

# The functionality deprecated in TBB 2020 and removed in oneTBB

---

## 1 Contents

2	Motivation.....	1
3	The functionality deprecated in TBB 2020 and removed in oneTBB .....	1
3.1	Pre C++11 compatibility API .....	1
3.2	PPL compatibility API.....	2
3.3	Other functionality .....	2
4	The functionality considered for reworking in oneTBB.....	3
5	Contacts.....	4

## 2 Motivation

Our goal is to improve Threading Building Blocks (TBB) and update it in accordance with the latest C++ standards to increase usability. To do this, we are reevaluating TBB functionality. Features under consideration are mapped to updated options as described below.

This document describes functionality deprecated in TBB 2020 and removed in oneAPI Threading Building Blocks (oneTBB).

## 3 The functionality deprecated in TBB 2020 and removed in oneTBB

The following subsections list the affected functionality, in addition to already mentioned functionality in TBB Developer Reference (Appendices->Compatibility Features).

### 3.1 Pre C++11 compatibility API

We decided to drop C++03/98 support as well as functionality that is obsolete with C++11.

The following table summarizes the TBB functionality that can be directly replaced with C++11.

Deprecated/removed TBB functionality	Replacement
tbb::atomic	std::atomic
tbb::flow::tuple (incl. helper classes)	std::tuple
tbb::mutex	std::mutex
tbb::recursive_mutex	std::recursive_mutex

tbb::critical_section (incl. tbb::improper_lock)	std::mutex
tbb::hash (incl. tbb::hasher)	std::hash
tbb::tbb_thread / std::thread / std::this_thread	std::thread with possible minimal changes related to std::chrono
std::lock_guard / std::unique_lock (incl. helper classes)	Minimal changes related to std::chrono might be required
std::condition_variable (incl. std::cv_status, std::timeout, std::no_timeout)	Minimal changes related to std::chrono might be required
tbb::aligned_space	std::aligned_storage
tbb::tbb_exception / tbb::captured_exception / tbb::movable_exception	No more needed due to TBB exact exception propagation

### 3.2 PPL compatibility API

The following functionality was provided for compatibility with Microsoft\* Parallel Patterns Library (PPL).

Deprecated/removed TBB functionality	Replacement
Concurrency::critical_section	std::mutex
Concurrency::reader_writer_lock (incl. Concurrency::improper_lock)	std::shared_mutex (*It will be provided by TBB in pre-C++17 environments)
Concurrency::parallel_invoke	tbb::parallel_invoke
Concurrency::parallel_for (first, last, f)	tbb::parallel_for (first, last, f)
Concurrency::parallel_for_each	tbb::parallel_for_each
Concurrency::task_group (incl. helper classes)	tbb::task_group
Concurrency::structured_task_group (incl. helper classes)	tbb::task_group

### 3.3 Other functionality

The following table summarizes the TBB functionality that significantly duplicates other existing functionality or has little practical usage.

Deprecated/removed TBB functionality	Replacement
Task API (tbb::task, tbb::empty_task, tbb::task_list and related functions)	No direct replacement, the majority use cases can be covered with tbb::task_group, tbb::flow::graph. Task priorities can be covered with Flow graph node priorities and static arena-level priorities.
tbb::task_scheduler_init	tbb::task_arena, tbb::global_control (it will be extended to support the blocking terminate preview functionality)
tbb::pipeline (incl. tbb::filter, tbb::thread_bound_filter)	tbb::parallel_pipeline, tbb::flow::async_node, resumable tasks
tbb::flow::sender / tbb::flow::receiver / tbb::flow::continue_receiver	Remain as unspecified base types for flow graph classes
Allocator template parameter for the flow graph nodes	No replacement is planned
tbb::flow_async_msg, tbb::flow::streaming_node, tbb::flow::opencl_node	No replacement is planned. To interact with asynchronous/heterogeneous activity use tbb::flow::async_node or resumable tasks
(preview) tbb::serial::parallel_for	Limit the number of threads to 1 with task_arena or global_control
(preview) runtime_loader (aka tbbproxy library)	No replacement is planned
tbb::structured_task_group (incl. helper classes)	tbb::task_group
tbb::parallel_do	tbb::parallel_for_each
tbb::flow::source_node	tbb::flow::input_node

#### 4 The functionality considered for reworking in oneTBB

The following list contains the functionality considered for reworking. It might be removed, reworked or left “as is” depending on evaluation feedback.

Removed TBB functionality	Replacement
tbb::reader_writer_lock	std::shared_mutex  (*Will be implemented in oneTBB in pre-C++17 environments)

## 5 Contacts

If you have any questions or concerns, please email [IntelTBBDevelopers@intel.com](mailto:IntelTBBDevelopers@intel.com).