

Doc No: P0413r0
Date: 2016-07-07
Audience: LEWG (and SG1)
Authors: Pablo Halpern, Intel Corp.
phalpern@halpernwrightsoftware.com

Updating Parallel Execution Policy Names in the Parallelism TS

Contents

1	Summary	1
2	Consistency issues not addressed by this paper.....	1
2.1	Versioning namespaces	1
2.2	The parallel namespace.....	2
2.3	Initializers for execution policy objects	2
3	Formal Wording	2
4	References	4

1 Summary

The C++17 Working Draft and CD contain changes to parallel execution policy names adopted from [P0336r1](#) in June 2016 in Oulu. When I wrote P0336, I assumed that those parts of the Parallelism TS that have been adopted into the C++17 WD would be removed from the TS. Unfortunately, ISO rules say that a TS must be based on either an IS or DIS, not a CD. Thus, the duplicate material must remain in the TS WD until C++17 reaches the DIS stage. This means that the changes described in P0336 must also be adopted into the Parallelism TS. This paper proposes exactly that.

The changes proposed here are targeted for the Parallelism TS Version 2.

2 Consistency issues not addressed by this paper

This paper addresses only the changes from P0336 that are needed to make the Parallelism TS consistent with the most recent C++17 working draft. However, at least three other consistency issues remain, one relating to consistency with Version 1 of the parallelism TS, and two relating to consistency with C++17. I mention these issues here because they will eventually need to be addressed and because either or both are likely to be raised during discussion of this paper.

2.1 Versioning namespaces

The current TS changes the inline version namespace from `v1` to `v2`. Since this paper proposes changes the names of existing identifiers, it may be desirable to put aliases for these identifiers, using the old names, in the `v1` namespace. Whether or not to do this, and the best way to handle this kind of versioning issue is a larger LEWG discussion.

2.2 The parallel namespace

When parts of the parallelism TS were adopted, the `parallel` namespace (nested within `experimental`) was dropped. It is not clear what should be done with this namespace in the parallelism TS. One possibility is to make it `inline`, so that new code can choose to not to use it but old code would continue to compile. Another possibility would be to combine it with the version namespace to create `parallel_v2`, which would allow implementations to alias it with `using namespace parallel = parallel_v2`.

2.3 Initializers for execution policy objects

The TS working draft shows empty initializers for the execution policy objects. The C++17 working draft and CD shows *unspecified* in the initializers. The TS should probably be updated to match the CD.

3 Formal Wording

All section names and numbers are relative to the **March 2016 working draft of the Parallelism TS, N4578**.

In section 1.5 [parallel.general.features], update the `__cpp_lib_experimental_parallel_algorithm` feature:

Table 1 – Feature Test Macro(s)

Name	Value	Header
<code>__cpp_lib_experimental_parallel_algorithm</code>	201505 201606	<code><experimental/algorithm></code> <code><experimental/exception_list></code> <code><experimental/execution_policy></code> <code><experimental/numeric></code>
<code>__cpp_lib_experimental_parallel_task_block</code>	201510	<code><experimental/task_block></code>

In section 2.2 [execpol.syn], rename the `<execution_policy>` header, add an execution namespace, and rename the execution policies:

20.18.2 Header `<execution_policy>` synopsis [parallel.execution_policy.synopsis]

```
namespace std {
namespace experimental {
namespace parallel {
inline namespace v2 {
    // 2.3, execution policy type trait:
    template<class T> struct is_execution_policy;
    template<class T>
        constexpr bool is_execution_policy_v = is_execution_policy<T>::value;

    namespace execution {

        // 2.4, sequential/sequenced execution policy:
        class sequential_executionsequenced_policy;
```

```

// 2.5, parallel execution policy:
class parallel_execution_policy;

// 2.6, parallel+vectorunsequenced execution policy:
class parallel_vector_executionunsequenced_policy;

// 2.7, Dynamic execution policy

// 20.18.7, execution policy objects:
constexpr sequenced_policy seq{};
constexpr parallel_policy par{};
constexpr parallel_unsequenced_policy par_unseq{};

```

The above three definitions constitute a “drive by” fix in that these definitions were missing from the synopsis of the TS working draft.

```

}
}
}
}
}

```

In section 2.8, rename the execution policy objects:

2.8 Execution policy objects [parallel.execpol.objects]

```

constexpr sequential_executionsequenced_policy seq{};
constexpr parallel_execution_policy par{};
constexpr parallel_vector_executionunsequenced_policy par_vecunseq{};

```

The header `<experimental/execution_policy>` declares a global object associated with each type of execution policy defined by this Technical Specification.

Throughout the remainder of the TS WD, perform the following replacements:

Replace occurrences of:	with:
<code><experimental/execution_policy></code>	<code><experimental/execution></code>
<code>sequential_execution_policy</code>	<code>execution::sequenced_policy</code>
<code>parallel_execution_policy</code>	<code>execution::parallel_policy</code>
<code>parallel_vector_execution_policy</code>	<code>execution::parallel_unsequenced_policy</code>
<code>seq</code> (when referring to the policy token)	<code>execution::seq</code>
<code>par</code>	<code>execution::par</code>
<code>par_vec</code>	<code>execution::par_unseq</code>

4 References

[P4578](#) *Working Draft, Technical Specification for C++ Extensions for Parallelism Version 2*, Jared Hoberock, editor, 2016-02-22

[P0336r1](#) *Better Names for Parallel Execution Policies in C++17*, Pablo Halpern, 2016-06-25