

**Proposal for C2X
WG14 N2416**

Title: floating-point negation and conversion
Author, affiliation: C FP group
Date: 2019-08-24
Proposal category: Editorial
Reference: N2385 ISO/IEC 9899 working draft, June 2019

Problem:

Users might be surprised to discover that the converted value of negated floating constants can differ from the result of a library conversion function with corresponding input, even on the best implementations. For example, `-0.1` and `strtod("-0.1", (char**)NULL)` differ in some rounding direction modes, regardless of the evaluation method, when the implementation rounds correctly. The NOTE suggested below shows when this happens and a way for users to avoid the differences.

With the addition of the rounding control pragmas, the differences might be more noticed.

Suggested change:

To 6.4.4.2, after the Recommended Practice, append:

NOTE Floating constants do not include a sign and are negated by the unary `-` operator (6.5.3.3) which negates the rounded value of the constant. In contrast, the numeric conversion functions in the `strto` family (7.22.1.5, 7.22.1.6) include the sign as part of the input value and convert and round the negated input. Negating before rounding and negating after rounding might yield different results, depending on the rounding direction and whether the results are correctly rounded. For example, the results are the same when both are correctly rounded using rounding to nearest or rounding toward zero, but the results are different when they are inexact and correctly rounded using rounding toward positive infinity or rounding toward negative infinity.

Conversions yielding exact results require no rounding, so are not affected by the order of negating and rounding. For types with radix 10, decimal floating constants expressed within the precision and range of the evaluation format convert exactly. For types whose radix is a power of 2, hexadecimal floating constants expressed within the precision and range of the evaluation format convert exactly.