p1411r0 - Please reconsider <scope> for C++20

Peter Sommerlad

2019-01-21

Document Number:	p1411r0
Date:	2019-01-21
Project:	Programming Language C++
Audience:	LEWG
Target:	C++20

1 Introduction

Timing is hard. When LEWG blessed p0052 (scope guards and unique_resource) to advance (see https://issues.isocpp.org/show_bug.cgi?id=6), neither C++20 nor a library fundamentals TS was open and the paper was not ready for C++17. The current working draft was for C++17 and the library fundamentals TS 2 was (about to be) published. Thinking ahead LEWG voted p0052 for a future library fundamentals TS 3. I believe that went under LEWG's assumption that such a vehicle would open early enough so that the highly desired and long brewing feature could be included into C++20. However, library fundamentals TS 3 was only opened for business summer 2018 and thus too late to include new features as a staging area for C++20. While getting improved through LWG feedback under the assumption it could and should be included into C++20, it turned out, that such a formal blessing by LEWG is missing. In addition some minor design question arose (default constructability of unique_resource) that must be addressed by LEWG. Many people have expressed the desire to get p0052 into C++20, or at least its unique_resource part.

To not overwhelm LEWG with the wording and rationale of p0052 I would like to ask for answering 2-3 simple questions as soon as possible, so that I can be relieved of p0052 (which is reaching an age it has to go to elementary school). I understand that LEWG does not bless for inclusion in the working draft, but we need a formal forward decision by LEWG to not get bailed with that paper when it comes to plenary.

2 Decisions to be made

2.1 Decision 1: Allow default constructability of unique_resource

I have user requests that desire `unique_resource` to provide a default constructor if its resource and deleter types are default constructible. The reason is to ease using member variables and containers of such unique_resource objects. The default constructor would create the unique_resource in a released state to be later reassigned or reset. This would not introduced additional overhead, since the underlying infrastructure must already deal with such a released state.

LEWG question: Should unique_resource provide a default constructor creating a resource in a released state if both template argument types allow so?

2.2 Decision 2: Bless all of p0052 forward to LWG to consider for inclusion in C++20

LEWG question: Should p0052 be forwarded for C++20? (it was in spirit, but not in fact).

2.3 Decision 3 (only if 2 is NO): Bless unique_resource part of p0052 to LWG to consider for inclusion in C++20

This question is only relevant if Decision 2 is NO. There is stronger user desire for unique_resource than for scope guards.

LEWG question: Should only the unique_resource part of p0052 be forwarded to LWG for C++20?

2.4 Outlook

If neither decision 2 or 3 are positive I am eagerly looking for a future champion and co-author that takes this paper further. I have other priorities to work on and am burnt out by the many iterations this paper took.