Doc. no.: N4754
Date: 2018-06-12
Reply to: Titus Winters (titus@google.com)
Audience: WG21

Rapperswil 2018 LEWG Summary

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

In brief: we discussed 44 papers during the meeting, and 5 more informally on Saturday afternoon after plenary. There are approximately 23 papers that have been sent to LEWG that have not been discussed or re-routed - this is roughly the same volume as what remained after the Jacksonville 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 San Diego later this year.

Acknowledgements

Thank you to Michal Sudwoj for note-taker roughly half of this meeting. Additional thanks to everyone else that was forced into service when Michal was unavailable..

Additional thanks to everyone that participated in the discussions during the week.

Approved LEWG Policy Changes

- Produce / maintain a standing document detailing the types of changes we may make in the future. (P0921r1)
- Compile-time libraries and metafunctions should go to SG7 first, then LEWG.
- Text-related things (formatting, splitting, etc) should be seen by SG16 to ensure that they are future-compatible with SG16 direction.
- We want to encourage new library designs to lvalue-ref qualify methods that return a reference to *this, or iterator with similar lifetime behavior. (That is, begin(), end(), operator=, etc should consider whether to forbid invocation on temporaries.)

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 - Write a paper to remove all algorithms recently added to the WD in favor of the Ranges forms of those algorithms. Also double-check constexpr additions for Ranges and legacy algorithms.

- Marc Mutz (assisted by Casey Carter and David Stone) - Write a paper to re-introduce the relevant “3-legged algorithms” (notably, std::equals / std::ranges::equals) for Ranges usage. Provide summary of why most of these are not necessary with Ranges.
- Corentin Jabot - Write a paper to introduce move-only iterators under the Ranges iterator paradigm. Andrew Hunter had previously proposed this in P0902 - the changes in iterator definitions in Ranges is a good time to do this cleanly.
- Jorg Brown - Investigate std::ssize() and/or adding ::ssize() members to containers as an alternate path to resolving unsigned-size designs.
- Titus Winters - Update the newly produced SD based on P0921 to mention that we reserve the right to add new member tags to types in namespace std. Also, discuss preferred extension mechanisms: should we be inspecting tags in user-defined types? Or providing clear guidance on std extension points? Are we concerned with defining meaning/semantics of previously unspecified tags in user-defined types?
- Tony van Eerd - Provide detailed feedback to the author of P0959 and help revise the paper.

Papers Forwarded to LWG for the Working Draft

- Merging the Ranges TS (P0896r1)
- Better, Safer Range Access Customization Points (P0970r1)
- Contiguous Ranges (P0944r0)
- Deep Integration of the Ranges TS (P1037r0)
- Rangify the uninitialised memory algorithms! (P1033r0)
- constexpr comparison operators for std::array (P1023r0)
- Making std::vector constexpr (P1004r0)
- Constexpr in std::pointer_traits (P1006r0)
- Tightening the constraints on std::function (P0932r0)
- Well-behaved interpolation for numbers and pointers (P0811r2)
- Utility functions to implement uses-allocator construction (P0591r3)
- Allocator-aware basic stringbuf (P0407r2)
- Unwrapping reference_wrapper (P0318r1)
- Target Vectorization Policies from Parallelism V2 TS to C++20 (P1001r1)
- Usability Enhancements for std::span (P1024r0)
- zero(), min(), and max() should be noexcept (P0972r0)
- Smart pointer creation with default initialization (P1020r0)
- Misc constexpr bits (P1032r0)
- function_ref: a non-owning reference to a Callable (P0792r2)
- Add coroutine task type (P1056r0)
- Improving atomic flag (P0955r0)
- fixed_capacity_vector (P0843r1)

Papers Forwarded to LWG for Parallelism TS2

- Finding the right set of traits for simd (P0964r1)
- Feedback on P0214 (simd) (P0820r3)
- Hazard pointers (P0566r5)

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

These are topics that LEWG believes are worth focusing on in the San Diego and Kona meetings with the 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).

- Text Formatting (P0645r2)
- A Proposal to add stack trace library (P0881r1)
- Generic none() factories for Nullable types (P0196r5)
- Monadic operations for std::optional (P0798r0)

Discussed but not Approved nor Forwarded

- A Unified Executors Proposal for C++ (P0443r7)
- A Modest Executor Proposal (P1055r0)
- Define basic_string_view(nullptr) (P0903r1)
- Narrow contracts in string_view versus P0903 (P1043r0)
- A friendlier tuple get (P0825r1)
- Fractional Numeric Type (P1050r0)
- std::embed (P1040r0)
- Fixing the partial_order comparison algorithm (P0863r0)
- Fixed-Point Real Numbers (P0037r5)
- Adding support for type-based metaprogramming to the standard library (P0949r0)
- User-defined Literals for std::filesystem::path (P0882)
- A Proposal to Add split/join of string/string_view to the Standard Library (P0540r1)
- regex_iterator should be iterable (P0757r0)
- A Proposal for a UUID Library (P0959r0)
- Input range adaptors (P1035r0)
- A call for a Data Persistence (iostream v2) study group (P1026r0)
- Sizes Should Only span Unsigned (P1089r0)
- Should Span Be Regular (P1085r0)
- Safe integral comparisons (P0586r0)
- Zero-overhead deterministic exceptions: Throwing values (just the library parts in 4.2/4.3) (P0709r0)

Discussed and Informally Approved after Plenary

I expect to open brief discussion during LEWG early in the week at the next meeting.

- std::assume_aligned (P1007r0)

Discussed Informally after Plenary

- Revised Latches and Barriers for C++20 (P0666r2)
- String View Conversion for Function Arguments (P0994r0)
- Add c_array() member function to std::array (P0635r0)
- Constant Pointer View - std::as_const Strikes Back! -- presented by ADAM David Alan Martin (P1011r0)