OpenMP 5.0 API Reference Guide: Tasking

The OpenMP® API gives parallel programmers a simple and flexible interface for developing portable, scalable parallel applications in C/C++ and Fortran. The OpenMP tasking features are suitable for complex applications that require to parallelize irregular algorithms. OpenMP tasks are a modern way of expressing concurrence and parallelism.

Functionality new/changed in OpenMP 5.0 is in this color, and in OpenMP 4.5 is in this color. *Deprecated* in the 5.0 spec.

[n.n.n] Sections in the 5.0 spec.  [n.n.n] Sections in the 4.5 spec.

## Directives and Constructs

An OpenMP executable directive applies to the succeeding structured block. A structured-block is an OpenMP construct or a block of executable statements with a single entry at the top and a single exit at the bottom. OpenMP directives except SIMD and declare target directives may not appear in PURE or ELEMENTAL procedures.

### Team management constructs

**parallel** [2.3.6] [2.5.3]

Forms a team of threads and starts parallel execution.

```
#pragma omp parallel [clause[ , clause] ...]

|$omp parallel [clause[ , clause] ...]
```

### Single tasking constructs

**single** [2.3.2] [2.3.3]

Specifies that the associated structured block is executed by only one of the threads in the team.

```
#pragma omp single [clause[ , clause] ...]

|$omp single [clause[ , clause] ...]
```

### Tasking constructs

**task** [2.10.1] [2.9.1]

Defines an explicit task. The data environment of the task is limited to the data environment of the thread of the team.

```
#pragma omp task [clause[ , clause] ...]

|$omp task [clause[ , clause] ...]
```

### Synchronization constructs

**taskwait** [2.10.5] [2.13.1]

Specifies a wait on the completion of child tasks of the current task.

```
#pragma omp taskwait [clause[ , clause] ...]

|$omp taskwait [clause[ , clause] ...]
```

**taskgroup** [2.17.6] [2.13.3]

Specifies a wait on the completion of child tasks of the current task, and waits for descendant tasks.

```
#pragma omp taskgroup [clause[ , clause] ...]

|$omp taskgroup [clause[ , clause] ...]
```

### Dependent objects

**depobj construct**

```
#pragma omp depobj [depobj-clause]

|$omp depobj [depobj-clause]
```

### Cancellation constructs

**cancel** [2.18.1] [2.14.1]

Requests cancellation of the innermost enclosing region of the type specified.

```
#pragma omp cancel [construct-type-clause[ , ] if-clause]

|$omp cancel [construct-type-clause[ , ] if-clause]
```

### Task creation

**task** [2.8.2] [2.8.3]

```
!$omp task [clause[ , clause] ...]

|$omp task [clause[ , clause] ...]
```

**taskloop** [2.10.2-3] [2.9.2-3]

```
!$omp taskloop [simd]

|$omp taskloop [simd]
```

**taskgroup** [2.17.7] [2.13.4]

```
!$omp taskgroup

|$omp taskgroup
```

### Declare directive

**declare reduction** [2.19.5.7] [2.16]

Declares a reduction-identifier used in a reduction clause.

```
#pragma omp declare reduction [ reduction-identifier : type-name-list ] [ [ initializer-clause ] ]

|$omp declare reduction [ reduction-identifier : type-name-list ] [ [ initializer-clause ] ]
```

### Functionality new/changed in OpenMP 5.0

- **depobj construct**
- **taskgroup**
- **taskwait**
- **taskgroup**
- **declare reduction**

Sections in the 5.0 spec. are in this color, and in OpenMP 4.5 spec. are in this color. *Deprecated* in the 5.0 spec.

[n.n.n] Sections in the 5.0 spec.  [n.n.n] Sections in the 4.5 spec.

### Fortran content

Sections in the 5.0 spec. are in this color, and in OpenMP 4.5 spec. are in this color. *Deprecated* in the 5.0 spec.

[n.n.n] Sections in the 5.0 spec.  [n.n.n] Sections in the 4.5 spec.
Clauses

Allocate Clause [2.11.4]

allocate (allocator: list)

Specifies the memory allocator to be used to obtain storage for private variables of a directive.

allocator:

C/C++:

An expression of type omp_allocator_handle_t

For integer expression of kindomp_allocator_handle_kind

Data Sharing Attribute Clauses [2.19.4][2.15.3]

Applies only to variables whose names are visible in the construct on which the clause appears.

default (shared | none) C/C++

default (private | firstprivate | shared | none) Fortran

shared (list)

Declares list items to be shared by tasks generated by parallel, teams, or task-generating construct.

private (list)

Declares list items to be private to a task and initializes.

firstprivate (list)

Declares list items to be private to a task, and initializes each of them with the value that the corresponding original item has when the construct is encountered.

lastprivate (list)

Declares one or more list items to be private to an implicit task or SIMD lane, and causes the corresponding original list item to be updated after the end of the region.

lastprivate-modifier: conditional

Dataflow Depend Clause [2.17.11][2.13.9]

Enforces additional constraints on the scheduling of tasks or loop iterations, establishing dependences only between sibling tasks or between loop iterations.

depend (depend-modifier, dependence-type : locator-list)

depend-modifier: Iterator (iterators-definition)

dependence-type: in, out, init, mutexinoutset, depend

• in: Generated task will be dependent of all previously generated sibling tasks that reference at least one of the list items in an in or out depend-type list.

• out and init: Generated task will be dependent of all previous sibling tasks referencing at least one of the list items in an in, out, or init depend-type list.

• mutexinoutset: If the storage location of at least one list item matches that of one appearing in a depend clause with an in, out, or init depend-type on a construct from which a sibling task was previously generated, then the generated task will be a dependent task of that sibling. If the storage location of at least one of the list items is the same as that of a list item appearing in a depend clause with a mutexinoutset dependence-type on a construct from which a sibling task was previously generated, then the sibling tasks will be mutually exclusive.

Parallel Dataflow Depend Clause [2.13.7-9]

OpenMP parallel constructs:

$omp parallel depend (depend-modifier: iterator -type)

$omp parallel depend (depend-modifier: iterator -type)

Parallel Dataflow Depend Clause [2.13.7-9]

OpenMP parallel tasks and SIMD:

$omp parallel task depend (depend-modifier: iterator -type)

$omp parallel task depend (depend-modifier: iterator -type)

Reduction Clauses [2.19.5]

in_reduction (reduction-identifier: list)

Specifies that a task participates in a reduction

reduction-identifier: Same as for reduction

reduction_task (reduction-identifier: list)

Specifies a reduction among tasks.

reduction-identifier: Same as for reduction

reduction reduction (reduction-identifier: list)

A non-negative numerical scalar expression that specifies a hint for the priority of the generated task.

private (list)

See Data Sharing Attribute Clauses. (Not used with task.)

num_tasks (num-tasks)

Create as many tasks as the minimum of the num-tasks expression and the number of logical loop iterations. (Not used with task.)

priority (priority-value)

A hint to each created task to be greater than or equal to the minimum of the value of the grain-size expression and the number of logical loop iterations, but less than twice the value of the grain-size expression. (Not used with task.)

If Clause [2.15][2.12]

Effect depends on the construct to which it is applied. For combined or composite constructs, it only applies to the semantics of the construct named in the directive-name-modifier if one is specified. If none is specified for a combined or composite construct then the if clause applies to all constructs to which an if clause can apply.

if ( [ directive-name-modifier: ] scalar-expression ) C/C++

if ( [ directive-name-modifier: ] scalar-logical-expression ) Fortran

See If Clause in this guide.

in_reduction (reduction-identifier: list)

See Reduction Clauses in this guide.

Reduction Clauses in this guide.

lastprivate (list)

See Data Sharing Attribute Clauses. (Not used with task.)

mergeable

Specifies that the generated task is a mergeable task.

nogroup

Prevents an implicit taskgroup region to be created. (Not used with task.)

num_tasks (num-tasks)

Create as many tasks as the minimum of the num-tasks expression and the number of logical loop iterations. (Not used with task.)

priority (priority-value)

A non-negative numerical scalar expression that specifies a hint for the priority of the generated task.

private (list)

See Data Sharing Attribute Clauses in this guide.

num_tasks (num-tasks)

See Reduction Clauses on this page. (Not used with task.)

share (list)

See Data Sharing Attribute Clauses in this guide.

united

If present, any thread in the team can resume the task region after a suspension.

Tasking Clauses [2.10][2.9]

affinity (aff-modifier: locator-list)

A hint to execute closely to the location of the list items. aff-modifier is iterator (iterators-definition). (Not used with taskloop.)

allocate (allocorator: list)

See Allocate Clause above.

collapse (n)

See SIMD Clauses on this page. (Not used with task.)

final (scalar-expression) C/C++

final (scalar-logical-expression) Fortran

See Reduction Clauses on this page. (Not used with task.)

firstprivate (list)

See Data Sharing Attribute Clauses in this guide.

grainsize (grain-size)

The generated task will be a final task if the final expression evaluates to true.

size of logical loop iterations assigned

parallel master taskloop [simd] [2.13.7-9]

Shortcut for specifying a parallel master construct containing a taskloop [simd] construct and no other statements.

#pragma omp parallel master taskloop [simd] [clause[ ]clause...]

for-loops

$omp parallel master taskloop [simd] [clause[ ]clause...]

do-loops

$omp master taskloop [simd] [clause[ ]clause...]

do-loops

$omp parallel taskloop [simd] [clause[ ]clause...]

for-loops

Scheduling and Ordering Attributes

spmd (spmd)

Spmd modifier is applied to the construct containing the spmd directive.

Scheduling and Ordering Attributes in this guide.