Doc. no.:	N4785
Date:	2018-11-13
Reply to:	Titus Winters (titus@google.com)
Audience:	WG21

San Diego 2018 LEWG Summary

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

In brief: we discussed 64 papers during the meeting, and 4 more informally on Saturday afternoon after plenary. There are approximately 5 papers that may target C++20 that have been sent to LEWG that have not been discussed or re-routed, and another 5-10 for TSes that will be published based on C++20. This is about 40% of what remained after the Rapperswil meeting. We have begun to discuss some papers between meetings via weekly proposals on the isocpp-lib-ext reflector - those discussions and the post-plenary discussions will form the set of “tentatively ready” papers in Kona next this year. Bryce Adelstein-Lelbach (SG18 / LEWGI chair) has volunteered to run these reflector discussions between meetings going forward.

Acknowledgements

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

Additional 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.

- Tony van Eerd - following up from P1035 - Go and talk to implementers about const-qualified assignment on pair/tuple as a stand-in for getting a language-level solution to container-of-references vs. reference-to-container.
- Victor Zverovich - Investigate extensions via ADL / inline-friend functions. Talk to Eric Niebler and/or Eric Fiselier.
- Tony van Eerd, Titus Winters - Write a paper about LEWG policy on explicit c’tors.
- Titus Winters - Write a paper to update SD-8 per the discussion in P1250.

Papers Forwarded to LWG for the Working Draft

- P1085 Should Span Be Regular
- P1024 Usability Enhancements for std::span

- P0784 Standard containers and constexpr
- P0533 constexpr for <cmath> and <cstdlib>
- P1251 A more constexpr bitset
- P0595 std::is_constant_evaluated()
- P0340 Making std::underlying_type SFINAE-friendly
- P0920 Precalculated hash values in lookup
- P0627 Function to mark unreachable code
- P1165 Fixing allocator usage for operator+(basic_string)
- P1210 Completing the Rebase of Library Fundamentals, Version 3, Working Draft
- P0325 to_array from LFTS with updates
- P1083 Move resource_adaptor from Library TS to the C++ WP
- P1209 Adopt Consistent Container Erasure from Library Fundamentals 2 for C++20
- P1187 A type trait for std::compare_3way()'s type
- P1191 Adding operator<=> to types that are not currently comparable
- P1248 Fixing Relations
- P1295 Spaceship library update
- P1252 Ranges Design Cleanup
- P1243 Rangify New Algorithms
- P0645 Text Formatting
- P0881 A Proposal to add stack trace library
- P0883 Fixing Atomic Initialization
- P1233 Shift-by-negative in shift_left and shift_right
- P0549 Adjuncts to std::hash
- P1272 Byteswapping for fun&&nuf
- P0631 Math Constants
- P1280 Integer Width Literals
- P1186 When do you actually use <=>?
- P1135 The C++20 Synchronization Library
- P0201 polymorphic_value
- P1227 Signed size() functions

Papers/Topics that will be prioritized for C++20

These are topics that LEWG believes are worth focusing on in the Kona meeting with the possible aim of getting them into the C++20 IS. Presence on this list is no guarantee that these papers will pass, but is notice to the community that we are going to focus on these areas (whether in these specific papers or not).

- P1208 Adopt source_location from Library Fundamentals V3 for C++20
- P1293 ostream_joiner
- P1035 Input range adaptors
- P1206 Range constructors for standard containers and views
- P0798 Monadic operations for std::optional
- P0443 Unified Executors

- P0437 Numeric Traits for the Next Standard Library
- P0586 Safe integral comparisons
- P1222 A Standard flat_set
- P0660 A Cooperatively Interruptible Joining Thread
- P1072 Optimized Initialization for basic_string and vector

Discussed but not Approved nor Forwarded

- P1259 Merge most of Networking TS into C++ Working Draft
- P0863 Fixing the partial_order comparison algorithm
- P1312 Comparison Concepts
- P1291 std::ranges::less<> Should Be More!
- P1255 A view of 0 or 1 elements: view::maybe
- P0943 Support C atomics in C++
- P1175 a simple and practical optional reference for C++
- P0891 Let strong_order Truly Be a Customization Point!
- P0657 - Deprecate certain declarations in the global namespace
- P1182 - New names for the power-of-2 templates
- P0310 - Splitting node and array allocation in allocators
- P1163 Explicitly Implicifying explicit Constructors
- P0262 A Class for Status and Optional Value