Rename `async_scope_token`

Document Number: P3685R0
Date: 2025-05-07
Reply-to: Robert Leahy <rleahy@rleahy.ca>
Audience: LEWG

Abstract

This paper proposes renaming the `std::execution::async_scope_token` concept to `std::execution::scope_token`.

Background

P3149 [1] has been considered and forwarded from LEWG to LWG. It adds the following entities to the `std::execution` namespace:

- `async_scope_token`
- `nest`
- `spawn`
- `spawn_future`
- `simple_counting_scope`
- `counting_scope`

Discussion

Consistency With P2300

P2300 [2] added the `std::execution` namespace. The introduction of that paper (id. at §1) describes the purpose thereof as:

“[...] propos[ing] a self-contained design for a Standard C++ framework for managing asynchronous execution [...]”

Despite this explicit attestation that the purpose of the paper is “asynchronous execution” none of the entities added thereby have a name with an `async_` prefix. The paper does use the aforementioned prefix in several examples ([2] at §1.3.2, §1.3.3, and §1.4) but proposes none of those names.

The paper introduces a new word of power: “Asynchronous operation” ([2] at §34.3). Moreover it is directly stated (ibid.):

“An asynchronous operation has associated state known as its operation state*.”*

Note that this is not called an “asynchronous operation state.” Note also the names of the related C++ entities in the `std::execution` namespace:

- `operation_state_t` (not `async_operation_state_t`) ([2] at §34.4)
- `operation_state` (note `async_operation_state`) (id. at §34.8)

Note also that the nested type alias which is required of operation state is `operation_state_concept`, not `async_operation_state_concept`.

From the above we can describe the status quo: Entities in `std::execution` are presumptively asynchronous and do not attest to their own asynchronicity via an `async_` prefix. This is the case even when they are referred to as “asynchronous” in plain language description.

Consistency Within P3149

P3149 adds only one entity to `std::execution` with an `async_` prefix: `async_scope_token`, a concept.

This can be parsed in one of two ways:

- The scope is asynchronous (i.e. bracketed as (`async_scope`)`_token`), or
- The token (rather than the scope) is asynchronous (i.e. bracketed as `async_`(`scope_token`))

Inspection of the token’s interface reveals it adds no asynchronicity not already endemic to the associated scope, and therefore the latter interpretation can be disregarded. Moreover the paper makes repeated common language references to the idea of an “async scope” (e.g. [1] at §2.1: “The general concept of an **async scope** to manage work [...]” (emphasis added)) which supports this bracketing.

Despite the above the paper neglects to use the following names:

- `simple_counting_async_scope`
- `counting_async_scope`

Revealing a preference for using `scope` rather than `async_scope` when naming C++ entities.

Moreover the wording for `async_scope_token` ([1] at §8.7) proposes adding a section entitled “[s]cope concepts” (not “async scope concepts”) and proposes a stable name therefor of “exec.scope.concepts” (not “exec.async_scope.concepts”).

Proposal

Rename `std::execution::async_scope_token` to `std::execution::scope_token`.

References

[1] I. Petersen et al. `async_scope` – Creating scopes for non-sequential concurrency P3149R9
[2] M. Dominiak et al. std::execution P2300R10