LIBRARY EVOLUTION LEADERSHIP'S UNDERSTANDING OF THE NOEXCEPT POLICY HISTORY

Bryce Adelstein Lelbach
Nevin Liber
Robert Leahy
Ben Craig
Fabio Fracassi
Guy Davidson

2011 Madrid

- N3279 Conservative use of noexcept in the Library by *Alisdair Meredith* & *John Lakos*.
  - *Narrow contracts and wide contracts.*

  - No library destructor should throw.
    - *Note: there is library wording in the standard for this.*
  - Each library function having a wide contract, that the LWG agree cannot throw, should be marked as unconditionally noexcept.
  - If a library swap function, move-constructor, or move-assignment operator is conditionally-wide (i.e. can be proven to not throw by applying the noexcept operator) then it should be marked as conditionally noexcept. No other function should use a conditional noexcept specification.
  - Library functions designed for compatibility with "C" code (such as the atomics facility), may be marked as unconditionally noexcept.

2011 Madrid

- N3274 Minutes of WG21 Meeting, March 21, 2011
  - Motion 8 Move we apply the proposed resolution from N3279, Conservative use of noexcept in the library, to the C++0X Working Paper. Note that this reverts many applications of 'noexcept' at the last meeting. A 'Throws: Nothing' clause was restored only in the cases where that guarantee was in the pre-noexcept wording. This means some function contracts may have changed since the previous WP, such as std::align, by reverting to the contract in the FCD. This is only because there are no ballot comments requesting such changes, and 'Throws : Nothing' clauses are likely additions to the next TC.
    - PL22.16: 23-0-0
    - WG21: 8-0-0

2011 Madrid

- Subsequently, the committee believed this vote established the Lakos Rule.
  - Can be relaxed in specific circumstances with justification.
    - E.g.: dereferencing smart pointers is a narrow contract marked noexcept.
  - Implementors can add noexcept at their discretion.

2018 Jacksonville

- P0884 Extending the noexcept Policy, Rev0 by *Nicolai Josuttis* proposed that wrapper types should be conditionally noexcept.
  - If a library type has wrapping semantics to transparently provide the same behavior as the underlying type, then default constructor, copy constructor, and copy-assignment operator should be marked as conditionally noexcept the underlying exception specification still holds.
  - We grant ourselves that we may mark operations conditionally noexcept for “wrapper types” - 5-14-1-0-0
  - *Intended to be sent to plenary for final approval.*

2020 Prague

- P1656 "Throws: Nothing" should be noexcept by *Agustín Bergé* proposed overturning the Lakos Rule.
  - Many stakeholders present for LEWG discussion.
  - Adopt Proposed Policy: Minimal - 6-14-0-4-1
  - Adopt Proposed Policy: Untangled from Contracts - 9-9-0-5-2
  - Adopt Proposed Policy: Untangled from Contracts, keeping C library bit - 8-10-2-5-0
  - *Intended to be sent to plenary for final approval.*

2023 Varna

- Lakos Rule was scheduled for discussion in Library Evolution.
- One of the relevant stakeholders indicated a desire to defer the discussion until a later time.
  - P2837 Planning to Revisit the Lakos Rule by *Alisdair Meredith & Harold Bott Jr.*
- In light of that information and subsequent discussion, Library Evolution leadership agreed that it should be deferred and stated it would be removed from the schedule.

2023 Varna

- LEWG held an evening session to discuss how Library Evolution should establish and maintain policy.
- “No need to present. We'll be talking more about policy in general. We may use the narrow noexcept discussion as an example, but we won't litigate it.”

2023 Varna Tuesday Evening Session

2023 Varna

- The Convener informed us
  - Note what was adopted [in 2011 Madrid Motion 8] was the proposed resolution of specific library wording diffs to be applied to the draft standard.
  - We've never taken a plenary poll about any EWG or LEWG design policy.

Current LEWG noexcept Policy

How to move forward

- In Kona 2023, we will debate noexcept policy and determine if we have consensus.
- Sufficient notice will be provided for such a discussion.
- Stakeholders should agree that we are ready for the discussion.