Tooling Developers

Latest Revision: https://thephd.github.io/vendor/future_cxx/papers/source/n2594.html

Abstract:

This paper removes the ability to concatenate wide string literals (u, U, and L prefixed) together if they have a different prefix.

1 Introduction & Motivation

This paper is a compatibility-parity and query paper to resolve a Liason request from WG21 - Programming Languages, C++. It is almost identical to [p2201](#).

String concatenation involving string-literals with encoding-prefixes mixing L^{""}, u8^{""}, u^{""}, and U^{""} is currently conditionally-supported with implementation-defined behavior ([N2573 §6.4.5 String literals, Semantics, paragraph 5](#)).

None of [icc](#), [gcc](#), [clang](#), [MSVC](#) supports such mixed concatenations; all issue an error. Test code:

```
void f() {  
  
    { const void* a = L"" u""; }  
    { const void* a = L"" u8""; }  
    { const void* a = L"" U""; }  
  
    { const void* a = u8"" L""; }  
    { const void* a = u8"" u""; }  
    { const void* a = u8"" U""; }  
  
    { const void* a = u"" L""; }  
    { const void* a = u"" u8""; }  
    { const void* a = u"" U""; }  
  
    { const void* a = U"" L""; }  
    { const void* a = U"" u""; }  
    { const void* a = U"" u8""; }  
}
```

[SDCC, the Small Device C Compiler](#), does support such mixed concatenations, apparently taking the first encoding-prefix. One of its primary maintainers expressed sentiment that [the feature is not actually used much](#).

No meaningful use-case for such mixed concatenations is known, other than potential macro concatenation. However, no such usage experience was brought forth, and it is unlikely that code that uses multiple toolsets would be capable of taking advantage of such a feature since it is not present in many tools.

Therefore, this paper makes such mixed concatenations ill-formed.

2 Wording

The following wording is relative to [N2573](#).

Add the following sentence to §6.4.5 String Literals, Constraints

² A sequence of adjacent string literal tokens shall not include both a wide

string literal and a UTF-8 string literal. Adjacent wide string literal tokens shall have the same prefix.

Remove the following words §6.4.5 String literals, Semantics

⁵ In translation phase 6, the multibyte character sequences specified by any sequence of adjacent character and identically-prefixed string literal tokens are concatenated into a single multibyte character sequence. If any of the tokens has an encoding prefix, the resulting multibyte character sequence is treated as having the same prefix; otherwise, it is treated as a character string literal. ~~Whether differently prefixed wide string literal tokens can be concatenated and, if so, the treatment of the resulting multibyte character sequence are implementation defined.~~