

Doc. no.: N4803
Date: 2019-02-24
Reply to: Titus Winters (titus@google.com)
Audience: WG21

Kona 2019 LEWG Summary

This paper is a summary of the activities of the Library Evolution Working Group during the recent WG21 meeting in Kona.

In brief: we discussed 53 papers during the meeting (we did not meet after Saturday plenary). We believe that all library papers targeting C++20 were discussed. There are approximately 20 papers that may target C++23 or a TS that have been sent to LEWG, most of these have been discussed by LEWGI - very few papers in the pipeline have not been sent feedback at this point. Under the LEWGI banner we have begun to discuss some papers between meetings via weekly proposals on the isocpp-lib-ext reflector. Bryce Adelstein-Lelbach (SG18 / LEWGI chair) has volunteered to continue to run these reflector discussions between meetings.

Acknowledgements

Thank you to Jonathan Coe for note-taking almost the entirety of this meeting. Additional thanks to everyone else that was forced into service the rest of the time.

My most profound thanks to everyone that participated in the discussions during the week.

Commitments/Homework

A number of people volunteered to write follow-up papers or otherwise help make progress on some in-flight proposals during the week. The following is a listing of those promises.

- Titus Winters - Write a paper to update SD-8 per the discussion in P1250.
- Titus Winters - Write a paper to describe LEWG policy on explicit (single and multi-argument) c'tors
- TODO

Papers Forwarded to LWG for the Working Draft

Text Formatting

- P0645 Text Formatting
- P1361 Integration of chrono with text formatting

Ranges and Algorithms

- P1035 Input range adaptors

- P1456 Move-only views
- P1207 Movability of Single-pass Iterators
- P1223 `find_backward`
- P1391 Range constructor for `std::string_view`
- P1206 `ranges::to`: A function to convert any range to a container
- P1394 Range constructor for `std::span`

Modules

- P1502 Minimal standard library modules for C++20

Spaceship

- P0891 Make `strong_order` a Customization Point!
- P0790 Effect of `operator<=>` on the C++ Standard Library
- P1189 Adding `<=>` to library

New Types

- P1208 Adopt `source_location` from Library Fundamentals V3 for C++20
- P1222 A Standard `flatset`
- P1293 `ostream_joiner`
- P0660 Stop Tokens and a Joining Thread
- P0288 `unique_function`: a move-only `std::function`
 - May appear in some notes as P0228 as a result of a typo in the mailing
- P0448 A `stringstream` replacement using `span` as buffer
- P1132 `out_ptr` - a scalable output pointer abstraction

Misc

- P1328 Making `std::type_info::operator==` `constexpr`
- P0466 Layout-compatibility and Pointer-interconvertibility Traits
- P1355 Exposing a narrow contract for `ceil2`
- P1374 Resolving LWG #2307 for C++20: Consistently Explicit Constructors
- P1423 `char8_t` backward compatibility remediation
- P0798 Monadic operations for `std::optional`
- P1466 Miscellaneous minor fixes for `chrono`
- P1227 Signed `ssize()` functions, unsigned `size()` functions
- P0408 Efficient Access to `basic_stringbuf`'s Buffer
- P0553 Bit operations
- P1419 A SFINAE-friendly trait to determine the extent of statically sized containers

Forwarded for LFTS3

- P0843 `static_vector`
- P0052 Generic Scope Guard and RAII Wrapper for the Standard Library

Forwarded for SD-8

- P1339 Disallowing the friending of names in namespace std

Forwarded for C++Next

- P1393 A General Property Customization Mechanism
 - This means Executors is not C++20

Discussed but not Approved nor Forwarded

- P1072 basic_string::resize_default_init
- P1369 Guidelines for Formulating Library Semantics Specifications (we liked it, but doesn't require LEWG).
- P1453 Modularizing the Standard Library is a Reorganization Opportunity (decided on a direction)
- P1411 Please reconsider <scope> for C++20
- P1473 Shadow namespaces (concerned about experience and time before the IS)
- P1410 Remove deprecated stringstream (concerned about unnecessary impact on "working" code)
- P0813 construct() shall Return the Replaced Address (concerned about changing the API of Allocator, suggestion that this needs CWG resolution)
- P1496 Formatting of negative zero (not convinced this needs handling by us directly)