

**Proposal for C2x**  
**WG14 N2265**

**Title:** Harmonizing `static_assert` with C++  
**Author, affiliation:** Aaron Ballman, GrammaTech  
**Date:** 2018-07-06  
**Proposal category:** New features  
**Target audience:** Developers working on combined C and C++ code bases

**Abstract:** C++17 changed the behavior of the `static_assert` declaration so that it no longer requires a string literal message. This proposal adds similar functionality to C.

**Prior art:** C++, Clang.

# Harmonizing `static_assert` with C++

Reply-to: Aaron Ballman (aaron@aaronballman.com)

Document No: N2265

Date: 2018-07-06

## Summary of Changes

N2265

- Editorial switch from “if one is supplied” to “if present”

N2197

- Original proposal

## Introduction and Rationale

C++17 adopted a new feature where the string literal message in a `static_assert` declaration is now optional [WG21 N3928]. This functionality was requested because the primary utility of a `static_assert` declaration is to terminate compilation when certain conditions do not hold, and sometimes the message is extraneous to the failing condition itself. Rather than asking the programmer to come up with a unique description of the failure condition, which frequently would involve repeating the condition itself, the message can be elided to receive an implementation-defined message instead. GCC, Clang, MSVC, and ICC all terminate with the same diagnostic when the string literal is elided in C++: “error: static assertion failed”.

This feature should be implementable by only modifying the core language wording and should not require any changes to the `static_assert` macro in `<assert.h>`.

Given that it is not uncommon to share header files between C and C++ code, and that `static_assert` declarations are used to guarantee specific conditions are met that may vary between implementations, harmonizing the syntax of this declaration with C++ would be useful. Clang already implements this feature in C as an extension.

## Proposed Wording

The wording proposed is a diff from the committee draft of ISO/IEC 9899-2017. **Green** text is new text, while **red** text is deleted text.

Modify 6.7.10p1:

*static\_assert-declaration:*

```
_Static_assert ( constant-expression , string-literal ) ;  
_Static_assert ( constant-expression ) ;
```

Modify 6.7.10.p3:

The constant expression shall be an integer constant expression. If the value of the constant expression compares unequal to 0, the declaration has no effect. Otherwise, the constraint is violated and the implementation shall produce a diagnostic message that includes the text of the string literal, **if present**, except that characters not in the basic source character set are not required to appear in the message.

## References

[WG21 N3928]

Extending static\_assert, v2. Walter Brown. <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3928.pdf>