

Proposal for C2X WG14 N2407

Title: TS 18661-5abc supplementary attributes
Author: C FP group
Date: 2019-07-27
References: TS 18661-5, N2120, N2121, N2122
Proposal category: New features
Target audience: Performance-precision balancing, performance-reliability balancing, porting, debugging, reproducibility

This proposal incorporates the ISO/IEC TS 18661-5 pragmas to control evaluation methods, optimizations, and reproducibility into C2X. These pragmas provide attributes recommended in the IEC 60559:2008 floating-point standard, and in the updated IEEE 754-2019 standard. This proposal does not include the TS 18661-5 pragma for alternate exception handling.

The pragmas are intended to help with historically difficult problems that are often related to value changing optimizations and differences in evaluation methods.

1. *Porting floating-point code between platforms and tool sets, including debugging ported code.* Program development tools typically provide controls to manage optimizations and evaluation methods. These controls are implementation specific, both in syntax and semantics, and are often vaguely defined. It's difficult to impossible to map controls on one system to equivalent ones on another. Standard pragmas for evaluation methods and optimizations are intended to address this problem.
2. *Balancing performance against precision and reliability.* Current implementation-specific controls are usually compiler options that apply to the whole translation unit. However, many programs need aggressive optimizations only for relatively small performance-critical blocks. Applying the optimizations where they aren't needed unnecessarily risks floating-point anomalies throughout the entire program. Similarly, extra precision might be needed only in relatively small precision-critical blocks. Using extra precision throughout the program might unnecessarily degrade performance. The block-scope semantics of the pragmas address this problem.
3. *Obtaining reproducible results (on same or different platforms).* Some users want results that are the same on different platforms and that remain the same after tool set updates. Usually variations in floating-point results are harmless, but not always, and the cost to determine whether a difference is benign or the result of a serious bug can be great. Potential causes of differences in floating-point results are many and difficult for most programmers to avoid. The pragma for reproducible results is intended to help with this problem.

TS 18661-5a includes a pragma to set supported evaluation methods. 5a requires

support for the evaluation method characterized by **FLT_EVAL_METHOD** equal 0 (evaluation to type) for standard types. It requires support for the evaluation method characterized by **DEC_EVAL_METHOD** equal 1 if decimal types are supported. Most implementations already provide evaluation to type for standard types. 5a also specifies a user definable macro to have `tgmath` macros follow the evaluation method like operators do. This is to allow wide evaluation that is consistent for all floating-point operations.

5b specifies pragmas to allow common optimizations. The affected optimizations are well defined.

5c specifies a pragma to set the evaluation method, optimization, and other existing controls to facilitate reproducible results, and provides guidelines for code intended to be reproducible.

Notes:

A low-quality or initial implementation could have a conformance mode where only evaluate to type is supported, optimizations are disabled, and pragmas are ignored.

TS 18661-5abc could be recast as an annex to C2X.

5a and 5b are essentially independent of each other and of 5c. 5c depends on 5a and 5b, at least as currently written.