

OpenMP 5.0/5.1 Feature Subset Support
in XL C/C++ and XL Fortran for Linux, V16.1.1
Compilers

List of Feature Subset

- Unified shared memory
- Task detach
- Memory management
- Metadirective

These features are added as of XL C/++ and XL Fortran for Linux, V16.1.1 Fix Pack 12.

For the details of these features, please refer to the OpenMP 5.0/5.1 specifications.

<https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5.0.pdf>

<https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5-1.pdf>

1. Unified Shared Memory

Directive

Directive	Clause	Description
requires	unified_shared_memory	It guarantees that the host and target devices use a unified address space. With this clause, the MAP clause on TARGET constructs is optional, and the DECLARE TARGET directive is also optional.
Rules: <ul style="list-style-type: none">• The directive must appear lexically before any device constructs, or device routines.• The directive must appear in all compilation units that contain device constructs or device routines, or in none of them.		

Environment Variables

Env Var	Suboption	Description
XLSMPOPTS	targetmem=uonly	It enables the unified shared memory for all applicable variables. This is the default option.
	targetmem=uimplicit	It enables the unified shared memory for only the implicitly mapped variables, and declare target variables inside the device execution context. The explicitly mapped variables by the MAP clause still follow the data-mapping rules defined in the OpenMP specification.
Note: <ul style="list-style-type: none">• The suboptions above are effective only when the REQUIRES directive with unified_shared_memory clause is specified in the program.		

Compiler Options

Option	Description
-qxflag=check_missing_requires	It issues an informational message on potentially missing REQUIRES directive in a program unit when the use of REQUIRES directive is required.

2. Task Detach

Directive

Directive	Clause	Description
task	detach	It creates a new allow-completion event and connects it to the completion of the associated TASK. The list item of the clause is the event handle, which is considered as if it was specified on a FIRSTPRIVATE clause for the TASK construct. The TASK is completed when the execution of its associated structured block is completed and the allow-completion event is fulfilled.

Runtime Routine

Routine	Description
omp_fulfill_event	It fulfills and destroys an OpenMP allow-completion event.

3. Memory Management

Directive (Fortran only)

Directive	Clause	Description
allocate	allocator	It specifies an OpenMP memory allocator.
	align	It specifies the alignment, which is a constant scalar positive expression with a value that is a power of two.
Rules: <ul style="list-style-type: none"> • ALLOCATE directives that specify list items must be preceded by an executable statement or OpenMP construct. • ALLOCATE directives must be immediately followed by a Fortran allocate statement. • If multiple ALLOCATE directives are associated with a Fortran allocate statement, at most one directive may specify no list items. 		

Runtime Routines

Routine	Description
omp_init_allocator	It initializes an allocator and associates it with a memory space.
omp_destroy_allocator	It releases all resources used by the allocator handle.
omp_set_default_allocator	It sets the default memory allocator to be used by OpenMP allocation calls and ALLOCATE directives that do not specify an allocator.
omp_get_default_allocator	It returns a handle to the default memory allocator.
omp_alloc	It requests a memory allocation from a memory allocator.
omp_aligned_alloc	It requests a memory allocation from a memory allocator with a specified alignment.
omp_free	It deallocates previously allocated memory.
omp_calloc	It requests a zero initialized memory allocation from a memory allocator.
omp_aligned_calloc	It requests a zero initialized memory allocation from a memory allocator with a specified alignment.
omp_realloc	It deallocates previously allocated memory and requests a memory allocation from a memory allocator.

Environment Variables

Env Var	Description
OMP_ALLOCATOR	It sets the initial default allocator to one of the predefined allocators for allocation calls and directives that do not specify an allocator.

Memory Allocator Traits

Trait	Possible values	Default value	Description
alignment	A positive value integer value in byte that is a power of 2.	1 byte.	The allocated memory will be byte aligned to at least the value specified.
fallback	null_fb.	null_fb.*	It specifies how the allocator behaves when it cannot fulfill an allocation request. With null_fb, the allocator returns the value zero if it fails to allocate the memory.
pinned	false, true.	false.	It ensures that the memory allocated remains in the same storage resource at the same location for its entire lifetime.
Note: * Since only null_fb is supported in XL compiler, the default value is not the same as OpenMP 5.0 defines, which is default_mem_fb.			

Predefined Allocators

Allocator	Memory space	Non-default trait values
omp_default_mem_alloc	omp_default_mem_space	None.
ompx_pinned_mem_alloc*	omp_default_mem_space	pinned:true.
Note: * ompx_pinned_mem_alloc is an XL extension.		

4. Metadirective (Fortran only)

Directive

Directive	Clause	Description
metadirective OR begin metadirective end metadirective	when	It specifies a set of context selectors, and a directive variant that is executed if all the context selectors specified are evaluated to true.
	default	It specifies a directive variant that is executed if none of the context selectors at the WHEN clauses are evaluated to true.
Note: <ul style="list-style-type: none"> All OpenMP 4.5 directives are supported as the directive variants, except the following: <ul style="list-style-type: none"> sections, parallel sections, atomic The “metadirective” syntax is for loop-based and standalone directive variant. The “begin metadirective” and “end metadirective” syntax is for loop-based and construct-based directive variant. 		

Supported OpenMP Context Trait Sets and Traits

trait-set-selector-name	trait-selector-name	trait-score (optional)	trait-property
construct	target	N/A	N/A
	teams		
	parallel		
	do		
implementation	vendor	N/A	(vendor-name-list)
user	condition	score (compile-time only)	(scalar-logical-expr)