

# P2300

## TOWARDS REVISION 2

Michael Garland, August 2021

## AUTHORS

Michał Dominiak,

Lewis Baker,

Lee Howes,

Kirk Shoop,

Michael Garland,

Eric Niebler &

Bryce Adelstein Lelbach

# GOALS OF THIS PAPER

Produce a single, usable design for controlling execution in C++

Refine design of core features in P0443 based on feedback from LEWG & SG1

Consolidate selected features needed to write effective code, especially from:

- P2181r1 *Correcting the design of bulk execution*
- P1897r3 *Towards C++23 executors: A proposal for an initial set of algorithms*
- P2175r0 *Composable cancellation for sender-based async operations*

Produce a self-contained, well-documented design suitable for inclusion in C++23

# SELECTED PLANS FOR REVISION 2

Responding to feedback received in these discussions so far

How does P2300 integrate with co-routines?

We will provide additional material explaining this along with some motivating examples.

How much algorithm customization is typically required when writing schedulers?

We will include an example scheduler that demonstrate what needs to be done and summarize our own implementation experience.

# SELECTED PLANS FOR REVISION 2

Responding to feedback received in these discussions so far

What progress semantics are allowed for bulk?

We will expand the discussion in the Design section to explain what semantics are permitted.

How do we envision this functionality composing with parallel algorithms taking execution policies?

We will expand on the future direction that was only briefly sketched in our initial presentation.

# SELECTED PLANS FOR REVISION 2

Responding to feedback received in these discussions so far

What conventions should be used for calling `set_error` vs. `set_value`?

The decision should correspond exactly with the choice one would make to either throw an exception or return a value from a traditional (synchronous) interface. We will add material explaining this correspondence and highlighting expected patterns of use.

Does this do everything properties in 443 could do?

The design of P2300 leaves room for all such functionality to be covered in a separate paper. We will summarize how this could be done in case there is interest in pursuing that.

# ADDITIONAL PLANS

Various other bugs and requests

Visible on our issue tracker:

[https://github.com/brycelelbach/wg21\\_p2300\\_std\\_execution/issues](https://github.com/brycelelbach/wg21_p2300_std_execution/issues)

**Questions or Comments?**