

### ThreadX Interface APIs

The following table provides more information on ThreadX Interface API level of support across different target OS's.

ThreadX API	Android	NetBSD	Linux / UNIX	RT Linux	LynxOS	QNX	Solaris	MQX	Nucleus	VxWorks	Windows 7/8	Windows Phone/CE	µITRON	µC/OS-III	FreeRTOS	ThreadX
<b>Kernel Initialization</b>																
tx_kernel_enter	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Static Memory</b>																
tx_block_allocate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_block_pool_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_block_pool_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_block_pool_info_get	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	N/A
tx_block_pool_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_block_pool_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_block_pool_prioritize	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_block_release	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Dynamic Memory</b>																
tx_byte_allocate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_byte_pool_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_byte_pool_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_byte_pool_info_get	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	Y <sup>1</sup>	N/A
tx_byte_pool_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_byte_pool_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_byte_pool_prioritize	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_block_release	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Queues</b>																
tx_queue_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_queue_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_queue_flush	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_queue_front_send	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_queue_info_get	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_queue_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_queue_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_queue_prioritize	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_queue_receive	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_queue_send	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_queue_send_notify	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Semaphores</b>																

ThreadX API	Android	NetBSD	Linux / UNIX	RT Linux	LynxOS	QNX	Solaris	MQX	Nucleus	VxWorks	Windows 7 / 8	Windows Phone/CE	µITRON	µC/OS-III	FreeRTOS	ThreadX
tx_semaphore_ceiling_put	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_semaphore_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_semaphore_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_semaphore_get	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_semaphore_info_get	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_semaphore_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_semaphore_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_semaphore_prioritize	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_semaphore_put	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_semaphore_put_notify	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Mutex</b>																
tx_mutex_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_mutex_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_mutex_get	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_mutex_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_mutex_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_mutex_prioritize	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_mutex_put	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Events</b>																
tx_event_flags_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_event_flags_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_event_flags_get	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_event_flags_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_event_flags_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_event_flags_set	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_event_flags_set_notify	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
<b>Task Control Block</b>																
tx_thread_create	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	Y <sup>2</sup>	N/A
tx_thread_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_entry_exit_notify	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_thread_identify	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_info_get	N	N	Y	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_thread_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_thread_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_thread_preemption_change	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	Y <sup>3</sup>	N/A
tx_thread_priority_change	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_relinquish	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_reset	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_resume	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A

ThreadX API	Android	NetBSD	Linux / UNIX	RT Linux	LynxOS	QNX	Solaris	MQX	Nucleus	VxWorks	Windows 7 / 8	Windows Phone/CE	µITRON	µC/OS-II	FreeRTOS	ThreadX
tx_thread_sleep	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_suspend	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_terminate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_thread_time_slice_change	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_thread_wait_abort	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
<b>Timer</b>																
tx_time_get	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_time_set	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_timer_activate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_timer_change	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_timer_create	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_timer_deactivate	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_timer_delete	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
tx_timer_info_get	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	Y <sup>4</sup>	N/A
tx_timer_performance_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
tx_timer_performance_system_info_get	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N/A
<b>Adaptive Native Thread</b>																
int_tx_task_entry	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A
os_adapt_native_thread_to_threadx_interface	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N/A

**Notes**

1: Only reports name and total blocks/bytes

2: ThreadX has a concept of prioritized preemption. Mapusoft do not support this feature at this point. Any positive value passed will be treated as pre-emptible task ( OS\_PREEMPT ). A zero value will be treated as a Task that cannot be pre-empted (OS\_NO\_PREEMPT)

3: Any positive value passed into preemption will be treated as OS\_PREEMPT. Values returned in old preemption will be equivalent to either OS\_PREEMPT or OS\_NO\_PREEMPT

4: Only reports name, active, remaining ticks and reschedule ticks