

Annex D means deprecated

Document #: P1702R0
Date: 2019-06-17
Project: Programming Language C++
Audience: LEWG
Reply-to: Corentin Jabot <corentin.jabot@gmail.com>

Proposed change

We propose that conforming implementations of the Standard be mandated to mark names and entities declared in annex D with the `[[deprecated]]` attribute.

Motivation

Deprecation is a powerful tool that should be used carefully with the express intent to remove deprecated features after a reasonable time has passed. Time that needs to be used by downstream projects to move away from APIs whose demise is foretold. As few people read the standard we rely on implementers to warn their users that an action is needed.

Because removing names is impactful, it cannot be done unless all C++ users are aware reasonably ahead of time (no less than one iteration of the standard) of an impending removal.

However, our ability to remove deprecated names holds only as long as our credibility that we will do so. C++ Users have indeed no reason to believe that names have been silently deprecated for over 2 decades will suddenly be removed.

If while reviewing this paper LEWG felt strongly that specific entities should not be marked `[[deprecated]]` it might be worth considering moving them out of annex D.

Not removing these names will also prevent us from ever reusing them.

A concern expressed at Kona was that marking symbols deprecated might break the build of people compiling with warnings as errors. However this already doesn't scale well as downstream projects end-up being affected by warning in upstream code (with the assumption that there are many more users and projects downstream), as such we do not think this usage warrants never marking anything `[[deprecated]]`.

Wording



Compatibility features

[depr]

This Clause describes features of the C++ Standard that are specified for compatibility with existing implementations.

These are deprecated features, where *deprecated* is defined as: Normative for the current edition of this International Standard, but having been identified as a candidate for removal from future revisions. An implementation **may** shall declare library names and entities described in this Clause with the `deprecated` attribute.

Acknowledgments

The author would like to thank Titus Winters for his valuable feedback.

References

[P0684] Titus Winters *C++ Stability, Velocity and Deployment* <https://wg21.link/P0684>