
CHAPTER FIVE

REQUIREMENTS

Further details for each requirement may be found on the RTEMS Classic API Guide [[con21c](#)].

5.1 Functional requirements

5.1.1 spec:/acfg/req/appl-disable-filesystem

spec:/acfg/req/appl-disable-filesystem

The application configuration option shall disable the system initialization of filesystems.

rationale: N/A

functional-type: function

This requirement refines [spec:/acfg/req/group](#).

Traced design component: RTEMSApplConfigFilesystemConfiguration - CONFIG-
URE_APPLICATION_DISABLE_FILESYSTEM

5.1.2 spec:/acfg/req/appl-does-not-need-clock-driver

spec:/acfg/req/appl-does-not-need-clock-driver

The application configuration option shall disable the system initialization of the Clock Driver.

rationale: N/A

functional-type: function

This requirement refines [spec:/acfg/req/group](#).

Traced design component: RTEMSApplConfigDeviceDriverConfiguration - CONFIG-
URE_APPLICATION_DOES_NOT_NEED_CLOCK_DRIVER

5.1.3 spec:/acfg/req/appl-needs-clock-driver

spec:/acfg/req/appl-needs-clock-driver

The application configuration option shall enable the initialization of the Clock Driver during system initialization.

rationale: N/A

functional-type: function

This requirement refines [spec:/acfg/req/group](#).

Traced design component: RTEMSApplConfigDeviceDriverConfiguration - CONFIGURATION_APPLICATION_NEEDS_CLOCK_DRIVER

5.1.4 spec:/acfg/req/bsp-initial-extension

spec:/acfg/req/bsp-initial-extension

Where CONFIGURE_DISABLE_BSP_SETTINGS is not defined by the application, where BSP_INITIAL_EXTENSION is defined by the BSP, when the application configuration is initialized, the extension set defined by BSP_INITIAL_EXTENSION shall be registered in the initial extension sets.

rationale: N/A

functional-type: function

This requirement refines [spec:/acfg/req/init](#).

5.1.5 spec:/acfg/req/bsp-prerequisite-drivers

spec:/acfg/req/bsp-prerequisite-drivers

Where CONFIGURE_DISABLE_BSP_SETTINGS is not defined by the application, where CONFIGURE_BSP_PREREQUISITE_DRIVERS is defined by the BSP, when the application configuration is initialized, the device driver entry points defined by CONFIGURE_BSP_PREREQUISITE_DRIVERS shall be registered in the Device Driver Table.

rationale: N/A

functional-type: function

This requirement refines [spec:/acfg/req/init](#).

5.1.6 spec:/acfg/req/disable-bsp-settings

spec:/acfg/req/disable-bsp-settings

The application configuration option shall disable the use of BSP-provided default values for application configuration options.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURATION_DISABLE_BSP_SETTINGS

5.1.7 spec:/acfg/req/disable-newlib-reentrancy

spec:/acfg/req/disable-newlib-reentrancy

The application configuration option shall disable the thread-specific Newlib reentrancy support.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURATION_DISABLE_NEWLIB_REENTRANCY

5.1.8 spec:/acfg/req/fatal-init-task-construct-failed

spec:/acfg/req/fatal-init-task-construct-failed

Where the application was configured to construct the RTEMS initialization task using CONFIGURE_INIT_TASK_CONSTRUCT_STORAGE_SIZE, while the system is initialized, if the task construction fails, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_RTEMS_INIT_TASK_CONSTRUCT_FAILED fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/acfg/req/init-task-construct-storage-size*.

5.1.9 spec:/acfg/req/fatal-too-large-tls-size

spec:/acfg/req/fatal-too-large-tls-size

Where the application was configured with a value not equal to zero for CONFIGURE_MAXIMUM_THREAD_LOCAL_STORAGE_SIZE, while the system is initialized, if the actual thread-local storage size of the application is greater than the configure maximum size, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_TOO_LARGE_TLS_SIZE fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/acfg/req/max-thread-local-storage-size*.

5.1.10 spec:/acfg/req/idle-task-body

spec:/acfg/req/idle-task-body

The application configuration option shall define the IDLE task body.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigIdleTaskConfiguration - CONFIGURATION_IDLE_TASK_BODY

5.1.11 spec:/acfg/req/idle-task-init-appl

spec:/acfg/req/idle-task-init-appl

The application configuration option shall indicate that the user has configured **no** user initialization tasks or threads and that the user-provided IDLE task will perform application initialization and then transform itself into an IDLE task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigIdleTaskConfiguration - CONFIGURATION_IDLE_TASK_INITIALIZES_APPLICATION

5.1.12 spec:/acfg/req/idle-task-stack-size

spec:/acfg/req/idle-task-stack-size

The application configuration option shall define the task stack size for an IDLE task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigIdleTaskConfiguration
-
CONFIG-
URE_IDLE_TASK_STACK_SIZE

5.1.13 spec:/acfg/req/init

spec:/acfg/req/init

While CONFIGURE_INIT is defined, while rtems/confdefs.h is included in a translation unit after the definition of CONFIGURE_INIT, when the translation unit is preprocessed and compiled, the system settings defined by present application configuration options shall be statically allocated and initialized.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/bsp-initial-extension*
- *spec:/acfg/req/bsp-prerequisite-drivers*
- *spec:/acfg/req/malloc-bsp-supports-sbrk*

5.1.14 spec:/acfg/req/init-task-arguments

spec:/acfg/req/init-task-arguments

The application configuration option shall define the task argument of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_ARGUMENTS

5.1.15 spec:/acfg/req/init-task-attributes

spec:/acfg/req/init-task-attributes

The application configuration option shall define the task attributes of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_ATTRIBUTES

5.1.16 spec:/acfg/req/init-task-construct-storage-size

spec:/acfg/req/init-task-construct-storage-size

The application configuration option shall define the task storage size of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/fatal-init-task-construct-failed*

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_CONSTRUCT_STORAGE_SIZE

5.1.17 spec:/acfg/req/init-task-entrypoint

spec:/acfg/req/init-task-entrypoint

The application configuration option shall define the entry point of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_ENTRY_POINT

5.1.18 spec:/acfg/req/init-task-initial-modes

spec:/acfg/req/init-task-initial-modes

The application configuration option shall define the initial execution mode of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_INITIAL_MODES

5.1.19 spec:/acfg/req/init-task-name

spec:/acfg/req/init-task-name

The application configuration option shall define the name of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_NAME

5.1.20 spec:/acfg/req/init-task-priority

spec:/acfg/req/init-task-priority

The application configuration option shall define the initial priority of the Classic API initialization task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
CONFIGURE_INIT_TASK_PRIORITY

5.1.21 spec:/acfg/req/initial-extensions

spec:/acfg/req/initial-extensions

The application configuration option shall add the value to the initial extension sets.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURATION_INITIAL_EXTENSIONS

5.1.22 spec:/acfg/req/interrupt-stack-size

spec:/acfg/req/interrupt-stack-size

The application configuration option shall define the size of an interrupt stack in bytes.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURATION_INTERRUPT_STACK_SIZE

5.1.23 spec:/acfg/req/malloc-bsp-supports-sbrk

spec:/acfg/req/malloc-bsp-supports-sbrk

Where CONFIGURE_DISABLE_BSP_SETTINGS is not defined by the application, where CONFIGURE_MALLOC_BSP_SUPPORTS_SBRK is defined by the BSP, when the application configuration is initialized, not all memory shall made available to the C Program Heap immediately at system initialization time.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/init*.

5.1.24 spec:/acfg/req/max-barriers

spec:/acfg/req/max-barriers

The application configuration option shall define the maximum number of Classic API Barriers that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIGURATION_MAXIMUM_BARRIERS

5.1.25 spec:/acfg/req/max-file-descriptors

spec:/acfg/req/max-file-descriptors

The application configuration option shall define the maximum number of file like objects that can be concurrently open.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURATION_MAXIMUM_FILE_DESCRIPTORSS

5.1.26 spec:/acfg/req/max-message-queues

spec:/acfg/req/max-message-queues

The application configuration option shall define the maximum number of Classic API Message Queues that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIG-
URE_MAXIMUM_MESSAGE_QUEUES

5.1.27 spec:/acfg/req/max-partitions

spec:/acfg/req/max-partitions

The application configuration option shall define the maximum number of Classic API Partitions that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIG-
URE_MAXIMUM_PARTITIONS

5.1.28 spec:/acfg/req/max-periods

spec:/acfg/req/max-periods

The application configuration option shall define the maximum number of Classic API Periods that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIG-
URE_MAXIMUM_PERIODS

5.1.29 spec:/acfg/req/max-priority

spec:/acfg/req/max-priority

The application configuration option shall define the maximum numeric priority of tasks.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSchedulerConfiguration - CONFIGURE_MAXIMUM_PRIORITY

5.1.30 spec:/acfg/req/max-processors

spec:/acfg/req/max-processors

The application configuration option shall define the maximum number of processors an application intends to use.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURE_MAXIMUM_PROCESSORS

5.1.31 spec:/acfg/req/max-semaphores

spec:/acfg/req/max-semaphores

The application configuration option shall define the maximum number of Classic API Semaphore that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIGURE_MAXIMUM_SEMAPHORES

5.1.32 spec:/acfg/req/max-tasks

spec:/acfg/req/max-tasks

The application configuration option shall define the maximum number of Classic API Tasks that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIGURATION_MAXIMUM_TASKS

5.1.33 spec:/acfg/req/max-thread-local-storage-size

spec:/acfg/req/max-thread-local-storage-size

The application configuration option shall define the maximum thread-local storage size in bytes.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/fatal-too-large-tls-size*

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIGURATION_MAXIMUM_THREAD_LOCAL_STORAGE_SIZE

5.1.34 spec:/acfg/req/max-timers

spec:/acfg/req/max-timers

The application configuration option shall define the maximum number of Classic API Timers that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIGURATION_MAXIMUM_TIMERS

5.1.35 spec:/acfg/req/max-user-extensions

spec:/acfg/req/max-user-extensions

The application configuration option shall define the maximum number of Classic API User Extensions that can be concurrently active.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration - CONFIG-
URE_MAXIMUM_USER_EXTENSIONS

5.1.36 spec:/acfg/req/microseconds-per-tick

spec:/acfg/req/microseconds-per-tick

The application configuration option shall define the length of time in microseconds between clock ticks (clock tick quantum).

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIG-
URE_MICROSECONDS_PER_TICK

5.1.37 spec:/acfg/req/min-task-stack-size

spec:/acfg/req/min-task-stack-size

The application configuration option shall define the minimum stack size in bytes for every user task or thread in the system.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIG-
URE_MINIMUM_TASK_STACK_SIZE

5.1.38 spec:/acfg/req/min-tasks-with-user-provided-storage

spec:/acfg/req/min-tasks-with-user-provided-storage

The application configuration option shall define the minimum count of Classic API Tasks which are constructed by rtems_task_construct.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIConfiguration
-
CONFIGURATION_MINIMUM_TASKS_WITH_USER_PROVIDED_STORAGE

5.1.39 spec:/acfg/req/rtems-init-tasks-table

spec:/acfg/req/rtems-init-tasks-table

The application configuration option shall enable that exactly one Classic API initialization task is configured.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigClassicAPIInitializationTaskConfiguration
-
CONFIGURE_RTEMS_INIT_TASKS_TABLE

5.1.40 spec:/acfg/req/scheduler-assign-attributes

spec:/acfg/req/scheduler-assign-attributes

While the RTEMS_SCHEDULER_ASSIGN() macro is used to define an entry of a CONFIGURATION_SCHEDULER_ASSIGNMENTS application configuration option value, the attributes defined by the second parameter shall define the scheduler assignment attributes for the processor associated with the entry.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/scheduler-assignments*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/scheduler-assign-processor-mandatory*
- *spec:/acfg/req/scheduler-assign-processor-optional*

Traced design component: None

5.1.41 spec:/acfg/req/scheduler-assign-no-scheduler

spec:/acfg/req/scheduler-assign-no-scheduler

While the RTEMS_SCHEDULER_ASSIGN_NO_SCHEDULER define is used to define an entry of a CONFIGURE_SCHEDULER_ASSIGNMENTS application configuration option value, the processor associated with the entry shall not have a scheduler assigned.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/scheduler-assignments*.

Traced design component: None

5.1.42 spec:/acfg/req/scheduler-assign-processor-mandatory

spec:/acfg/req/scheduler-assign-processor-mandatory

While the RTEMS_SCHEDULER_ASSIGN_PROCESSOR_MANDATORY define is used as the second parameter in the RTEMS_SCHEDULER_ASSIGN() macro in an entry of a CONFIGURE_SCHEDULER_ASSIGNMENTS application configuration option value, the presence of the processor associated with the entry in the target shall be mandatory.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/scheduler-assign-attributes*.

Traced design component: None

5.1.43 spec:/acfg/req/scheduler-assign-processor-optional

spec:/acfg/req/scheduler-assign-processor-optional

While the RTEMS_SCHEDULER_ASSIGN_PROCESSOR_OPTIONAL define is used as the second parameter in the RTEMS_SCHEDULER_ASSIGN() macro in an entry of a CONFIGURE_SCHEDULER_ASSIGNMENTS application configuration option value, the presence of the processor associated with the entry in the target shall be optional.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/scheduler-assign-attributes*.

Traced design component: None

5.1.44 spec:/acfg/req/scheduler-assign-scheduler

spec:/acfg/req/scheduler-assign-scheduler

While the RTEMS_SCHEDULER_ASSIGN() macro is used to define an entry of a CONFIGURE_SCHEDULER_ASSIGNMENTS application configuration option value, the scheduler associated with the first parameter shall be assigned to the processor associated with the entry.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/scheduler-assignments*.

Traced design component: None

5.1.45 spec:/acfg/req/scheduler-assignments

spec:/acfg/req/scheduler-assignments

The application configuration option shall define the initial scheduler to processor assignments.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/scheduler-assign-attributes*
- *spec:/acfg/req/scheduler-assign-no-scheduler*
- *spec:/acfg/req/scheduler-assign-scheduler*

Traced design component: RTEMSApplConfigGeneralSchedulerConfiguration - CONFIGURE_SCHEDULER_ASSIGNMENTS

5.1.46 spec:/acfg/req/scheduler-edf-smp

spec:/acfg/req/scheduler-edf-smp

The application configuration option shall make the Earliest Deadline First SMP Scheduler algorithm available to the application.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSchedulerConfiguration - CONFIGURATION_SCHEDULER_EDF_SMP

5.1.47 spec:/acfg/req/scheduler-name

spec:/acfg/req/scheduler-name

The application configuration option shall define the name of the default scheduler.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSchedulerConfiguration - CONFIGURATION_SCHEDULER_NAME

5.1.48 spec:/acfg/req/scheduler-priority

spec:/acfg/req/scheduler-priority

The application configuration option shall make the Deterministic Priority Scheduler algorithm available to the application.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSchedulerConfiguration - CONFIGURATION_SCHEDULER_PRIORITY

5.1.49 spec:/acfg/req/scheduler-table-entries

spec:/acfg/req/scheduler-table-entries

The application configuration option shall define the entries of the table of configured schedulers.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSchedulerConfiguration - CONFIGURATION_SCHEDULER_TABLE_ENTRIES

5.1.50 spec:/acfg/req/task-stack-allocator

spec:/acfg/req/task-stack-allocator

The application configuration option shall define the stack allocator allocate handler.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigTaskStackAllocatorConfiguration - CONFIGURATION_TASK_STACK_ALLOCATOR

5.1.51 spec:/acfg/req/task-stack-allocator-for-idle

spec:/acfg/req/task-stack-allocator-for-idle

The application configuration option shall define the stack allocator allocate handler for an IDLE task.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigTaskStackAllocatorConfiguration - CONFIGURATION_TASK_STACK_ALLOCATOR_FOR_IDLE

5.1.52 spec:/acfg/req/task-stack-deallocator

spec:/acfg/req/task-stack-deallocator

The application configuration option shall define the stack allocator deallocate handler.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigTaskStackAllocatorConfiguration - CONFIGURE_TASK_STACK_DEALLOCATOR

5.1.53 spec:/acfg/req/task-stack-no-workspace

spec:/acfg/req/task-stack-no-workspace

The application configuration option shall indicate that the task stack allocator does not use the RTEMS Workspace.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigTaskStackAllocatorConfiguration - CONFIGURE_TASK_STACK_ALLOCATOR_AVOIDS_WORK_SPACE

5.1.54 spec:/acfg/req/ticks-per-time-slice

spec:/acfg/req/ticks-per-time-slice

The application configuration option shall define the length of the timeslice quantum for each task in clock ticks.

rationale: N/A

functional-type: function

This requirement refines *spec:/acfg/req/group*.

Traced design component: RTEMSApplConfigGeneralSystemConfiguration - CONFIGURE_TICKS_PER_TIMESLICE

5.1.55 spec:/bsp/req/interrupt-spurious

spec:/bsp/req/interrupt-spurious

When the bsp_interrupt_spurious() (see: spec:/bsp/if/interrupt-spurious) directive is called.

rationale: N/A

functional-type: action

Traced design component: bsp_interrupt - bsp_interrupt_spurious

5.1.55.1 pre-conditions

First

Null While the pointer to the first interrupt entry of the interrupt vector specified by the vector parameter is equal to NULL.

Entry While the pointer to the first interrupt entry of the interrupt vector specified by the vector parameter references an object of type rtems_interrupt_entry.

5.1.55.2 post-conditions

Result

FatalError A fatal error shall occur.

Dispatch The interrupt entries installed at the interrupt vector specified by the vector parameter shall be dispatched.

FatalSource

SpuriousInterrupt The fatal source shall be equal to RTEMS_FATAL_SOURCE_SPURIOUS_INTERRUPT.

FatalCode

Vector The fatal code shall be equal to the vector parameter.

5.1.55.3 transition-map

Result = *FatalError*, FatalSource = *SpuriousInterrupt*, FatalCode = *Vector*

- First = *Null*

Result = *Dispatch*, FatalSource = NA, FatalCode = NA

- First = *Entry*

5.1.56 spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-fix

spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-fix

The TN0018_FIX macro shall provide an instruction sequence for *Workaround #1* to be placed immediatly before a jumpl followed by rett pair.

rationale: N/A

functional-type: function

This requirement refines [spec:/bsp/sparc/leon3/req/errata-tn-0018-macros](#).

5.1.57 spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-wait-iflush

spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-wait-iflush

The TN0018_WAIT_IFLUSH macro shall flush the instruction cache.

rationale: N/A

functional-type: function

This requirement refines [spec:/bsp/sparc/leon3/req/errata-tn-0018-macros](#).

5.1.58 spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-write-psr

spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-write-psr

The TN0018_WRITE_PSR macro shall write the source register to register %psr.

rationale: N/A

functional-type: function

This requirement refines [spec:/bsp/sparc/leon3/req/errata-tn-0018-macros](#).

5.1.59 spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-boot

spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-boot

While the system is initialized, if the data cache snooping is disabled on the boot processor, then the system shall terminate with the RTEMS_FATAL_SOURCE_BSP fatal source and the LEON3_FATAL_INVALID_CACHE_CONFIG_BOOT_PROCESSOR fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.1.60 spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-secondary

spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-secondary

While the system is initialized, if the data cache snooping is disabled on a secondary processor, then the system shall terminate with the RTEMS_FATAL_SOURCE_BSP fatal source and the LEON3_FATAL_INVALID_CACHE_CONFIG_SECONDARY_PROCESSOR fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.1.61 spec:/bsp/sparc/leon3/req/fatal-clock-initialization

spec:/bsp/sparc/leon3/req/fatal-clock-initialization

If the installation of the clock interrupt entry fails, then the system shall terminate with the RTEMS_FATAL_SOURCE_BSP fatal source and the LEON3_FATAL_CLOCK_INITIALIZATION fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.1.62 spec:/bsp/sparc/leon3/req/fatal-shutdown-request

spec:/bsp/sparc/leon3/req/fatal-shutdown-request

While the fatal source is not RTEMS_FATAL_SOURCE_SMP or the fatal code is not SMP_SHUTDOWN_RESPONSE, the bsp_fatal_extension() directive shall request all other configured and present processors to power down.

rationale: N/A

functional-type: function

Traced design component: None

5.1.63 spec:/bsp/sparc/leon3/req/fatal-shutdown-response

spec:/bsp/sparc/leon3/req/fatal-shutdown-response

While the fatal source is RTEMS_FATAL_SOURCE_SMP, while the fatal code is SMP_SHUTDOWN_RESPONSE, the bsp_fatal_extension() directive shall power down the processor.

rationale: N/A

functional-type: function

Traced design component: None

5.1.64 spec:/bsp/sparc/leon3/req/fatal-shutdown-self

spec:/bsp/sparc/leon3/req/fatal-shutdown-self

While the fatal source is not RTEMS_FATAL_SOURCE_SMP or the fatal code is not SMP_SHUTDOWN_RESPONSE, while all shutdown requests responded or a timeout occurred, the bsp_fatal_extension() directive shall halt the processor.

rationale: N/A

functional-type: function

Traced design component: None

5.1.65 spec:/bsp/sparc/leon3/req/fatal-shutdown-timeout

spec:/bsp/sparc/leon3/req/fatal-shutdown-timeout

While the fatal source is not RTEMS_FATAL_SOURCE_SMP or the fatal code is not SMP_SHUTDOWN_RESPONSE, while all shutdown requests were issued, the bsp_fatal_extension() directive shall wait for at most approximately one millisecond for the power down response of all other configured and present processors.

rationale: N/A

functional-type: function

Traced design component: None

5.1.66 spec:/bsp/sparc/leon3/req/idle-task-body

spec:/bsp/sparc/leon3/req/idle-task-body

The BSP shall define BSP_IDLE_TASK_BODY to bsp_idle_thread().

rationale: N/A

functional-type: function

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/leon3/req/idle-task-power-down*
- *spec:/bsp/sparc/leon3/req/idle-task-power-down-errata*

Traced design component: None

5.1.67 spec:/bsp/sparc/leon3/req/idle-task-power-down

spec:/bsp/sparc/leon3/req/idle-task-power-down

The bsp_idle_thread() function shall perform an endless loop which sets the processor into the power-down mode using a write to %asr19 in each iteration.

rationale: N/A

functional-type: function

This requirement refines *spec:/bsp/sparc/leon3/req/idle-task-body*.

5.1.68 spec:/bsp/sparc/leon3/req/idle-task-power-down-errata

spec:/bsp/sparc/leon3/req/idle-task-power-down-errata

The bsp_idle_thread() function shall be implemented according to Workaround 2 of {errata-gr712rc-08}.

rationale: N/A

functional-type: function

This requirement refines *spec:/bsp/sparc/leon3/req/idle-task-body*.

5.1.69 spec:/bsp/sparc/req/grlib-load-08

spec:/bsp/sparc/req/grlib-load-08

The grlib_load_8 directive shall load an unsigned 8-bit value from the address specified by address and return it.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_load_8

5.1.70 spec:/bsp/sparc/req/grlib-load-16

spec:/bsp/sparc/req/grlib-load-16

The grlib_load_16 directive shall load an unsigned 16-bit value from the address specified by address and return it.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_load_16

5.1.71 spec:/bsp/sparc/req/grlib-load-32

spec:/bsp/sparc/req/grlib-load-32

The grlib_load_32 directive shall load an unsigned 32-bit value from the address specified by address and return it.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_load_32

5.1.72 spec:/bsp/sparc/req/grlib-load-64

spec:/bsp/sparc/req/grlib-load-64

The grlib_load_64 directive shall load an unsigned 64-bit value from the address specified by address and return it.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_load_64

5.1.73 spec:/bsp/sparc/req/grlib-store-08

spec:/bsp/sparc/req/grlib-store-08

The grlib_store_8 directive shall store the unsigned 8-bit value specified by value to the address specified by address.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_store_8

5.1.74 spec:/bsp/sparc/req/grlib-store-16

spec:/bsp/sparc/req/grlib-store-16

The grlib_store_16 directive shall store the unsigned 16-bit value specified by value to the address specified by address.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_store_16

5.1.75 spec:/bsp/sparc/req/grlib-store-32

spec:/bsp/sparc/req/grlib-store-32

The grlib_store_32 directive shall store the unsigned 32-bit value specified by value to the address specified by address.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_store_32

5.1.76 spec:/bsp/sparc/req/grlib-store-64

spec:/bsp/sparc/req/grlib-store-64

The grlib_store_64 directive shall store the unsigned 64-bit value specified by value to the address specified by address.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBIO - grlib_store_64

5.1.77 spec:/c/req/clock-nanosleep

spec:/c/req/clock-nanosleep

When the clock_nanosleep() (see: spec:/c/if/clock-nanosleep) directive is called.

rationale: N/A

functional-type: action

Traced design component: None

5.1.77.1 pre-conditions

ClockId

Monotonic While the `clock_id` parameter is equal to `CLOCK_MONOTONIC`.

Realtime While the `clock_id` parameter is equal to `CLOCK_REALTIME`.

Invalid While the `clock_id` parameter is an invalid clock identifier.

Abstime

Yes While the `flags` parameter indicates an absolute time.

No While the `flags` parameter does not indicate an absolute time.

RQTp

Valid While the `rqtp` parameter references an object of type `struct timespec`.

Null While the `\rqtp` parameter is equal to `NULL`.

RQTpNSec

Valid While the `tv_nsec` member of the object referenced by the `rqtp` parameter is a valid nanoseconds value.

Invalid While the `tv_nsec` member of the object referenced by the `rqtp` parameter is an invalid nanoseconds value.

RQTpSec

Negative While the `tv_sec` member of the object referenced by the `rqtp` parameter is negative.

FarFuture While the `tv_sec` member of the object referenced by the `rqtp` parameter specifies a time point which is past the implementation limit.

Future While the `tv_sec` member of the object referenced by the `rqtp` parameter specifies a time point which is after the current time of the clock specified by the `clock_id` parameter and is within the implementation limits.

PastOrNow While the `tv_sec` member of the object referenced by the `rqtp` parameter is non-negative and specifies a time point which is before or at the current time of the clock specified by the `clock_id` parameter.

RMTp

Valid While the `rmtp` parameter references an object of type `struct timespec`.

Null While the `\rmtp` parameter is equal to `NULL`.

5.1.77.2 post-conditions

Status

Zero The return value of `clock_nanosleep()` shall be equal to zero.

ENOTSUP The return value of `clock_nanosleep()` shall be equal to ENOTSUP.

EINVAL The return value of `clock_nanosleep()` shall be equal to EINVAL.

Timer

Inactive The timer of the calling task shall be inactive.

Monotonic The timer of the calling task shall be active using the CLOCK_MONOTONIC.

Realtime The timer of the calling task shall be active using the CLOCK_REALTIME.

Expire

Last The timer of the calling task shall expire at the last valid time point of the clock specified by the `clock_id` parameter.

Absolute The timer of the calling task shall expire at the time point specified by the `rqtp` parameter.

Relative The timer of the calling task shall expire at the time point specified by the sum of the current time of the clock specified by the `clock_id` parameter and the interval specified by the `rqtp` parameter.

Scheduler

Block The calling task shall be blocked by the scheduler exactly once by the `clock_nanosleep()` call.

BlockUnblock The calling task shall be blocked exactly once by the scheduler and then unblocked in the same thread dispatch critical section by the `clock_nanosleep()` call.

Nop The calling task shall not be altered by the scheduler by the `clock_nanosleep()` call.

RMTp

Zero The object referenced by the `rmtp` parameter shall be cleared to zero after the return of the `clock_nanosleep()` call.

Nop Objects referenced by the `rmtp` parameter in past calls to `clock_nanosleep()` shall not be accessed by the `clock_nanosleep()` call.

5.1.77.3 skip-reasons

NeedClock The terms past and future need a clock for reference.

5.1.77.4 transition-map

Status = *Zero*, Timer = *Monotonic*, Expire = *Last*, Scheduler = *Block*, RMTp = *Zero*

- ClockId = *Monotonic*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *FarFuture*, RMTp = *Valid*

Status = *Zero*, Timer = *Monotonic*, Expire = *Relative*, Scheduler = *Block*, RMTp = *Zero*

- ClockId = *Monotonic*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *Future*, RMTp = *Valid*

Status = *Zero*, Timer = *Monotonic*, Expire = *Relative*, Scheduler = *Block*, RMTp = *Nop*

- ClockId = *Monotonic*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *Future*, RMTp = *Null*

Status = *Zero*, Timer = *Realtime*, Expire = *Last*, Scheduler = *Block*, RMTp = *Zero*

- ClockId = *Realtime*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *FarFuture*, RMTp = *Valid*

Status = *Zero*, Timer = *Realtime*, Expire = *Relative*, Scheduler = *Block*, RMTp = *Zero*

- ClockId = *Realtime*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *Future*, RMTp = *Valid*

Status = *Zero*, Timer = *Realtime*, Expire = *Relative*, Scheduler = *Block*, RMTp = *Nop*

- ClockId = *Realtime*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *Future*, RMTp = *Null*

Status = *Zero*, Timer = *Monotonic*, Expire = *Absolute*, Scheduler = *Block*, RMTp = *Nop*

- ClockId = *Monotonic*, Abstime = *Yes*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *Future*, RMTp = { *Valid*, *Null* }

Status = *Zero*, Timer = *Inactive*, Expire = NA, Scheduler = *BlockUnblock*, RMTp = *Zero*

- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *PastOrNow*, RMTp = *Valid*

Status = *Zero*, Timer = *Realtime*, Expire = *Absolute*, Scheduler = *Block*, RMTp = *Nop*

- ClockId = *Realtime*, Abstime = *Yes*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *Future*, RMTp = { *Valid*, *Null* }

Status = *Zero*, Timer = *Monotonic*, Expire = *Last*, Scheduler = *Block*, RMTp = *Nop*

- ClockId = *Monotonic*, Abstime = *Yes*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *FarFuture*, RMTp = { *Valid*, *Null* }

- ClockId = *Monotonic*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *FarFuture*, RMTp = *Null*

Status = *Zero*, Timer = *Realtime*, Expire = *Last*, Scheduler = *Block*, RMTp = *Nop*

- ClockId = *Realtime*, Abstime = *Yes*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *FarFuture*, RMTp = { *Valid*, *Null* }
- ClockId = *Realtime*, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *FarFuture*, RMTp = *Null*

Status = *Zero*, Timer = *Inactive*, Expire = NA, Scheduler = *BlockUnblock*, RMTp = *Nop*

- ClockId = { *Monotonic*, *Realtime* }, Abstime = *Yes*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = { *Negative*, *PastOrNow* }, RMTp = { *Valid*, *Null* }
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Valid*, RQTpSec = *PastOrNow*, RMTp = *Null*

Status = *EINVAL*, Timer = *Inactive*, Expire = NA, Scheduler = *BlockUnblock*, RMTp = *Zero*

- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Valid*, RQTpNSec = { *Valid*, *Invalid* }, RQTpSec = *Negative*, RMTp = *Valid*
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Invalid*, RQTpSec = { *FarFuture*, *Future*, *PastOrNow* }, RMTp = *Valid*
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Null*, RQTpNSec = NA, RQTpSec = NA, RMTp = *Valid*

Status = *ENOTSUP*, Timer = *Inactive*, Expire = NA, Scheduler = *Nop*, RMTp = *Nop*

- ClockId = *Invalid*, Abstime = { *Yes*, *No* }, RQTp = *Valid*, RQTpNSec = { *Valid*, *Invalid* }, RQTpSec = { *Negative*, *FarFuture* }, RMTp = { *Valid*, *Null* }
- ClockId = *Invalid*, Abstime = { *Yes*, *No* }, RQTp = *Null*, RQTpNSec = NA, RQTpSec = NA, RMTp = { *Valid*, *Null* }

Status = *EINVAL*, Timer = *Inactive*, Expire = NA, Scheduler = *BlockUnblock*, RMTp = *Nop*

- ClockId = { *Monotonic*, *Realtime* }, Abstime = *Yes*, RQTp = *Valid*, RQTpNSec = *Invalid*, RQTpSec = { *Negative*, *FarFuture*, *Future*, *PastOrNow* }, RMTp = { *Valid*, *Null* }
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *Yes*, RQTp = *Null*, RQTpNSec = NA, RQTpSec = NA, RMTp = { *Valid*, *Null* }
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Valid*, RQTpNSec = { *Valid*, *Invalid* }, RQTpSec = *Negative*, RMTp = *Null*
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Valid*, RQTpNSec = *Invalid*, RQTpSec = { *FarFuture*, *Future*, *PastOrNow* }, RMTp = *Null*
- ClockId = { *Monotonic*, *Realtime* }, Abstime = *No*, RQTp = *Null*, RQTpNSec = NA, RQTpSec = NA, RMTp = *Null*

NeedClock

- ClockId = *Invalid*, Abstime = { *Yes*, *No* }, RQTp = *Valid*, RQTpNSec = { *Valid*, *Invalid* }, RQTpSec = { *Future*, *PastOrNow* }, RMTp = { *Valid*, *Null* }

5.1.78 spec:/c/req/flsl

spec:/c/req/flsl

When the flsl() (see: spec:/c/if/flsl) directive is called.

rationale: N/A

functional-type: action

Traced design component: None

5.1.78.1 pre-conditions

Value

Zero While the parameter value is equal to zero.

NonZero While the parameter value is not equal to zero.

5.1.78.2 post-conditions

Result

Zero The return value of flsl() shall be equal to zero.

LastBitSet The return value of flsl() shall be equal to the index of the most-significant bit set in the parameter value.

5.1.78.3 transition-map

Result = *Zero*

- Value = *Zero*

Result = *LastBitSet*

- Value = *NonZero*

5.1.79 spec:/c/req/memcpy

spec:/c/req/memcpy

The memcpy() function shall be specified by C11.

rationale: N/A

functional-type: function

Traced design component: None

5.1.80 spec:/c/req/memset

spec:/c/req/memset

The memset() function shall be specified by C11.

rationale: N/A

functional-type: function

Traced design component: None

5.1.81 spec:/c/req posix-memalign

spec:/c/req posix-memalign

When the posix_memalign() (see: [spec:/c/if posix-memalign](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: None

5.1.81.1 pre-conditions

Memptr

Valid While the memptr parameter references an object of type void *.

Null While the memptr parameter is equal to NULL.

Alignment

Tiny While the alignment parameter is less than sizeof(void *).

NotPower2 While the alignment parameter is greater than or equal to sizeof(void *), while the alignment parameter is not a power of two.

Huge While the alignment parameter is greater than or equal to sizeof(void *), while the alignment parameter is a power of two, while the alignment parameter is too large to allocate a memory area with the specified alignment.

Valid While the alignment parameter is greater than or equal to sizeof(void *), while the alignment parameter is a power of two, while the alignment parameter is small enough to allocate a memory area with the specified alignment.

Size

Huge While the size parameter is not equal to zero, while the size parameter is too large to allocate a memory area with the specified size.

Zero While the size parameter is equal to zero.

Valid While the size parameter is not equal to zero, while the size parameter is small enough to allocate a memory area with the specified size.

5.1.81.2 post-conditions

Status

Zero The return value of posix_memalign() shall be equal to zero.

EINVAL The return value of posix_memalign() shall be equal to EINVAL.

ENOMEM The return value of posix_memalign() shall be equal to ENOMEM.

MemptrVar

AreaBegin The value of the object referenced by the memptr parameter shall be set to the begin address of the allocated memory area after the return of the posix_memalign() call.

Null The value of the object referenced by the memptr parameter shall be set to NULL after the return of the posix_memalign() call.

Nop Objects referenced by the memptr parameter in past calls to posix_memalign() shall not be accessed by the posix_memalign() call.

Alignment

Valid The begin address of the allocated memory area shall be an integral multiple of the alignment parameter.

Size

Valid The size of the allocated memory area shall greater than or equal to the size parameter.

5.1.81.3 transition-map

Status = **Zero**, MemptrVar = **AreaBegin**, Alignment = **Valid**, Size = **Valid**

- Memptr = **Valid**, Alignment = **Valid**, Size = **Valid**

Status = **Zero**, MemptrVar = **Null**, Alignment = **Valid**, Size = NA

- Memptr = **Valid**, Alignment = { **Huge**, **Valid** }, Size = **Zero**

Status = **ENOMEM**, MemptrVar = **Null**, Alignment = NA, Size = NA

- Memptr = **Valid**, Alignment = **Huge**, Size = { **Huge**, **Valid** }
- Memptr = **Valid**, Alignment = **Valid**, Size = **Huge**

Status = **EINVAL**, MemptrVar = **Null**, Alignment = NA, Size = NA

- Memptr = *Valid*, Alignment = { *Tiny*, *NotPower2* }, Size = { *Huge*, *Zero*, *Valid* }

Status = *EINVAL*, MemptrVar = *Nop*, Alignment = NA, Size = NA

- Memptr = *Null*, Alignment = { *Tiny*, *NotPower2*, *Huge*, *Valid* }, Size = { *Huge*, *Zero*, *Valid* }

5.1.82 spec:/dev/grlib/req/apbuart-inbyte-nonblocking

spec:/dev/grlib/req/apbuart-inbyte-nonblocking

When the apbuart_inbyte_nonblocking() (see: spec:/dev/grlib/if/apbuart-inbyte-nonblocking) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSDeviceGRLIBAPBUART - apbuart_inbyte_nonblocking

5.1.82.1 pre-conditions

DataReady

Yes While the data ready flag is set in the status register of the register block specified by regs parameter.

No While the data ready flag is cleared in the status register of the register block specified by regs parameter.

5.1.82.2 post-conditions

Result

Data The return value of apbuart_inbyte_nonblocking shall be the data read from the data register of the register block specified by regs.

MinusOne The return value of apbuart_inbyte_nonblocking shall be minus one.

ErrorFlags

Cleared The framing error, parity error, overrun, and break received flags in the status register of the register block specified by regs shall be cleared.

5.1.82.3 transition-map

Result = *Data*, ErrorFlags = *Cleared*

- DataReady = Yes

Result = *MinusOne*, ErrorFlags = *Cleared*

- DataReady = No

5.1.83 spec:/dev/grlib/req/apbuart-outbyte-polled

spec:/dev/grlib/req/apbuart-outbyte-polled

The apbuart_outbyte_polled directive shall wait until the transmitter FIFO is empty and then write the characters specified by ch as an unsigned 8-bit value to the data register of the register block specified by regs.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBAPBUART - apbuart_outbyte_polled

5.1.84 spec:/dev/grlib/req/apbuart-outbyte-wait

spec:/dev/grlib/req/apbuart-outbyte-wait

The apbuart_outbyte_wait directive shall wait until the transmitter FIFO is empty indicated by the status register of the register block specified by regs.

rationale: N/A

functional-type: function

Traced design component: RTEMSDeviceGRLIBAPBUART - apbuart_outbyte_wait

5.1.85 spec:/dev/grlib/req/irqamp-get-timestamp

spec:/dev/grlib/req/irqamp-get-timestamp

When the irqamp_get_timestamp_registers() (see: spec:/dev/grlib/if/irqamp-get-timestamp) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSDeviceGRLIBIRQAMP - irqamp_get_timestamp_registers

5.1.85.1 pre-conditions

NumberOfTimestampRegisterSets

Zero While the number of timestamp register sets indicated by the IRQ(A)MP register block specified by the irqamp_regs parameter is zero.

Positive While the number of timestamp register sets indicated by the IRQ(A)MP register block specified by the irqamp_regs parameter is positive.

5.1.85.2 post-conditions

Result

Registers The return value of irqamp_get_timestamp_registers shall be address of the timestamping register block contained in the IRQ(A)MP register block specified by the irqamp_regs parameter.

Null The return value of irqamp_get_timestamp_registers shall be false.

5.1.85.3 transition-map

Result = *Null*

- NumberOfTimestampRegisterSets = *Zero*

Result = *Registers*

- NumberOfTimestampRegisterSets = *Positive*

5.1.86 spec:/newlib/req/futex-wait

spec:/newlib/req/futex-wait

When the _Futex_Wait() (see: spec:/newlib/if/futex-wait) directive is called.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- spec:/score/tq/req/enqueue-fifo

This requirement refines spec:/newlib/req/futex.

Traced design component: futex_8c - _Futex_Wait

5.1.86.1 pre-conditions

State

Equal While the expected futex state value is equal to the actual futex state value.

NotEqual While the expected futex state value is not equal to the actual futex state value.

5.1.86.2 post-conditions

Result

Zero The return status of `_Futex_Wait()` shall be zero.

EAGAIN The return status of `_Futex_Wait()` shall be EAGAIN.

Enqueue

No The calling thread shall not be enqueued on the thread queue of the futex object.

Yes The calling thread shall be enqueued in FIFO order on the thread queue of the futex object.

5.1.86.3 transition-map

Result = *Zero*, Enqueue = *Yes*

- State = *Equal*

Result = *EAGAIN*, Enqueue = *No*

- State = *NotEqual*

5.1.87 spec:/newlib/req/futex-wake

spec:/newlib/req/futex-wake

When the `_Futex_Wake()` (see: `spec:/newlib/if/futex-wake`) directive is called.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- `spec:/score/tq/req/flush-fifo`
- `spec:/score/tq/req/flush-filter-stop`

This requirement refines `spec:/newlib/req/futex`.

Traced design component: futex_8c - _Futex_Wake

5.1.87.1 pre-conditions

Count

Negative While the count parameter is less than zero.

Partial While the count parameter is greater than or equal to zero, while the count parameter is less than the count of threads enqueued on the thread queue of the futex object.

All While the count parameter is greater than or equal to zero, while the count parameter is greater than or equal to the count of threads enqueued on the thread queue of the futex object.

5.1.87.2 post-conditions

Result

Count The return status of _Futex_Wake() shall be the count of threads extracted from the thread queue of the futex object.

Flush

No No thread shall be extracted from the thread queue of the futex object.

Partial The first count threads specified by the count parameter shall be extracted from the thread queue of the futex object in FIFO order.

All All threads shall be extracted from the thread queue of the futex object in FIFO order.

5.1.87.3 transition-map

Result = *Count*, Flush = *No*

- Count = *Negative*

Result = *Count*, Flush = *Partial*

- Count = *Partial*

Result = *Count*, Flush = *All*

- Count = *All*

5.1.88 spec:/newlib/req/sys-lock-mutex-acquire

spec:/newlib/req/sys-lock-mutex-acquire

The _Mutex_Acquire() directive shall seize the mutex as specified by /score mtx/req/seize-wait.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/mtx/req/seize-wait*

This requirement refines *spec:/newlib/req/sys-lock*.

This requirement is refined by the following requirements:

- *spec:/newlib/req/sys-lock-mutex-omip*
- *spec:/newlib/req/sys-lock-mutex-recursive-fatal*

Traced design component: None

5.1.89 spec:/newlib/req/sys-lock-mutex-acquire-timed

spec:/newlib/req/sys-lock-mutex-acquire-timed

The _Mutex_Acquire_timed() directive shall seize the mutex as specified by /score/mtx/req/seize-wait.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/mtx/req/seize-wait*

This requirement refines *spec:/newlib/req/sys-lock*.

This requirement is refined by the following requirements:

- *spec:/newlib/req/sys-lock-mutex-clock-realtime*
- *spec:/newlib/req/sys-lock-mutex-invalid-timeout*
- *spec:/newlib/req/sys-lock-mutex-omip*
- *spec:/newlib/req/sys-lock-mutex-recursive-fatal*

Traced design component: None

5.1.90 spec:/newlib/req/sys-lock-mutex-clock-realtime

spec:/newlib/req/sys-lock-mutex-clock-realtime

While the timeout parameter specifies a valid absolute CLOCK_REALTIME time point, while the mutex is not available at the specified time point, the directive shall return with a status code of ETIMEDOUT.

rationale: N/A

functional-type: function

This requirement refines [spec:/newlib/req/sys-lock-mutex-acquire-timed](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed](#).

5.1.91 spec:/newlib/req/sys-lock-mutex-invalid-timeout

spec:/newlib/req/sys-lock-mutex-invalid-timeout

While the mutex is not available, while the timeout parameter does not specify a valid absolute CLOCK_REALTIME time point, if an attempt to acquire the mutex is made, then the directive shall immediately return with a status code of EINVAL.

rationale: N/A

functional-type: function

This requirement refines [spec:/newlib/req/sys-lock-mutex-acquire-timed](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed](#).

5.1.92 spec:/newlib/req/sys-lock-mutex-omip

spec:/newlib/req/sys-lock-mutex-omip

The mutex operations shall use the OMIP locking protocol.

rationale: N/A

functional-type: function

This requirement refines [spec:/newlib/req/sys-lock-mutex-acquire](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-acquire-timed](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-release](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-recursive-acquire](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-recursive-release](#).

5.1.93 spec:/newlib/req/sys-lock-mutex-recursive-acquire

spec:/newlib/req/sys-lock-mutex-recursive-acquire

The _Mutex_recursive_Acquire() directive shall seize the mutex as specified by /score mtx/req/seize-wait.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- [spec:/score/mtx/req/seize-wait](#)

This requirement refines [spec:/newlib/req/sys-lock](#).

This requirement is refined by the following requirements:

- [spec:/newlib/req/sys-lock-mutex-omip](#)
- [spec:/newlib/req/sys-lock-mutex-recursive-allowed](#)

Traced design component: None

5.1.94 spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed

spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed

The _Mutex_recursive_Acquire_timed() directive shall seize the mutex as specified by /score mtx/req/seize-wait.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- [spec:/score/mtx/req/seize-wait](#)

This requirement refines [spec:/newlib/req/sys-lock](#).

This requirement is refined by the following requirements:

- [spec:/newlib/req/sys-lock-mutex-clock-realtime](#)
- [spec:/newlib/req/sys-lock-mutex-invalid-timeout](#)
- [spec:/newlib/req/sys-lock-mutex-omip](#)

- *spec:/newlib/req/sys-lock-mutex-recursive-allowed*

Traced design component: None

5.1.95 spec:/newlib/req/sys-lock-mutex-recursive-allowed

spec:/newlib/req/sys-lock-mutex-recursive-allowed

The mutex may be acquired recursively.

rationale: N/A

functional-type: function

This requirement refines *spec:/newlib/req/sys-lock-mutex-recursive-acquire*.

This requirement refines *spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed*.

This requirement refines *spec:/newlib/req/sys-lock-mutex-recursive-release*.

This requirement refines *spec:/newlib/req/sys-lock-mutex-recursive-try-acquire*.

5.1.96 spec:/newlib/req/sys-lock-mutex-recursive-busy

spec:/newlib/req/sys-lock-mutex-recursive-busy

When an attempt to recursively acquire the mutex is made, the directive shall immediately return with a status code of EBUSY.

rationale: N/A

functional-type: function

This requirement refines *spec:/newlib/req/sys-lock-mutex-try-acquire*.

5.1.97 spec:/newlib/req/sys-lock-mutex-recursive-fatal

spec:/newlib/req/sys-lock-mutex-recursive-fatal

When an attempt to recursively acquire the mutex is made, the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_THREAD_QUEUE_DEADLOCK fatal code.

rationale: N/A

functional-type: function

This requirement refines [spec:/newlib/req/sys-lock-mutex-acquire](#).

This requirement refines [spec:/newlib/req/sys-lock-mutex-acquire-timed](#).

5.1.98 spec:/newlib/req/sys-lock-mutex-recursive-release

spec:/newlib/req/sys-lock-mutex-recursive-release

The _Mutex_Release() directive shall surrender the mutex as specified by /score mtx/req/surrender.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- [spec:/score/mtx/req/surrender](#)

This requirement refines [spec:/newlib/req/sys-lock](#).

This requirement is refined by the following requirements:

- [spec:/newlib/req/sys-lock-mutex-omip](#)
- [spec:/newlib/req/sys-lock-mutex-recursive-allowed](#)

Traced design component: None

5.1.99 spec:/newlib/req/sys-lock-mutex-recursive-try-acquire

spec:/newlib/req/sys-lock-mutex-recursive-try-acquire

The _Mutex_recursive_Try_acquire() directive shall seize the mutex as specified by /score mtx/req/seize-try.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- [spec:/score/mtx/req/seize-try](#)

This requirement refines [spec:/newlib/req/sys-lock](#).

This requirement is refined by the following requirements:

- [spec:/newlib/req/sys-lock-mutex-recursive-allowed](#)

Traced design component: None

5.1.100 spec:/newlib/req/sys-lock-mutex-release

spec:/newlib/req/sys-lock-mutex-release

The `_Mutex_Release()` directive shall surrender the mutex as specified by `/score mtx/req/surrender`.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- `spec:/score/mtx/req/surrender`

This requirement refines `spec:/newlib/req/sys-lock`.

This requirement is refined by the following requirements:

- `spec:/newlib/req/sys-lock-mutex-omip`

Traced design component: None

5.1.101 spec:/newlib/req/sys-lock-mutex-try-acquire

spec:/newlib/req/sys-lock-mutex-try-acquire

The `_Mutex_Try_acquire()` directive shall seize the mutex as specified by `/score/mtx/req/seize-try`.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- `spec:/score/mtx/req/seize-try`

This requirement refines `spec:/newlib/req/sys-lock`.

This requirement is refined by the following requirements:

- `spec:/newlib/req/sys-lock-mutex-recursive-busy`

Traced design component: None

5.1.102 spec:/req/fatal-error

spec:/req/fatal-error

If the system detects a fatal error, then it shall terminate the system.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/terminate*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/fatal-init-task-construct-failed*
- *spec:/acfg/req/fatal-too-large-tls-size*
- *spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-boot*
- *spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-secondary*
- *spec:/bsp/sparc/leon3/req/fatal-clock-initialization*
- *spec:/score/interr/req/terminate*
- *spec:/score/object/req/fatal-allocator-mutex-deadlock*
- *spec:/score/smp/req/fatal-boot-processor-not-assigned-to-scheduler*
- *spec:/score/smp/req/fatal-mandatory-processor-not-present*
- *spec:/score/smp/req/fatal-multitasking-start-on-invalid-processor*
- *spec:/score/smp/req/fatal-multitasking-start-on-unassigned-processor*
- *spec:/score/smp/req/fatal-scheduler-requires-exactly-one-processor*
- *spec:/score/smp/req/fatal-shutdown-response*
- *spec:/score/smp/req/fatal-start-of-mandatory-processor-failed*
- *spec:/score/smp/req/fatal-start-on-not-online-processor*
- *spec:/score/smp/req/fatal-wrong-cpu-state-to-perform-jobs*
- *spec:/score/thread/req/fatal-bad-thread-dispatch-disable-level*
- *spec:/score/thread/req/fatal-bad-thread-dispatch-environment-ipi*
- *spec:/score/thread/req/fatal-thread-exitted*
- *spec:/score/tq/req/fatal-enqueue-sticky-from-bad-state*

5.1.103 spec:/rtems/barrier/req/create

spec:/rtems/barrier/req/create

When the rtems_barrier_create() (see: spec:/rtems/barrier/if/create) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicBarrier - rtems_barrier_create

5.1.103.1 pre-conditions

Name

Valid While the name parameter is valid.

Invalid While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

Class

Default While the attribute_set parameter specifies the default class.

Manual While the attribute_set parameter specifies the manual release class.

Auto While the attribute_set parameter specifies the automatic release class.

MaxWait

Zero While the maximum_waiters parameter is zero.

Positive While the maximum_waiters parameter is positive.

Free

Yes While the system has at least one inactive barrier object available.

No While the system has no inactive barrier object available.

5.1.103.2 post-conditions

Status

Ok The return status of rtems_barrier_create shall be RTEMS_SUCCESSFUL.

InvName The return status of rtems_barrier_create shall be RTEMS_INVALID_NAME.

InvAddr The return status of rtems_barrier_create shall be RTEMS_INVALID_ADDRESS.

InvNum The return status of rtems_barrier_create shall be RTEMS_INVALID_NUMBER.

TooMany The return status of rtems_barrier_create shall be RTEMS_TOO_MANY.

Name

Valid The unique object name shall identify the barrier created by the rtems_barrier_create call.

Invalid The unique object name shall not identify a barrier.

Class

Manual The class of the barrier created by the rtems_barrier_create call shall be manual release.

Auto The class of the barrier created by the rtems_barrier_create call shall be automatic release.

IdVar

Set The value of the object referenced by the id parameter shall be set to the object identifier of the created barrier after the return of the rtems_barrier_create call.

Nop Objects referenced by the id parameter in past calls to rtems_barrier_create shall not be accessed by the rtems_barrier_create call.

5.1.103.3 transition-map

Status = *Ok*, Name = *Valid*, Class = *Auto*, IdVar = *Set*

- Name = *Valid*, Id = *Valid*, Class = *Auto*, MaxWait = *Positive*, Free = *Yes*

Status = *InvNum*, Name = *Invalid*, Class = NA, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Class = *Auto*, MaxWait = *Zero*, Free = { *Yes*, *No* }

Status = *Ok*, Name = *Valid*, Class = *Manual*, IdVar = *Set*

- Name = *Valid*, Id = *Valid*, Class = { *Default*, *Manual* }, MaxWait = { *Zero*, *Positive* }, Free = *Yes*

Status = *TooMany*, Name = *Invalid*, Class = NA, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Class = { *Default*, *Manual* }, MaxWait = { *Zero*, *Positive* }, Free = *No*
- Name = *Valid*, Id = *Valid*, Class = *Auto*, MaxWait = *Positive*, Free = *No*

Status = *InvAddr*, Name = *Invalid*, Class = NA, IdVar = *Nop*

- Name = *Valid*, Id = *Null*, Class = { *Default*, *Manual*, *Auto* }, MaxWait = { *Zero*, *Positive* }, Free = { *Yes*, *No* }

Status = *InvName*, Name = *Invalid*, Class = NA, IdVar = *Nop*

- Name = *Invalid*, Id = { *Valid*, *Null* }, Class = { *Default*, *Manual*, *Auto* }, MaxWait = { *Zero*, *Positive* }, Free = { *Yes*, *No* }

5.1.104 spec:/rtems/barrier/req/delete

spec:/rtems/barrier/req/delete

When the rtems_barrier_delete() (see: [spec:/rtems/barrier/if/delete](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicBarrier - rtems_barrier_delete

5.1.104.1 pre-conditions

Id

NoObj While the id parameter is not associated with a barrier.

Barrier While the id parameter is associated with a barrier.

5.1.104.2 post-conditions

Status

Ok The return status of rtems_barrier_delete shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_barrier_delete shall be RTEMS_INVALID_ID.

Name

Valid The unique object name shall identify a barrier.

Invalid The unique object name shall not identify a barrier.

Flush

Yes Tasks waiting at the barrier shall be unblocked.

No Tasks waiting at the barrier shall remain blocked.

5.1.104.3 transition-map

Status = *InvId*, Name = *Valid*, Flush = *No*

- Id = *NoObj*

Status = *Ok*, Name = *Invalid*, Flush = *Yes*

- Id = *Barrier*

5.1.105 spec:/rtems/barrier/req/ident

spec:/rtems/barrier/req/ident

The rtems_barrier_ident directive shall identify an Classic API barrier class object by its name as specified by /rtems/req/ident-local.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicBarrier - rtems_barrier_ident

5.1.106 spec:/rtems/barrier/req/release

spec:/rtems/barrier/req/release

When the rtems_barrier_release() (see: [spec:/rtems/barrier/if/release](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicBarrier - rtems_barrier_release

5.1.106.1 pre-conditions

Id

NoObj While the id parameter is not associated with a barrier.

Manual While the id parameter is associated with a manual release barrier.

Auto While the id parameter is associated with an automatic release barrier.

Released

Valid While the released parameter references an object of type uint32_t.

Null While the released parameter is NULL.

Waiting

Zero While the number of tasks waiting at the barrier is zero.

Positive While the number of tasks waiting at the barrier is positive.

5.1.106.2 post-conditions

Status

Ok The return status of rtems_barrier_release shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_barrier_release shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_barrier_release shall be RTEMS_INVALID_ADDRESS.

ReleasedVar

Set The value of the object referenced by the released parameter shall be set to the number of released tasks after the return of the rtems_barrier_release call.

Nop Objects referenced by the released parameter in past calls to rtems_barrier_release shall not be accessed by the rtems_barrier_release call.

5.1.106.3 transition-map

Status = *InvId*, ReleasedVar = *Nop*

- Id = *NoObj*, Released = *Valid*, Waiting = NA

Status = *Ok*, ReleasedVar = *Set*

- Id = { *Manual*, *Auto* }, Released = *Valid*, Waiting = { *Zero*, *Positive* }

Status = *InvAddr*, ReleasedVar = *Nop*

- Id = { *NoObj*, *Manual*, *Auto* }, Released = *Null*, Waiting = NA

5.1.107 spec:/rtems/barrier/req/wait

spec:/rtems/barrier/req/wait

When the rtems_barrier_wait() (see: spec:/rtems/barrier/if/wait) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicBarrier - rtems_barrier_wait

Traced design component: RTEMSAPIClassicTypes - RTEMS_NO_TIMEOUT

5.1.107.1 pre-conditions

Id

NoObj While the id parameter is not associated with a barrier.

Manual While the id parameter is associated with a manual release barrier.

Auto While the id parameter is associated with an automatic release barrier.

Timeout

Ticks While the released parameter is a clock tick interval.

Forever While the released parameter is RTEMS_NO_TIMEOUT.

Satisfy

Never While the calling task waits at the barrier, while the barrier is not released or deleted.

Wait While calling the directive releases the barrier.

Release While the calling task waits at the barrier, while the barrier is released.

Delete While the calling task waits at the barrier, while the barrier is deleted.

5.1.107.2 post-conditions

Status

Ok The return status of rtems_barrier_wait shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_barrier_wait shall be RTEMS_INVALID_ID.

Timeout The return status of rtems_barrier_wait shall be RTEMS_TIMEOUT.

ObjDel The return status of rtems_barrier_wait shall be RTEMS_OBJECT_WAS_DELETED.

NoReturn The call to rtems_barrier_wait shall not return to the calling task.

5.1.107.3 skip-reasons

NoWaitRelease Manual release barriers cannot be released by calling the directive.

5.1.107.4 transition-map

Status = *Timeout*

- Id = { *Manual*, *Auto* }, Timeout = *Ticks*, Satisfy = *Never*

Status = *NoReturn*

- Id = { *Manual*, *Auto* }, Timeout = *Forever*, Satisfy = *Never*

Status = *ObjDel*

- Id = { *Manual*, *Auto* }, Timeout = { *Ticks*, *Forever* }, Satisfy = *Delete*

Status = *Ok*

- Id = *Manual*, Timeout = { *Ticks*, *Forever* }, Satisfy = *Release*
- Id = *Auto*, Timeout = { *Ticks*, *Forever* }, Satisfy = { *Wait*, *Release* }

Status = *InvId*

- Id = *NoObj*, Timeout = NA, Satisfy = NA

NoWaitRelease

- Id = *Manual*, Timeout = { *Ticks*, *Forever* }, Satisfy = *Wait*

5.1.108 spec:/rtems/basedefs/req/symbol-name

spec:/rtems/basedefs/req/symbol-name

The RTEMS_SYMBOL_NAME macro shall expand to the ELF symbol name associated with the C language identifier specified by the _name parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIBaseDefs - RTEMS_SYMBOL_NAME

5.1.109 spec:/rtems/basedefs/req/symbol-name-expand

spec:/rtems/basedefs/req/symbol-name-expand

The _name parameter value shall be expanded by the C preprocessor.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIBaseDefs - RTEMS_SYMBOL_NAME

5.1.110 spec:/rtems/cache/req/disable-data

spec:/rtems/cache/req/disable-data

Where the target has a data cache, where the data cache can be disabled, when the rtems_cache_disable_data directive is called, the data cache shall be disabled.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_disable_data

5.1.111 spec:/rtems/cache/req/disable-instruction

spec:/rtems/cache/req/disable-instruction

Where the target has an instruction cache, where the instruction cache can be disabled, when the rtems_cache_disable_instruction directive is called, the instruction cache shall be disabled.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_disable_instruction

5.1.112 spec:/rtems/cache/req/enable-data

spec:/rtems/cache/req/enable-data

Where the target has a data cache, where the data cache can be enabled, when the rtems_cache_enable_data directive is called, the data cache shall be enabled.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_enable_data

5.1.113 spec:/rtems/cache/req/enable-instruction

spec:/rtems/cache/req/enable-instruction

Where the target has an instruction cache, where the instruction cache can be enabled, when the rtems_cache_enable_instruction directive is called, the instruction cache shall be enabled.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_enable_instruction

5.1.114 spec:/rtems/cache/req/flush-entire-data

spec:/rtems/cache/req/flush-entire-data

Where the target has a data cache, where the data cache can be flushed, when the rtems_cache_flush_entire_data directive is called, the data cache shall be flushed.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_flush_entire_data

5.1.115 spec:/rtems/cache/req/flush-multiple-data-lines

spec:/rtems/cache/req/flush-multiple-data-lines

Where the target has a data cache, where data cache lines can be flushed, where the data cache is not coherent with all bus masters, when the rtems_cache_flush_multiple_data_lines directive is called, the data cache lines covering the memory area specified by begin and size shall be flushed.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_flush_multiple_data_lines

5.1.116 spec:/rtems/cache/req/freeze-data

spec:/rtems/cache/req/freeze-data

Where the target has a data cache, where the data cache can be frozen, when the rtems_cache_freeze_data directive is called, the data cache shall be frozen.

rationale: N/A

functional-type: function

Traced design component: cache_8h - rtems_cache_freeze_data

5.1.117 spec:/rtems/cache/req/freeze-instruction

spec:/rtems/cache/req/freeze-instruction

Where the target has an instruction cache, where the instruction cache can be frozen, when the rtems_cache_freeze_instruction directive is called, the instruction cache shall be frozen.

rationale: N/A

functional-type: function

Traced design component: cache_8h - rtems_cache_freeze_instruction

5.1.118 spec:/rtems/cache/req/get-data-line-size

spec:/rtems/cache/req/get-data-line-size

Where the target has a data cache, the rtems_cache_get_data_line_size directive shall return the size of the data cache line.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_data_line_size

5.1.119 spec:/rtems/cache/req/get-data-size

spec:/rtems/cache/req/get-data-size

Where the target has a data cache, while the level parameter is greater than zero, while the level parameter is associated with a data cache level, the rtems_cache_get_data_cache_size directive shall return the size of the data cache of the level specified by level.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_data_cache_size

5.1.120 spec:/rtems/cache/req/get-data-size-level-zero

spec:/rtems/cache/req/get-data-size-level-zero

Where the target has a data cache, while the level parameter is equal to zero, the rtems_cache_get_data_cache_size directive shall return the size of the entire data cache.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_data_cache_size

5.1.121 spec:/rtems/cache/req/get-data-size-no-level

spec:/rtems/cache/req/get-data-size-no-level

Where the target has a data cache, while the level parameter is greater than zero, while the level parameter is not associated with a data cache level, the rtems_cache_get_data_cache_size directive shall return zero.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_data_cache_size

5.1.122 spec:/rtems/cache/req/get-instruction-line-size

spec:/rtems/cache/req/get-instruction-line-size

Where the target has an instruction cache, the rtems_cache_get_instruction_line_size directive shall return the size of the instruction cache line.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_instruction_line_size

5.1.123 spec:/rtems/cache/req/get-instruction-size

spec:/rtems/cache/req/get-instruction-size

Where the target has an instruction cache, while the level parameter is greater than zero, while the level parameter is associated with an instruction cache level, the rtems_cache_get_instruction_cache_size directive shall return the size of the instruction cache of the level specified by level.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_instruction_cache_size

5.1.124 spec:/rtems/cache/req/get-instruction-size-level-zero

spec:/rtems/cache/req/get-instruction-size-level-zero

Where the target has an instruction cache, while the level parameter is equal to zero, the rtems_cache_get_instruction_cache_size directive shall return the size of the entire instruction cache.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_instruction_cache_size

5.1.125 spec:/rtems/cache/req/get-instruction-size-no-level

spec:/rtems/cache/req/get-instruction-size-no-level

Where the target has an instruction cache, while the level parameter is greater than zero, while the level parameter is not associated with an instruction cache level, the rtems_cache_get_instruction_cache_size directive shall return zero.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_instruction_cache_size

5.1.126 spec:/rtems/cache/req/get-maximal-line-size

spec:/rtems/cache/req/get-maximal-line-size

Where the target has a data cache or an instruction cache, the rtems_cache_get_maximal_line_size directive shall return the maximal cache line size of all caches.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_get_maximal_line_size

5.1.127 spec:/rtems/cache/req/instruction-sync-after-code-change

spec:/rtems/cache/req/instruction-sync-after-code-change

Where the target has an instruction cache, where the instruction cache needs to be synchronized after code changes, when the rtems_cache_instruction_sync_after_code_change directive is called, the instruction cache shall be synchronized so that the code in the memory area specified by begin and size will be fetched when it needs to be executed.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_instruction_sync_after_code_change

5.1.128 spec:/rtems/cache/req/invalidate-entire-data

spec:/rtems/cache/req/invalidate-entire-data

Where the target has a data cache, where the data cache can be flushed and invalidated, when the rtems_cache_invalidate_entire_data directive is called, the data cache shall be flushed and invalidated.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_invalidate_entire_data

5.1.129 spec:/rtems/cache/req/invalidate-entire-instruction

spec:/rtems/cache/req/invalidate-entire-instruction

Where the target has an instruction cache, where the instruction cache can be invalidated, when the rtems_cache_invalidate_entire_instruction directive is called, the instruction cache shall be invalidated.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_invalidate_entire_instruction

5.1.130 spec:/rtems/cache/req/invalidate-multiple-data-lines

spec:/rtems/cache/req/invalidate-multiple-data-lines

Where the target has a data cache, where data cache lines can be invalidated, where the data cache is not coherent with all bus masters, when the rtems_cache_invalidate_multiple_data_lines directive is called, the data cache lines covering the memory area specified by begin and size shall be invalidated.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_invalidate_multiple_data_lines

5.1.131 spec:/rtems/cache/req/invalidate-multiple-instruction-lines

spec:/rtems/cache/req/invalidate-multiple-instruction-lines

Where the target has an instruction cache, where instruction cache lines can be invalidated, where the instruction cache is not coherent with all bus masters, when the rtems_cache_invalidate_multiple_instruction_lines directive is called, the instruction cache lines covering the memory area specified by begin and size shall be invalidated.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicCache - rtems_cache_invalidate_multiple_instruction_lines

5.1.132 spec:/rtems/cache/req/unfreeze-data

spec:/rtems/cache/req/unfreeze-data

Where the target has a data cache, where the data cache can be frozen, when the rtems_cache_unfreeze_data directive is called, the data cache shall be not frozen.

rationale: N/A

functional-type: function

Traced design component: cache_8h - rtems_cache_unfreeze_data

5.1.133 spec:/rtems/cache/req/unfreeze-instruction

spec:/rtems/cache/req/unfreeze-instruction

Where the target has an instruction cache, where the instruction cache can be frozen, when the rtems_cache_unfreeze_instruction directive is called, the instruction cache shall be not frozen.

rationale: N/A

functional-type: function

Traced design component: cache_8h - rtems_cache_unfreeze_instruction

5.1.134 spec:/rtems/clock/req/get-boot-time

spec:/rtems/clock/req/get-boot-time

The directive shall return a time point during system initialization which is used by CLOCK_REALTIME.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- spec:/score/timecounter/req/get-coarse

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_boot_time

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_boot_time_bintime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_boot_time_timeval

5.1.135 spec:/rtems/clock/req/get-monotonic

spec:/rtems/clock/req/get-monotonic

The directive shall return the time elapsed since some fixed time point in the past measured using the CLOCK_MONOTONIC at some time point during the directive call.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/timecounter/req/get*

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic_bintime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic_sbintime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic_timeval

5.1.136 spec:/rtems/clock/req/get-monotonic-coarse

spec:/rtems/clock/req/get-monotonic-coarse

The directive shall return the time elapsed since some fixed time point in the past measured using the CLOCK_MONOTONIC at some time point close to the directive call.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/timecounter/req/get-coarse*

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic_coarse

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic_coarse_bintime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_monotonic_coarse_timeval

5.1.137 spec:/rtems/clock/req/get-realtime

spec:/rtems/clock/req/get-realtime

The directive shall return the time elapsed since the Unix epoch measured using the CLOCK_REALTIME at some time point during the directive call.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/timecounter/req/get*

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_realtime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_realtime_bintime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_realtime_timeval

5.1.138 spec:/rtems/clock/req/get-realtime-coarse

spec:/rtems/clock/req/get-realtime-coarse

The directive shall return the time elapsed since the Unix epoch measured using the CLOCK_REALTIME at some time point close to the directive call.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/timecounter/req/get-coarse*

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_realtime_coarse

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_realtime_coarse_bintime

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_realtime_coarse_timeval

5.1.139 spec:/rtems/clock/req/get-tod

spec:/rtems/clock/req/get-tod

When the rtems_clock_get_tod() (see: [spec:/rtems/clock/if/get-tod](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_tod

5.1.139.1 pre-conditions

ToD

Arbitrary While the CLOCK_REALTIME indicates an arbitrary valid date and time between 1988-01-01T00:00:00.000000000Z and 2514-05-30T01:53:03.999999999Z.

Leap4 While the CLOCK_REALTIME indicates a date for a leap year with the value of 29th of February.

Leap400 While the CLOCK_REALTIME indicates a date for a leap year with the value of 29th of February.

Youngest While the CLOCK_REALTIME indicates the youngest date and time accepted (1988-01-01T00:00:00.000000000Z).

Oldest While the CLOCK_REALTIME indicates the oldest date and time accepted (2099-12-31T23:59:59.999999999Z).

NotSet While the CLOCK_REALTIME has not been set before.

Param

Valid While the time_of_day parameter references an object of type rtems_time_of_day.

Null While the time_of_day parameter is NULL.

5.1.139.2 post-conditions

Status

Ok The return status of rtems_clock_get_tod shall be RTEMS_SUCCESSFUL

InvAddr The return status of rtems_clock_get_tod shall be RTEMS_INVALID_ADDRESS.

NotDef The return status of rtems_clock_get_tod shall be RTEMS_NOT_DEFINED.

Value

TimeOfDay The value of the object referenced by the time_of_day parameter shall be set to the value of the CLOCK_REALTIME at a point in time during the call to rtems_clock_get_tod.

Unchanged Object referenced by the time_of_day parameter in past call to rtems_clock_get_tod shall not be modified by the rtems_clock_get_tod call.

5.1.139.3 transition-map

Status = *NotDef*, Value = *Unchanged*

- ToD = *NotSet*, Param = *Valid*

Status = *Ok*, Value = *TimeOfDay*

- ToD = { *Arbitrary*, *Leap4*, *Leap400*, *Youngest*, *Oldest* }, Param = *Valid*

Status = *InvAddr*, Value = *Unchanged*

- ToD = { *Arbitrary*, *Leap4*, *Leap400*, *Youngest*, *Oldest*, *NotSet* }, Param = *Null*

5.1.140 spec:/rtems/clock/req/get-uptime

spec:/rtems/clock/req/get-uptime

When the rtems_clock_get_uptime() (see: spec:/rtems/clock/if/get-uptime) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicClock - rtems_clock_get_uptime

5.1.140.1 pre-conditions

Uptime

Valid While the uptime parameter references an object of type struct timespec.

Null While the uptime parameter is NULL.

5.1.140.2 post-conditions

Status

Ok The return status of rtems_clock_get_uptime shall be RTEMS_SUCCESSFUL

InvAddr The return status of rtems_clock_get_uptime shall be RTEMS_INVALID_ADDRESS.

Uptime

Set The value of the object referenced by the uptime parameter shall be set to seconds and nanoseconds elapsed since a point in time during the system initialization and a point in time during the call of rtems_clock_get_uptime using CLOCK_MONOTONIC as result of the rtems_clock_get_uptime call.

Unchanged Objects referenced by the uptime parameter in calls to rtems_clock_get_uptime shall not be modified by the rtems_clock_get_uptime call.

5.1.140.3 transition-map

Status = *Ok*, Uptime = *Set*

- Uptime = *Valid*

Status = *InvAddr*, Uptime = *Unchanged*

- Uptime = *Null*

5.1.141 spec:/rtems/clock/req/set

spec:/rtems/clock/req/set

When the rtems_clock_set() (see: [spec:/rtems/clock/if/set](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicClock - rtems_clock_set

5.1.141.1 pre-conditions

ToD

Valid While the time_of_day parameter references an arbitrary valid date and time between 1988-01-01T00:00:00.000000000Z and 2105-12-31T23:59:59.999999999Z.

ValidLeap4 While the time_of_day parameter references a date for a leap year with the value of 29th of February.

ValidLeap400 While the time_of_day parameter references a date for a leap year with the value of 29th of February.

Youngest While the time_of_day parameter references the youngest date and time accepted (1988-01-01T00:00:00.000000000Z).

Oldest While the time_of_day parameter references the oldest date and time accepted (2099-12-31T23:59:59.999999999Z).

TooJung While the time_of_day parameter references a valid date and time younger than 1988-01-01T00:00:00.000000000Z.

TooOld While the time_of_day parameter references a valid date and time older than 2105-12-31T23:59:59.999999999Z.

InvMonth0 While the time_of_day parameter is invalid because the value of the month is 0.

InvMonth While the time_of_day parameter is invalid because the value of the month is larger than 12.

InvDay0 While the time_of_day parameter is invalid because the value of the day is 0.

InvDay While the time_of_day parameter is invalid because the value of the day is larger than the days of the month.

InvHour While the time_of_day parameter is invalid because the value of the hour is larger than 23.

InvMinute While the time_of_day parameter is invalid because the value of the minute is larger than 59.

InvSecond While the time_of_day parameter is invalid because the value of the second is larger than 59.

InvTicks While the time_of_day parameter is invalid because the value of the ticks are larger or equal to the ticks per second.

InvLeap4 While the time_of_day parameter is invalid because the value 30th of February does not exist in a leap year.

InvLeap100 While the time_of_day parameter is invalid because the value 29th of February does not exist in a non-leap year.

InvLeap400 While the time_of_day parameter is invalid because the value 30th of February does not exist in a leap year.

AtTimer While the time_of_day parameter references the same point in time when a timer should fire.

BeforeTimer While the time_of_day parameter references a point in time before a timer should fire.

AfterTimer While the time_of_day parameter references a point in time after a timer should fire.

Null While the time_of_day parameter is NULL.

Hook

None While no TOD hook is registered.

Ok While all TOD hooks invoked by the rtems_clock_set call return a status code equal to STATUS_SUCCESSFUL.

NotOk While at least one TOD hook invoked by the rtems_clock_set call returns a status code not equal to STATUS_SUCCESSFUL.

5.1.141.2 post-conditions

Status

Ok The return status of rtems_clock_set shall be RTEMS_SUCCESSFUL

InvAddr The return status of rtems_clock_set shall be RTEMS_INVALID_ADDRESS.

InvClk The return status of rtems_clock_set shall be RTEMS_INVALID_CLOCK.

Hook The return status of rtems_clock_set shall be derived from the status returned by the TOD hook.

Clock

Set The CLOCK_REALTIME shall be set to the values of the object referenced by the time_of_day parameter during the rtems_clock_set call.

Nop The state of the CLOCK_REALTIME shall not be changed by the rtems_clock_set call.

Timer

Triggered The timer routine shall be executed once after the CLOCK_REALTIME has been set and before the execution of the rtems_clock_set call terminates.

Nop The timer routine shall not be invoked during the rtems_clock_set call.

5.1.141.3 transition-map

Status = *InvAddr*, Clock = *Nop*, Timer = *Nop*

- ToD = *Null*, Hook = { *None*, *Ok*, *NotOk* }

Status = *Ok*, Clock = *Set*, Timer = *Triggered*

- ToD = { *AtTimer*, *AfterTimer* }, Hook = { *None*, *Ok* }

Status = *Hook*, Clock = *Nop*, Timer = *Nop*

- ToD = { *Valid*, *ValidLeap4*, *ValidLeap400*, *Youngest*, *Oldest*, *AtTimer*, *BeforeTimer*, *AfterTimer* }, Hook = *NotOk*

Status = *Ok*, Clock = *Set*, Timer = *Nop*

- ToD = { *Valid*, *ValidLeap4*, *ValidLeap400*, *Youngest*, *Oldest*, *BeforeTimer* }, Hook = { *None*, *Ok* }

Status = *InvClk*, Clock = *Nop*, Timer = *Nop*

- ToD = { *TooJung*, *TooOld*, *InvMonth0*, *InvMonth*, *InvDay0*, *InvDay*, *InvHour*, *InvMinute*, *InvSecond*, *InvTicks*, *InvLeap4*, *InvLeap100*, *InvLeap400* }, Hook = { *None*, *Ok*, *NotOk* }

5.1.142 spec:/rtems/config/req/get-idle-task

spec:/rtems/config/req/get-idle-task

The rtems_configuration_get_idle_task directive shall return the address of task body executed by each IDLE task.

rationale: The directive may be used to check the setting of the CONFIGURE_IDLE_TASK_BODY application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_idle_task

5.1.143 spec:/rtems/config/req/get-idle-task-stack-size

spec:/rtems/config/req/get-idle-task-stack-size

The rtems_configuration_get_idle_task_stack_size directive shall return the task stack size of each IDLE task.

rationale: The directive may be used to check the setting of the CONFIGURE_IDLE_TASK_STACK_SIZE application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_idle_task_stack_size

5.1.144 spec:/rtems/config/req/get-interrupt-stack-size

spec:/rtems/config/req/get-interrupt-stack-size

The rtems_configuration_get_interrupt_stack_size directive shall return the interrupt stack size.

rationale: The directive may be used to check the setting of the CONFIGURE_INTERRUPT_STACK_SIZE application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_interrupt_stack_size

5.1.145 spec:/rtems/config/req/get-maximum-barriers

spec:/rtems/config/req/get-maximum-barriers

The rtems_configuration_get_maximum_barriers directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicBarrier that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_BARRIERS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_barriers

5.1.146 spec:/rtems/config/req/get-maximum-extensions

spec:/rtems/config/req/get-maximum-extensions

The rtems_configuration_get_maximum_extensions directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicUserExt that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_USER_EXTENSIONS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_extensions

5.1.147 spec:/rtems/config/req/get-maximum-message-queues

spec:/rtems/config/req/get-maximum-message-queues

The rtems_configuration_get_maximum_message_queues directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicMessage that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_MESSAGE_QUEUES application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_message_queues

5.1.148 spec:/rtems/config/req/get-maximum-partitions

spec:/rtems/config/req/get-maximum-partitions

The rtems_configuration_get_maximum_partitions directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicPart that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_PARTITIONS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_partitions

5.1.149 spec:/rtems/config/req/get-maximum-periods

spec:/rtems/config/req/get-maximum-periods

The rtems_configuration_get_maximum_periods directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicRatemon that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_PERIODS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_periods

5.1.150 spec:/rtems/config/req/get-maximum-ports

spec:/rtems/config/req/get-maximum-ports

The rtems_configuration_get_maximum_ports directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicDPMem that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_PORTS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_ports

5.1.151 spec:/rtems/config/req/get-maximum-processors

spec:/rtems/config/req/get-maximum-processors

The rtems_configuration_get_maximum_processors directive shall return the maximum number of processors which can be managed by the system.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_PROCESSORS application configuration option. The maximum number of processors which are actually available for use by the application is returned by rtems_scheduler_get_processor_maximum.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_processors

5.1.152 spec:/rtems/config/req/get-maximum-regions

spec:/rtems/config/req/get-maximum-regions

The rtems_configuration_get_maximum_regions directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicRegion that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_REGIONS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_regions

5.1.153 spec:/rtems/config/req/get-maximum-semaphores

spec:/rtems/config/req/get-maximum-semaphores

The rtems_configuration_get_maximum_semaphores directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicSem that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_SEMAPHORES application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_semaphores

5.1.154 spec:/rtems/config/req/get-maximum-tasks

spec:/rtems/config/req/get-maximum-tasks

The rtems_configuration_get_maximum_tasks directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicTasks that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_TASKS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_tasks

5.1.155 spec:/rtems/config/req/get-maximum-timers

spec:/rtems/config/req/get-maximum-timers

The rtems_configuration_get_maximum_timers directive shall return the maximum number of objects managed by the @ref RTEMSAPIClassicTimer that can be concurrently active.

rationale: The directive may be used to check the setting of the CONFIGURE_MAXIMUM_TIMERS application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_maximum_timers

5.1.156 spec:/rtems/config/req/get-microseconds-per-tick

spec:/rtems/config/req/get-microseconds-per-tick

The rtems_configuration_get_microseconds_per_tick directive shall return the clock tick interval in microseconds.

rationale: The directive may be used to check the setting of the CONFIGURE_MICROSECONDS_PER_TICK application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_microseconds_per_tick

5.1.157 spec:/rtems/config/req/get-milliseconds-per-tick

spec:/rtems/config/req/get-milliseconds-per-tick

The rtems_configuration_get_milliseconds_per_tick directive shall return the clock tick interval in milliseconds.

rationale: The directive may be used to check the setting of the CONFIGURE_MICROSECONDS_PER_TICK application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_milliseconds_per_tick

5.1.158 spec:/rtems/config/req/get-nanoseconds-per-tick

spec:/rtems/config/req/get-nanoseconds-per-tick

The rtems_configuration_get_nanoseconds_per_tick directive shall return the clock tick interval in nanoseconds.

rationale: The directive may be used to check the setting of the CONFIGURE_MICROSECONDS_PER_TICK application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_nanoseconds_per_tick

5.1.159 spec:/rtems/config/req/get-stack-allocate-for-idle-hook

spec:/rtems/config/req/get-stack-allocate-for-idle-hook

The rtems_configuration_get_stack_allocate_for_idle_hook directive shall return the stack allocator allocate handler used to allocate the task stack of each IDLE task.

rationale: The directive may be used to check the setting of the CONFIGURE_TASK_STACK_ALLOCATOR_FOR_IDLE application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_stack_allocate_for_idle_hook

5.1.160 spec:/rtems/config/req/get-stack-allocate-hook

spec:/rtems/config/req/get-stack-allocate-hook

The rtems_configuration_get_stack_allocate_hook directive shall return the address of the task stack allocator which is used to allocate the stack for a task which is not an IDLE task.

rationale: The directive may be used to check the setting of the CONFIGURE_TASK_STACK_ALLOCATOR application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_stack_allocate_hook

5.1.161 spec:/rtems/config/req/get-stack-allocator-avoids-work-space

spec:/rtems/config/req/get-stack-allocator-avoids-work-space

The rtems_configuration_get_stack_allocator_avoids_work_space directive shall return the option value configured by CONFIGURE_TASK_STACK_ALLOCATOR_AVOIDS_WORK_SPACE.

rationale: The directive may be used to check the setting of the CONFIGURE_TASK_STACK_ALLOCATOR_AVOIDS_WORK_SPACE application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_stack_allocator_avoids_work_space

5.1.162 spec:/rtems/config/req/get-stack-free-hook

spec:/rtems/config/req/get-stack-free-hook

The rtems_configuration_get_stack_free_hook directive shall return the address of the task stack deallocator which is used to free the stack of a task which was allocated by the task stack allocator.

rationale: The directive may be used to check the setting of the CONFIGURE_TASK_STACK_DEALLOCATOR application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_stack_free_hook

5.1.163 spec:/rtems/config/req/get-ticks-per-timeslice

spec:/rtems/config/req/get-ticks-per-timeslice

The rtems_configuration_get_ticks_per_timeslice directive shall return the number of clock ticks per timeslice.

rationale: The directive may be used to check the setting of the CONFIGURE_TICKS_PER_TIMESLICE application configuration option.

functional-type: function

Traced design component: RTEMSAPIConfig - rtems_configuration_get_ticks_per_timeslice

5.1.164 spec:/rtems/cpuuse/req/reset

spec:/rtems/cpuuse/req/reset

When rtems_cpu_usage_reset is called, the CPU usage shall be set to zero for each thread.

rationale: N/A

functional-type: function

Traced design component: libmisc__cpuuse - rtems_cpu_usage_reset

5.1.165 spec:/rtems/event/req/all-events

spec:/rtems/event/req/all-events

The RTEMS_ALL_EVENTS constant shall be a constant expression which evaluates to a value of the bitwise or of RTEMS_EVENT_0, RTEMS_EVENT_1, RTEMS_EVENT_2, RTEMS_EVENT_3, RTEMS_EVENT_4, RTEMS_EVENT_5, RTEMS_EVENT_6, RTEMS_EVENT_7, RTEMS_EVENT_8, RTEMS_EVENT_9, RTEMS_EVENT_10, RTEMS_EVENT_11, RTEMS_EVENT_12, RTEMS_EVENT_13, RTEMS_EVENT_14, RTEMS_EVENT_15, RTEMS_EVENT_16, RTEMS_EVENT_17, RTEMS_EVENT_18, RTEMS_EVENT_19, RTEMS_EVENT_20, RTEMS_EVENT_21, RTEMS_EVENT_22, RTEMS_EVENT_23, RTEMS_EVENT_24, RTEMS_EVENT_25, RTEMS_EVENT_26, RTEMS_EVENT_27, RTEMS_EVENT_28, RTEMS_EVENT_29, RTEMS_EVENT_30, and RTEMS_EVENT_31.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicEvent - RTEMS_ALL_EVENTS

5.1.166 spec:/rtems/event/req/event-constant

spec:/rtems/event/req/event-constant

Let $J \in \{0, 1, \dots, 31\}$. The J -th event number constant shall be a constant expression which evaluates to the integer representation of event number J .

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_0

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_1

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_2

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_3

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_4

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_5

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_6

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_7

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_8

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_9

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_10

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_11

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_12

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_13

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_14

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_15

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_16

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_17

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_18

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_19

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_20

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_21

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_22

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_23

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_24

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_25

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_26

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_27

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_28

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_29

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_30

Traced design component: RTEMSAPIClassicEvent - RTEMS_EVENT_31

5.1.167 spec:/rtems/event/req/event-number

spec:/rtems/event/req/event-number

Let $J \in \{0, 1, \dots, 31\}$. The event number J shall be represented by the integer 2^J .

rationale: N/A

functional-type: function

This requirement refines *spec:/rtems/event/req/group*.

5.1.168 spec:/rtems/event/req/pending-events

spec:/rtems/event/req/pending-events

The RTEMS_PENDING_EVENTS constant shall be a constant expression which evaluates to a value of zero.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicEvent - RTEMS_PENDING_EVENTS

5.1.169 spec:/rtems/event/req/receive

spec:/rtems/event/req/receive

The rtems_event_receive directive shall receive events from the application event set of a task as specified by /rtems/event/req/send-receive.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicEvent - rtems_event_receive

5.1.170 spec:/rtems/event/req/send

spec:/rtems/event/req/send

The rtems_event_send directive shall send events to the application event set of a task as specified by /rtems/event/req/send-receive.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicEvent - rtems_event_send

5.1.171 spec:/rtems/event/req/send-receive

spec:/rtems/event/req/send-receive

When the event send and receive directive variants are called in the specified system condition.

rationale: N/A

functional-type: action

This requirement refines *spec:/rtems/event/req/group*.

Traced design component: RTEMSAPIClassicTypes - RTEMS_NO_TIMEOUT

5.1.171.1 pre-conditions

Id

InvId While the id parameter of the send directive is not associated with a task.

Task While the id parameter of the send directive is associated with a task.

Send

Zero While the event set sent is the empty.

Unrelated While the event set sent is unrelated to the event receive condition.

Any While the event set sent is contain at least one but not all events of the event receive condition.

All While the event set sent is contain all events of the event receive condition.

MixedAny While the event set sent is contain at least one but not all events of the event receive condition and at least one unrelated event.

MixedAll While the event set sent is contain all events of the event receive condition and at least one unrelated event.

ReceiverState

InvAddr While the receiver task calls the receive directive with the event set to receive parameter set to NULL.

NotWaiting While the receiver task is not waiting for events.

Poll While the receiver task polls for events.

Timeout While the receiver task waited for events with a timeout which occurred.

Lower While the receiver task is blocked waiting for events and the receiver task shall have a lower priority than the sender task.

Equal While the receiver task is blocked waiting for events and the receiver task shall have a priority equal to the sender task.

Higher While the receiver task is blocked waiting for events and the receiver task shall have a higher priority than the sender task.

Other While the receiver task is blocked waiting for events and the receiver task shall be on another scheduler instance than the sender task.

Intend While the receiver task intends to block for waiting for events.

Satisfy

All While the receiver task is interested in all input events.

Any While the receiver task is interested in any input event.

5.1.171.2 post-conditions

SendStatus

Ok The send event status shall be RTEMS_SUCCESSFUL.

InvId The send event status shall be RTEMS_INVALID_ID.

ReceiveStatus

None The receiver task shall not have pending events.

Pending The receiver task shall have all events sent pending.

Timeout The receive event status shall be RTEMS_TIMEOUT. The receiver task shall have all events sent after the timeout pending.

Satisfied The receive event status shall be RTEMS_SUCCESSFUL. The receiver task shall receive all events sent which are an element of the input events. The receiver task shall have all events sent which are not an element of the input events pending.

Unsatisfied The receive event status shall be RTEMS_UNSATISFIED. The receiver task shall have all events sent pending.

Blocked The receiver task shall remain blocked waiting for events after the directive call. The receiver task shall have all events sent pending.

InvAddr The receive event status shall be RTEMS_INVALID_ADDRESS. The receiver task shall have all events sent pending.

SenderPreemption

No When the sender task calls the directive to send the events, the sender task shall not be preempted as a result of the call.

Yes When the sender task calls the directive to send the events, the sender task shall be preempted as a result of the call.

5.1.171.3 skip-reasons

NoOtherScheduler In non-SMP configurations, there exists exactly one scheduler instance.

5.1.171.4 transition-map

SendStatus = *Ok*, ReceiveStatus = *Unsatisfied*, SenderPreemption = *No*

- Id = *Task*, Send = { *Zero*, *Unrelated* }, ReceiverState = *Poll*, Satisfy = { *All*, *Any* }
- Id = *Task*, Send = { *Any*, *MixedAny* }, ReceiverState = *Poll*, Satisfy = *All*

SendStatus = *Ok*, ReceiveStatus = *Satisfied*, SenderPreemption = *Yes*

- Id = *Task*, Send = { *Any*, *MixedAny* }, ReceiverState = *Higher*, Satisfy = *Any*
- Id = *Task*, Send = { *All*, *MixedAll* }, ReceiverState = *Higher*, Satisfy = { *All*, *Any* }

SendStatus = *Ok*, ReceiveStatus = *Timeout*, SenderPreemption = *No*

- Id = *Task*, Send = { *Zero*, *Unrelated*, *Any*, *All*, *MixedAny*, *MixedAll* }, ReceiverState = *Timeout*, Satisfy = { *All*, *Any* }

SendStatus = *Ok*, ReceiveStatus = *Pending*, SenderPreemption = *No*

- Id = *Task*, Send = { *Zero*, *Unrelated*, *Any*, *All*, *MixedAny*, *MixedAll* }, ReceiverState = { *InvAddr*, *NotWaiting* }, Satisfy = NA

SendStatus = *Ok*, ReceiveStatus = *Blocked*, SenderPreemption = *No*

- Id = *Task*, Send = { *Zero*, *Unrelated* }, ReceiverState = { *Lower*, *Equal*, *Higher*, *Other*, *Intend* }, Satisfy = { *All*, *Any* }
- Id = *Task*, Send = { *Any*, *MixedAny* }, ReceiverState = { *Lower*, *Equal*, *Higher*, *Other*, *Intend* }, Satisfy = *All*

SendStatus = *Ok*, ReceiveStatus = *Satisfied*, SenderPreemption = *No*

- Id = *Task*, Send = { *Any*, *MixedAny* }, ReceiverState = { *Poll*, *Lower*, *Equal*, *Other*, *Intend* }, Satisfy = *Any*
- Id = *Task*, Send = { *All*, *MixedAll* }, ReceiverState = { *Poll*, *Lower*, *Equal*, *Other*, *Intend* }, Satisfy = { *All*, *Any* }

SendStatus = *InvId*, ReceiveStatus = *None*, SenderPreemption = *No*

- Id = *InvId*, Send = NA, ReceiverState = NA, Satisfy = NA

5.1.172 spec:/rtems/event/req/system-receive

spec:/rtems/event/req/system-receive

The rtems_event_system_receive directive shall receive events from the system event set of a task as specified by /rtems/event/req/send-receive.

rationale: N/A

functional-type: function

Traced design component: event_8h - rtems_event_system_receive

5.1.173 spec:/rtems/event/req/system-send

spec:/rtems/event/req/system-send

The rtems_event_system_send directive shall send events to the system event set of a task as specified by /rtems/event/req/send-receive.

rationale: N/A

functional-type: function

Traced design component: event_8h - rtems_event_system_send

5.1.174 spec:/rtems/fatal/req/fatal

spec:/rtems/fatal/req/fatal

When rtems_fatal is called, the system shall terminate with a fatal source specified by the fatal_source parameter and a fatal code specified by the fatal_code parameter.

rationale: N/A

functional-type: function

Functions of this item are implemented by:

- *spec:/score/interr/req/terminate*

Traced design component: RTEMSAPIClassicFatal - rtems_fatal

5.1.175 spec:/rtems/intr/req/clear

spec:/rtems/intr/req/clear

When the rtems_interrupt_clear() (see: [spec:/rtems/intr/if/clear](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_clear

5.1.175.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

CanClear

Yes While the interrupt vector associated with the vector parameter can be cleared.

No While the interrupt vector associated with the vector parameter cannot be cleared.

5.1.175.2 post-conditions

Status

Ok The return status of rtems_interrupt_clear shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_interrupt_clear shall be RTEMS_INVALID_ID.

Unsat The return status of rtems_interrupt_clear shall be RTEMS_UNSATISFIED.

Cleared

Yes The pending state of the interrupt associated with the interrupt vector specified by vector shall be cleared for the processor executing the rtems_interrupt_clear call at some time point during the call.

No The pending state of the interrupt associated with the interrupt vector specified by vector shall not be cleared by the rtems_interrupt_clear call.

5.1.175.3 transition-map

Status = *Ok*, Cleared = *Yes*

- Vector = *Valid*, CanClear = *Yes*

Status = *Unsat*, Cleared = *No*

- Vector = *Valid*, CanClear = *No*

Status = *InvId*, Cleared = *NA*

- Vector = *Invalid*, CanClear = *NA*

5.1.176 spec:/rtems/intr/req/disable-interrupts

spec:/rtems/intr/req/disable-interrupts

When the directive is called, maskable interrupts shall be disabled on a processor which executed at least one instruction of the directive.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_local_disable

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_acquire

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_interrupt_disable

5.1.177 spec:/rtems/intr/req/entry-initialize-arg

spec:/rtems/intr/req/entry-initialize-arg

When the rtems_interrupt_entry_initialize directive is called, the arg member of the object referenced by the entry parameter shall be set to the value of the arg.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_entry_initialize

5.1.178 spec:/rtems/intr/req/entry-initialize-handler

spec:/rtems/intr/req/entry-initialize-handler

When the rtems_interrupt_entry_initialize directive is called, the handler member of the object referenced by the entry parameter shall be set to the value of the routine.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_entry_initialize

5.1.179 spec:/rtems/intr/req/entry-initialize-info

spec:/rtems/intr/req/entry-initialize-info

When the rtems_interrupt_entry_initialize directive is called, the info member of the object referenced by the entry parameter shall be set to the value of the arg.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_entry_initialize

5.1.180 spec:/rtems/intr/req/entry-initialize-next

spec:/rtems/intr/req/entry-initialize-next

When the rtems_interrupt_entry_initialize directive is called, the next member of the object referenced by the entry parameter shall be set to the value equal to NULL.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_entry_initialize

5.1.181 spec:/rtems/intr/req/entry-initializer-arg

spec:/rtems/intr/req/entry-initializer-arg

When the RTEMS_INTERRUPT_ENTRY_INITIALIZER is used to statically initialize an object of type rtems_interrupt_entry, the arg member of the object shall be initialized to the value of the _arg.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_ENTRY_INITIALIZER

5.1.182 spec:/rtems/intr/req/entry-initializer-handler

spec:/rtems/intr/req/entry-initializer-handler

When the RTEMS_INTERRUPT_ENTRY_INITIALIZER is used to statically initialize an object of type rtems_interrupt_entry, the handler member of the object shall be initialized to the value of the _routine.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_ENTRY_INITIALIZER

5.1.183 spec:/rtems/intr/req/entry-initializer-info

spec:/rtems/intr/req/entry-initializer-info

When the RTEMS_INTERRUPT_ENTRY_INITIALIZER is used to statically initialize an object of type rtems_interrupt_entry, the info member of the object shall be initialized to the value of the _info.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_ENTRY_INITIALIZER

5.1.184 spec:/rtems/intr/req/entry-initializer-next

spec:/rtems/intr/req/entry-initializer-next

When the RTEMS_INTERRUPT_ENTRY_INITIALIZER is used to statically initialize an object of type rtems_interrupt_entry, the next member of the object shall be initialized to a value equal to NULL.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_ENTRY_INITIALIZER

5.1.185 spec:/rtems/intr/req/entry-install

spec:/rtems/intr/req/entry-install

When the rtems_interrupt_entry_install() (see: [spec:/rtems/intr/if/entry-install](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_entry_install

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_REPLACE

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_SHARED

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_UNIQUE

5.1.185.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

Options

Unique While the options indicates that an unique entry shall be installed.

Shared While the options indicates that a shared entry shall be installed.

Replace While the options indicates that the entry handler routine shall be replaced.

Entry

Obj While the entry parameter references an object of type rtems_interrupt_entry.

Null While the entry parameter is equal to NULL.

Routine

Valid While the handler routine of the object referenced by the entry parameter is valid.

Null While the handler routine of the object referenced by the entry parameter is equal to NULL.

Init

Yes While the service is initialized.

No While the service is not initialized.

ISR

Yes While rtems_interrupt_entry_install is called from within interrupt context.

No While rtems_interrupt_entry_install is not called from within interrupt context.

CanEnable

Yes While the interrupt vector associated with the vector parameter can be enabled.

Maybe While the interrupt vector associated with the vector parameter may be enabled.

No While the interrupt vector associated with the vector parameter cannot be enabled.

Installed

None While the no entry is installed at the interrupt vector specified by the vector parameter.

Unique While a unique entry is installed at the interrupt vector specified by the vector parameter.

Other While at least one non-unique entry is installed at the interrupt vector specified by the vector parameter, while all entries installed at the interrupt vector specified by the vector parameter have a handler routine which is not equal to the handler routine of the object referenced by the entry parameter, while all entries installed at the interrupt vector specified by the vector parameter have a handler argument which is not equal to the handler argument of the object referenced by the entry parameter.

EqRoutine While at least one non-unique entry is installed at the interrupt vector specified by the vector parameter, while at least one entry installed at the interrupt vector specified by the vector parameter has a handler routine which is equal to the handler routine of the object referenced by the entry parameter, while all entries installed at the interrupt vector specified

by the vector parameter have a handler argument which is not equal to the handler argument of the object referenced by the entry parameter.

EqArg While at least one non-unique entry is installed at the interrupt vector specified by the vector parameter, while all entries installed at the interrupt vector specified by the vector parameter have a handler routine which is not equal to the handler routine of the object referenced by the entry parameter, while at least one entry installed at the interrupt vector specified by the vector parameter has a handler argument which is equal to the handler argument of the object referenced by the entry parameter.

Match While at least one non-unique entry with a handler routine which is equal to the handler routine of the object referenced by the entry parameter and with a handler argument which is equal to the handler argument of the object referenced by the entry parameter is installed at the interrupt vector specified by the vector parameter.

5.1.185.2 post-conditions

Status

Ok The return status of rtems_interrupt_entry_install shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_entry_install shall be RTEMS_INVALID_ADDRESS.

IncStat The return status of rtems_interrupt_entry_install shall be RTEMS_INCORRECT_STATE.

InvId The return status of rtems_interrupt_entry_install shall be RTEMS_INVALID_ID.

InvNum The return status of rtems_interrupt_entry_install shall be RTEMS_INVALID_NUMBER.

CalledFromISR The return status of rtems_interrupt_entry_install shall be RTEMS_CALLED_FROM_ISR.

InUse The return status of rtems_interrupt_entry_install shall be RTEMS_RESOURCE_IN_USE.

TooMany The return status of rtems_interrupt_entry_install shall be RTEMS_TOO_MANY.

Enable

Nop The enabled status of the interrupt vector specified by vector shall not be modified by the rtems_interrupt_entry_install call.

Yes The interrupt vector specified by vector shall be enabled.

Maybe The interrupt vector specified by vector may be enabled.

No The interrupt vector specified by vector shall not be enabled.

Installed

No The entry referenced by entry shall not be installed at the interrupt vector specified by vector.

Last The entry referenced by entry shall be installed as the last entry at the interrupt vector specified by vector.

5.1.185.3 transition-map

Status = *TooMany*, Enable = *Nop*, Installed = *No*

- Vector = *Valid*, Options = *Shared*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = *Match*

Status = *Ok*, Enable = *Yes*, Installed = *Last*

- Vector = *Valid*, Options = { *Unique*, *Shared* }, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = *Yes*, Installed = *None*
- Vector = *Valid*, Options = *Shared*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = *Yes*, Installed = { *Other*, *EqRoutine*, *EqArg* }

Status = *Ok*, Enable = *Maybe*, Installed = *Last*

- Vector = *Valid*, Options = { *Unique*, *Shared* }, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = *Maybe*, Installed = *None*
- Vector = *Valid*, Options = *Shared*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = *Maybe*, Installed = { *Other*, *EqRoutine*, *EqArg* }

Status = *Ok*, Enable = *No*, Installed = *Last*

- Vector = *Valid*, Options = { *Unique*, *Shared* }, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = *No*, Installed = *None*
- Vector = *Valid*, Options = *Shared*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = *No*, Installed = { *Other*, *EqRoutine*, *EqArg* }

Status = *InUse*, Enable = *Nop*, Installed = *No*

- Vector = *Valid*, Options = *Unique*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = { *Unique*, *Other*, *EqRoutine*, *EqArg*, *Match* }
- Vector = *Valid*, Options = *Shared*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = *Unique*

Status = *InvNum*, Enable = *Nop*, Installed = *No*

- Vector = *Valid*, Options = *Replace*, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *No*, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = { *None*, *Unique*, *Other*, *EqRoutine*, *EqArg*, *Match* }

Status = *CalledFromISR*, Enable = *Nop*, Installed = *No*

- Vector = *Valid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = *Yes*, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = { *None*, *Unique*, *Other*, *EqRoutine*, *EqArg*, *Match* }

Status = *InvAddr*, Enable = *Nop*, Installed = *No*

- Vector = *Valid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Obj*, Routine = *Null*, Init = *Yes*, ISR = { *Yes*, *No* }, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = { *None*, *Unique*, *Other*, *EqRoutine*, *EqArg*, *Match* }

Status = *InvId*, Enable = NA, Installed = NA

- Vector = *Invalid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Obj*, Routine = *Valid*, Init = *Yes*, ISR = { *Yes*, *No* }, CanEnable = NA, Installed = NA

Status = *IncStat*, Enable = *Nop*, Installed = *No*

- Vector = *Valid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Obj*, Routine = { *Valid*, *Null* }, Init = *No*, ISR = { *Yes*, *No* }, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = { *None*, *Unique*, *Other*, *EqRoutine*, *EqArg*, *Match* }

Status = *IncStat*, Enable = NA, Installed = NA

- Vector = *Invalid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Obj*, Routine = { *Valid*, *Null* }, Init = *No*, ISR = { *Yes*, *No* }, CanEnable = NA, Installed = NA

Status = *InvAddr*, Enable = *Nop*, Installed = NA

- Vector = *Valid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Null*, Routine = NA, Init = { *Yes*, *No* }, ISR = { *Yes*, *No* }, CanEnable = { *Yes*, *Maybe*, *No* }, Installed = { *None*, *Unique*, *Other*, *EqRoutine*, *EqArg*, *Match* }

Status = *InvAddr*, Enable = NA, Installed = NA

- Vector = *Invalid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Obj*, Routine = *Null*, Init = *Yes*, ISR = { *Yes*, *No* }, CanEnable = NA, Installed = NA
- Vector = *Invalid*, Options = { *Unique*, *Shared*, *Replace* }, Entry = *Null*, Routine = NA, Init = { *Yes*, *No* }, ISR = { *Yes*, *No* }, CanEnable = NA, Installed = NA

5.1.186 spec:/rtems/intr/req/entry-remove

spec:/rtems/intr/req/entry-remove

When the rtems_interrupt_entry_remove() (see: spec:/rtems/intr/if/entry-remove) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_entry_remove

5.1.186.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

Entry

Obj While the entry parameter references an object of type rtems_interrupt_entry.

Null While the entry parameter is equal to NULL.

Routine

Valid While the handler routine of the entry referenced by the options parameter is valid.

Null While the handler routine of the entry referenced by the options parameter is equal to NULL.

EntryObj

Installed While the entry referenced by the options parameter is installed at the interrupt vector specified by the vector parameter.

Match While the entry referenced by the options parameter not installed at the interrupt vector specified by the vector parameter, while the entry referenced by the options has a handler routine and argument pair which is equal to the handler routine and argument pair of an entry installed at the interrupt vector specified by the vector parameter.

NoMatch While the entry referenced by the options parameter not installed at the interrupt vector specified by the vector parameter, while the entry referenced by the options has a handler routine and argument pair which is not equal to the handler routine and argument pairs of all entries installed at the interrupt vector specified by the vector parameter.

Init

Yes While the service is initialized.

No While the service is not initialized.

ISR

Yes While rtems_interrupt_entry_remove is called from within interrupt context.

No While rtems_interrupt_entry_remove is not called from within interrupt context.

CanDisable

Yes While the interrupt vector associated with the vector parameter can be disabled.

Maybe While the interrupt vector associated with the vector parameter may be disabled.

No While the interrupt vector associated with the vector parameter cannot be disabled.

First

Yes While the entry referenced by the entry parameter is installed as the first entry at the interrupt vector specified by the vector parameter.

No While the entry referenced by the entry parameter is installed as not the first entry at the interrupt vector specified by the vector parameter.

Last

Yes While the entry referenced by the entry parameter is installed as the last entry at the interrupt vector specified by the vector parameter.

No While the entry referenced by the entry parameter is installed as not the last entry at the interrupt vector specified by the vector parameter.

5.1.186.2 post-conditions

Status

Ok The return status of rtems_interrupt_entry_remove shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_entry_remove shall be RTEMS_INVALID_ADDRESS.

IncStat The return status of rtems_interrupt_entry_remove shall be RTEMS_INCORRECT_STATE.

InvId The return status of rtems_interrupt_entry_remove shall be RTEMS_INVALID_ID.

CalledFromISR The return status of rtems_interrupt_entry_remove shall be RTEMS_CALLED_FROM_ISR.

Unsat The return status of rtems_interrupt_entry_remove shall be RTEMS_UNSATISFIED.

Disabled

Nop The enabled status of the interrupt vector specified by vector shall not be modified by the rtems_interrupt_entry_remove call.

Yes The interrupt vector specified by vector shall be disabled.

Maybe The interrupt vector specified by vector may be disabled.

No The interrupt vector specified by vector shall not be disabled.

Installed

No The entry referenced by entry shall not be installed at the interrupt vector specified by vector.

Nop The entries installed at the interrupt vector specified by vector shall not be modified by the rtems_interrupt_entry_remove call.

5.1.186.3 transition-map

Status = *Ok*, Disabled = *Yes*, Installed = *No*

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = *Installed*, Init = *Yes*, ISR = *No*, CanDisable = *Yes*, First = *Yes*, Last = *Yes*

Status = *Ok*, Disabled = *Maybe*, Installed = *No*

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = *Installed*, Init = *Yes*, ISR = *No*, CanDisable = *Maybe*, First = *Yes*, Last = *Yes*

Status = *Ok*, Disabled = *No*, Installed = *No*

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = *Installed*, Init = *Yes*, ISR = *No*, CanDisable = *No*, First = *Yes*, Last = *Yes*

Status = *Ok*, Disabled = *Nop*, Installed = *No*

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = *Installed*, Init = *Yes*, ISR = *No*, CanDisable = { *Yes*, *Maybe*, *No* }, First = *Yes*, Last = *No*
- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = *Installed*, Init = *Yes*, ISR = *No*, CanDisable = { *Yes*, *Maybe*, *No* }, First = *No*, Last = { *Yes*, *No* }

Status = *CalledFromISR*, Disabled = *Nop*, Installed = *Nop*

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = *Installed*, Init = *Yes*, ISR = *Yes*, CanDisable = { *Yes*, *Maybe*, *No* }, First = { *Yes*, *No* }, Last = { *Yes*, *No* }

Status = *CalledFromISR*, Disabled = *Nop*, Installed = NA

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = { *Match*, *NoMatch* }, Init = *Yes*, ISR = *Yes*, CanDisable = { *Yes*, *Maybe*, *No* }, First = NA, Last = NA

Status = *Unsat*, Disabled = *Nop*, Installed = NA

- Vector = *Valid*, Entry = *Obj*, Routine = *Valid*, EntryObj = { *Match*, *NoMatch* }, Init = *Yes*, ISR = *No*, CanDisable = { *Yes*, *Maybe*, *No* }, First = NA, Last = NA

Status = *InvId*, Disabled = NA, Installed = NA

- Vector = *Invalid*, Entry = *Obj*, Routine = *Valid*, EntryObj = NA, Init = *Yes*, ISR = { *Yes*, *No* }, CanDisable = NA, First = NA, Last = NA

Status = *IncStat*, Disabled = *Nop*, Installed = NA

- Vector = *Valid*, Entry = *Obj*, Routine = { *Valid*, *Null* }, EntryObj = NA, Init = *No*, ISR = { *Yes*, *No* }, CanDisable = { *Yes*, *Maybe*, *No* }, First = NA, Last = NA

Status = *IncStat*, Disabled = NA, Installed = NA

- Vector = *Invalid*, Entry = *Obj*, Routine = { *Valid*, *Null* }, EntryObj = NA, Init = *No*, ISR = { *Yes*, *No* }, CanDisable = NA, First = NA, Last = NA

Status = *InvAddr*, Disabled = *Nop*, Installed = NA

- Vector = *Valid*, Entry = *Obj*, Routine = *Null*, EntryObj = NA, Init = *Yes*, ISR = { *Yes*, *No* }, CanDisable = { *Yes*, *Maybe*, *No* }, First = NA, Last = NA
- Vector = *Valid*, Entry = *Null*, Routine = NA, EntryObj = NA, Init = { *Yes*, *No* }, ISR = { *Yes*, *No* }, CanDisable = { *Yes*, *Maybe*, *No* }, First = NA, Last = NA

Status = *InvAddr*, Disabled = NA, Installed = NA

- Vector = *Invalid*, Entry = *Obj*, Routine = *Null*, EntryObj = NA, Init = *Yes*, ISR = { *Yes*, *No* }, CanDisable = NA, First = NA, Last = NA
- Vector = *Invalid*, Entry = *Null*, Routine = NA, EntryObj = NA, Init = { *Yes*, *No* }, ISR = { *Yes*, *No* }, CanDisable = NA, First = NA, Last = NA

5.1.187 spec:/rtems/intr/req/get-affinity

spec:/rtems/intr/req/get-affinity

When the rtems_interrupt_get_affinity() (see: spec:/rtems/intr/if/get-affinity) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_get_affinity

5.1.187.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

CPUSetSize

Valid While the affinity_size parameter is an integral multiple of the size of long, while the affinity_size parameter specifies a processor set which is large enough to contain the processor affinity set of the interrupt specified by vector.

TooSmall While the affinity_size parameter is an integral multiple of the size of long, while the affinity_size parameter specifies a processor set which is not large enough to contain the processor affinity set of the interrupt specified by vector.

Askew While the affinity_size parameter is not an integral multiple of the size of long.

CPUSet

Valid While the affinity parameter references an object of type cpu_set_t.

Null While the affinity parameter is equal to NULL.

CanGetAffinity

Yes While getting the affinity for the interrupt vector specified by vector parameter is supported.

No While getting the affinity for the interrupt vector specified by vector parameter is not supported.

5.1.187.2 post-conditions

Status

Ok The return status of rtems_interrupt_get_affinity shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_get_affinity shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_interrupt_get_affinity shall be RTEMS_INVALID_ID.

InvSize The return status of rtems_interrupt_get_affinity shall be RTEMS_INVALID_SIZE.

Unsat The return status of rtems_interrupt_get_affinity shall be RTEMS_UNSATISFIED.

CPUSetObj

Set The value of the object referenced by the affinity parameter shall be set to the processor affinity set of the interrupt specified by the vector parameter at some point during the call after the return of the rtems_interrupt_get_affinity call.

Partial The value of the object referenced by the affinity parameter shall be set to the subset of the processor affinity set which fits into the object of the interrupt specified by the vector parameter at some point during the call after the return of the rtems_interrupt_get_affinity call.

Zero The value of the object referenced by the affinity parameter shall be set to zero.

Nop Objects referenced by the affinity parameter in past calls to rtems_interrupt_get_affinity shall not be accessed by the rtems_interrupt_get_affinity call.

5.1.187.3 transition-map

Status = *Ok*, CPUSetObj = *Set*

- Vector = *Valid*, CPUSetSize = *Valid*, CPUSet = *Valid*, CanGetAffinity = *Yes*

Status = *InvSize*, CPUSetObj = *Partial*

- Vector = *Valid*, CPUSetSize = *TooSmall*, CPUSet = *Valid*, CanGetAffinity = *Yes*

Status = *Unsat*, CPUSetObj = *Zero*

- Vector = *Valid*, CPUSetSize = { *Valid*, *TooSmall* }, CPUSet = *Valid*, CanGetAffinity = *No*

Status = *InvSize*, CPUSetObj = *Nop*

- Vector = *Valid*, CPUSetSize = *Askew*, CPUSet = *Valid*, CanGetAffinity = { *Yes*, *No* }

Status = *InvId*, CPUSetObj = *Zero*

- Vector = *Invalid*, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Valid*, CanGetAffinity = *NA*

Status = *InvAddr*, CPUSetObj = *Nop*

- Vector = *Valid*, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Null*, CanGetAffinity = { *Yes*, *No* }

- Vector = *Invalid*, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Null*, CanGetAffinity = *NA*

5.1.188 spec:/rtems/intr/req/get-attributes

spec:/rtems/intr/req/get-attributes

When the rtems_interrupt_get_attributes() (see: [spec:/rtems/intr/if/get-attributes](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_get_attributes

5.1.188.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

Attributes

Obj While the attributes parameter references an object of type rtems_interrupt_attributes.

Null While the attributes parameter is equal to NULL.

5.1.188.2 post-conditions

Status

Ok The return status of rtems_interrupt_get_attributes shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_get_attributes shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_interrupt_get_attributes shall be RTEMS_INVALID_ID.

Attributes

Nop Objects referenced by the attributes parameter in past calls to rtems_interrupt_get_attributes shall not be accessed by the rtems_interrupt_get_attributes call.

Zero The object referenced by the attributes parameter shall cleared to zero.

Set The members of the object referenced by the attributes parameter shall be set to the attributes of the interrupt vector specified by vector.

5.1.188.3 transition-map

Status = *Ok*, Attributes = *Set*

- Vector = *Valid*, Attributes = *Obj*

Status = *InvId*, Attributes = *Zero*

- Vector = *Invalid*, Attributes = *Obj*

Status = *InvAddr*, Attributes = *Nop*

- Vector = { *Valid*, *Invalid* }, Attributes = *Null*

5.1.189 spec:/rtems/intr/req/handler-iterate

spec:/rtems/intr/req/handler-iterate

When the rtems_interrupt_handler_iterate() (see: spec:/rtems/intr/if/handler-iterate) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_handler_iterate

5.1.189.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

Routine

Valid While the visitor routine specified by the routine parameter is valid.

Null While the visitor routine specified by the routine parameter is equal to NULL.

Init

Yes While the service is initialized.

No While the service is not initialized.

ISR

Yes While rtems_interrupt_handler_iterate is called from within interrupt context.

No While rtems_interrupt_handler_iterate is not called from within interrupt context.

5.1.189.2 post-conditions

Status

Ok The return status of rtems_interrupt_handler_iterate shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_handler_iterate shall be RTEMS_INVALID_ADDRESS.

IncStat The return status of rtems_interrupt_handler_iterate shall be RTEMS_INCORRECT_STATE.

InvId The return status of rtems_interrupt_handler_iterate shall be RTEMS_INVALID_ID.

CalledFromISR The return status of rtems_interrupt_handler_iterate shall be RTEMS_CALLED_FROM_ISR.

Visit

Yes For each interrupt entry installed at the interrupt vector specified by vector the visitor routine specified by routine shall be called with the argument specified by arg, the entry information, the entry install options, the entry handler routine, and the entry handler argument.

Nop The visitor routine specified by routine shall not be called.

5.1.189.3 transition-map

Status = *CalledFromISR*, Visit = *Nop*

- Vector = *Valid*, Routine = *Valid*, Init = *Yes*, ISR = *Yes*

Status = *Ok*, Visit = *Yes*

- Vector = *Valid*, Routine = *Valid*, Init = *Yes*, ISR = *No*

Status = *InvId*, Visit = *Nop*

- Vector = *Invalid*, Routine = *Valid*, Init = *Yes*, ISR = { *Yes*, *No* }

Status = *IncStat*, Visit = *Nop*

- Vector = { *Valid*, *Invalid* }, Routine = *Valid*, Init = *No*, ISR = { *Yes*, *No* }

Status = *InvAddr*, Visit = NA

- Vector = { *Valid*, *Invalid* }, Routine = *Null*, Init = *Yes*, ISR = { *Yes*, *No* }

Status = *IncStat*, Visit = NA

- Vector = { *Valid*, *Invalid* }, Routine = *Null*, Init = *No*, ISR = { *Yes*, *No* }

5.1.190 spec:/rtems/intr/req/is-pending

spec:/rtems/intr/req/is-pending

When the rtems_interrupt_is_pending() (see: spec:/rtems/intr/if/is-pending) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_is_pending

5.1.190.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

Pending

Obj While the pending parameter references an object of type bool.

Null While the pending parameter is equal to NULL.

IsPending

Yes While the interrupt associated with the interrupt vector specified by vector was pending for the processor executing the rtems_interrupt_is_pending call at some time point during the call.

No While the interrupt associated with the interrupt vector specified by vector was not pending for the processor executing the rtems_interrupt_is_pending call at some time point during the call.

5.1.190.2 post-conditions

Status

Ok The return status of rtems_interrupt_is_pending shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_is_pending shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_interrupt_is_pending shall be RTEMS_INVALID_ID.

IsPending

Nop Objects referenced by the pending parameter in past calls to rtems_interrupt_is_pending shall not be accessed by the rtems_interrupt_is_pending call.

Yes The value of the object referenced by the pending parameter shall be set to true.

No The value of the object referenced by the pending parameter shall be set to false.

5.1.190.3 transition-map

Status = **Ok**, IsPending = **Yes**

- Vector = **Valid**, Pending = **Obj**, IsPending = **Yes**

Status = **Ok**, IsPending = **No**

- Vector = **Valid**, Pending = **Obj**, IsPending = **No**

Status = **InvId**, IsPending = **Nop**

- Vector = **Invalid**, Pending = **Obj**, IsPending = **NA**

Status = **InvAddr**, IsPending = **Nop**

- Vector = **Valid**, Pending = **Null**, IsPending = { **Yes**, **No** }
- Vector = **Invalid**, Pending = **Null**, IsPending = **NA**

5.1.191 spec:/rtems/intr/req/local-disable-assign

spec:/rtems/intr/req/local-disable-assign

When the rtems_interrupt_local_disable directive is called, the status of the maskable interrupts before maskable interrupts are disabled by the directive shall be assigned to the variable specified by the _isr_cookie parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_local_disable

5.1.192 spec:/rtems/intr/req/local-enable

spec:/rtems/intr/req/local-enable

When the rtems_interrupt_local_enable directive is called, the status of maskable interrupts shall be restored according to the _isr_cookie parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_local_enable

5.1.193 spec:/rtems/intr/req/lock-acquire

spec:/rtems/intr/req/lock-acquire

When the directive is called, the lock contained in the object referenced by the _lock parameter shall be acquired.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_acquire

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_acquire_isr

5.1.194 spec:/rtems/intr/req/lock-declare-identifier

spec:/rtems/intr/req/lock-declare-identifier

When the RTEMS_INTERRUPT_LOCK_DECLARE is used, it shall expand to an object declaration with an identifier specified by the _designator parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_DECLARE

5.1.195 spec:/rtems/intr/req/lock-declare-specifier

spec:/rtems/intr/req/lock-declare-specifier

When the RTEMS_INTERRUPT_LOCK_DECLARE is used, it shall expand to an object declaration with a specifier specified by the _specifier parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_DECLARE

5.1.196 spec:/rtems/intr/req/lock-define-identifier

spec:/rtems/intr/req/lock-define-identifier

When the RTEMS_INTERRUPT_LOCK_DEFINE is used, it shall expand to an object definition with an identifier specified by the _designator parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_DEFINE

5.1.197 spec:/rtems/intr/req/lock-define-specifier

spec:/rtems/intr/req/lock-define-specifier

When the RTEMS_INTERRUPT_LOCK_DEFINE is used, it shall expand to an object definition with a specifier specified by the _specifier parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_DEFINE

5.1.198 spec:/rtems/intr/req/lock-destroy

spec:/rtems/intr/req/lock-destroy

When the rtems_interrupt_lock_destroy directive is called, the lock contained in the object referenced by the _lock parameter shall be destroyed.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_destroy

5.1.199 spec:/rtems/intr/req/lock-initialize

spec:/rtems/intr/req/lock-initialize

When the rtems_interrupt_lock_initialize directive is called, the lock contained in the object referenced by the _lock parameter shall be initialized so that it is available.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_initialize

5.1.200 spec:/rtems/intr/req/lock-initializer

spec:/rtems/intr/req/lock-initializer

When the RTEMS_INTERRUPT_LOCK_INITIALIZER is used to statically initialize an object of type rtems_interrupt_lock, the lock contained in the object shall be initialized so that it is available.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_INITIALIZER

5.1.201 spec:/rtems/intr/req/lock-member-identifier

spec:/rtems/intr/req/lock-member-identifier

When the RTEMS_INTERRUPT_LOCK_MEMBER is used, it shall expand to structure or union member declaration with an identifier specified by the _specifier parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_MEMBER

5.1.202 spec:/rtems/intr/req/lock-member-type

spec:/rtems/intr/req/lock-member-type

When the RTEMS_INTERRUPT_LOCK_MEMBER is used, it shall expand to structure or union member declaration with a type of rtems_interrupt_lock.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_MEMBER

5.1.203 spec:/rtems/intr/req/lock-reference-identifier

spec:/rtems/intr/req/lock-reference-identifier

When the RTEMS_INTERRUPT_LOCK_REFERENCE is used, it shall expand to an object reference definition with an identifier specified by the _designator parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_REFERENCE

5.1.204 spec:/rtems/intr/req/lock-reference-target

spec:/rtems/intr/req/lock-reference-target

When the RTEMS_INTERRUPT_LOCK_REFERENCE is used, it shall expand to an object reference definition which is initialized to the object reference specified by the _target parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_REFERENCE

5.1.205 spec:/rtems/intr/req/lock-reference-type

spec:/rtems/intr/req/lock-reference-type

When the RTEMS_INTERRUPT_LOCK_REFERENCE is used, it shall expand to an object reference definition with a type of a pointer to rtems_interrupt_lock.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - RTEMS_INTERRUPT_LOCK_REFERENCE

5.1.206 spec:/rtems/intr/req/lock-release

spec:/rtems/intr/req/lock-release

When the directive is called, the lock contained in the object referenced by the _lock parameter shall be released.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_release

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_release_isr

5.1.207 spec:/rtems/intr/req/lock-restore-interrupts

spec:/rtems/intr/req/lock-restore-interrupts

When the rtems_interrupt_lock_release directive is called, the status of maskable interrupts shall be restored according to the _lock_context parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_release

5.1.208 spec:/rtems/intr/req/lock-store-interrupt-status

spec:/rtems/intr/req/lock-store-interrupt-status

When the directive is called, the status of the maskable interrupts before maskable interrupts are disabled by the directive shall be stored to the lock context parameter.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_acquire

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_lock_interrupt_disable

5.1.209 spec:/rtems/intr/req/raise

spec:/rtems/intr/req/raise

When the rtems_interrupt_raise() (see: [spec:/rtems/intr/if/raise](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_raise

5.1.209.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

CanRaise

Yes While the interrupt vector associated with the vector parameter can be raised.

No While the interrupt vector associated with the vector parameter cannot be raised.

5.1.209.2 post-conditions

Status

Ok The return status of rtems_interrupt_raise shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_interrupt_raise shall be RTEMS_INVALID_ID.

Unsat The return status of rtems_interrupt_raise shall be RTEMS_UNSATISFIED.

Pending

Yes The interrupt associated with the interrupt vector specified by vector shall be made pending by the rtems_interrupt_raise call.

No The interrupt associated with the interrupt vector specified by vector shall not be made pending by the rtems_interrupt_raise call.

5.1.209.3 transition-map

Status = *Ok*, Pending = *Yes*

- Vector = *Valid*, CanRaise = *Yes*

Status = *Unsat*, Pending = *No*

- Vector = *Valid*, CanRaise = *No*

Status = *InvId*, Pending = NA

- Vector = *Invalid*, CanRaise = NA

5.1.210 spec:/rtems/intr/req/raise-on

spec:/rtems/intr/req/raise-on

When the rtems_interrupt_raise_on() (see: [spec:/rtems/intr/if/raise-on](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_raise_on

5.1.210.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

CPU

Online While the cpu_index parameter is associated with a configured processor, while the processor specified by cpu_index is not online.

NotOnline While the cpu_index parameter is associated with a configured processor, while the processor specified by cpu_index is not online.

NotConf While the cpu_index parameter is not associated with a configured processor.

CanRaiseOn

Yes While the interrupt vector associated with the vector parameter can be raised on a processor.

No While the interrupt vector associated with the vector parameter cannot be raised on a processor.

5.1.210.2 post-conditions

Status

- Ok** The return status of `rtems_interrupt_raise_on` shall be `RTEMS_SUCCESSFUL`.
- InvId** The return status of `rtems_interrupt_raise_on` shall be `RTEMS_INVALID_ID`.
- NotConf** The return status of `rtems_interrupt_raise_on` shall be `RTEMS_NOT_CONFIGURED`.
- IncStat** The return status of `rtems_interrupt_raise_on` shall be `RTEMS_INCORRECT_STATE`.
- Unsat** The return status of `rtems_interrupt_raise_on` shall be `RTEMS_UNSATISFIED`.

Pending

- Yes** The interrupt associated with the interrupt vector specified by vector shall be made pending by the `rtems_interrupt_raise_on` call.
- No** The interrupt associated with the interrupt vector specified by vector shall not be made pending by the `rtems_interrupt_raise_on` call.

5.1.210.3 skip-reasons

OnlyOneCPU Where the system is build with SMP support disabled, the system has exactly one processor and its processor is always online.

5.1.210.4 transition-map

Status = **Ok**, Pending = **Yes**

- Vector = **Valid**, CPU = **Online**, CanRaiseOn = **Yes**

Status = **Unsat**, Pending = **No**

- Vector = **Valid**, CPU = **Online**, CanRaiseOn = **No**

Status = **IncStat**, Pending = **No**

- Vector = **Valid**, CPU = **NotOnline**, CanRaiseOn = { **Yes**, **No** }

Status = **NotConf**, Pending = **No**

- Vector = **Valid**, CPU = **NotConf**, CanRaiseOn = { **Yes**, **No** }

Status = **InvId**, Pending = **NA**

- Vector = **Invalid**, CPU = { **Online**, **NotOnline**, **NotConf** }, CanRaiseOn = **NA**

5.1.211 spec:/rtems/intr/req/set-affinity

spec:/rtems/intr/req/set-affinity

When the rtems_interrupt_set_affinity() (see: [spec:/rtems/intr/if/set-affinity](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_set_affinity

5.1.211.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

CPUSetSize

Askew While the affinity_size parameter is not an integral multiple of the size of long.

Normal While the affinity_size parameter is an integral multiple of the size of long, while the affinity_size parameter is less than or equal to the maximum processor set size storable in the system.

Huge While the affinity_size parameter is an integral multiple of the size of long, while the affinity_size parameter is greater than the maximum processor set size storable in the system.

CPUSetOnline

Valid While the intersection of the processor set specified by the affinity_size and affinity parameters and the set of online processors is not empty, while the intersection of the processor set specified by the affinity_size and affinity parameters and the set of online processors is a processor affinity set supported by the interrupt vector.

Empty While the intersection of the processor set specified by the affinity_size and affinity parameters and the set of online processors is empty.

CPUSetHuge

NotZero While the processor set specified by the affinity_size and affinity parameters contains at least one processor which is not storable in a processor set supported by the system.

Zero While the processor set specified by the affinity_size and affinity parameters contains no processor which is not storable in a processor set supported by the system.

CPUSet

Valid While the affinity parameter references an object of type `cpu_set_t`.

Null While the affinity parameter is equal to `NULL`.

CanSetAffinity

Yes While setting the affinity for the interrupt vector specified by vector parameter is supported.

No While setting the affinity for the interrupt vector specified by vector parameter is not supported.

5.1.211.2 post-conditions

Status

Ok The return status of `rtems_interrupt_set_affinity` shall be `RTEMS_SUCCESSFUL`.

InvAddr The return status of `rtems_interrupt_set_affinity` shall be `RTEMS_INVALID_ADDRESS`.

InvId The return status of `rtems_interrupt_set_affinity` shall be `RTEMS_INVALID_ID`.

InvNum The return status of `rtems_interrupt_set_affinity` shall be `RTEMS_INVALID_NUMBER`.

Unsat The return status of `rtems_interrupt_set_affinity` shall be `RTEMS_UNSATISFIED`.

SetAffinity

Set The affinity set of the interrupt specified by vector shall be set to the processor set specified by `affinity_size` and `affinity` at some point during the `rtems_interrupt_set_affinity` call.

Nop The affinity set of the interrupt specified by vector shall not be modified by the `rtems_interrupt_set_affinity` call.

5.1.211.3 transition-map

Status = *Ok*, SetAffinity = *Set*

- Vector = *Valid*, CPUSetSize = *Normal*, CPUSetOnline = *Valid*, CPUSetHuge = NA, CPUSet = *Valid*, CanSetAffinity = *Yes*
- Vector = *Valid*, CPUSetSize = *Huge*, CPUSetOnline = *Valid*, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*, CanSetAffinity = *Yes*

Status = *Unsat*, SetAffinity = *Nop*

- Vector = *Valid*, CPUSetSize = *Normal*, CPUSetOnline = *Valid*, CPUSetHuge = NA, CPUSet = *Valid*, CanSetAffinity = *No*

- Vector = *Valid*, CPUSetSize = *Huge*, CPUSetOnline = *Valid*, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*, CanSetAffinity = *No*

Status = *InvNum*, SetAffinity = *Nop*

- Vector = *Valid*, CPUSetSize = *Askew*, CPUSetOnline = { *Valid*, *Empty* }, CPUSetHuge = NA, CPUSet = *Valid*, CanSetAffinity = { *Yes*, *No* }
- Vector = *Valid*, CPUSetSize = *Normal*, CPUSetOnline = *Empty*, CPUSetHuge = NA, CPUSet = *Valid*, CanSetAffinity = { *Yes*, *No* }
- Vector = *Valid*, CPUSetSize = *Huge*, CPUSetOnline = *Empty*, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*, CanSetAffinity = { *Yes*, *No* }

Status = *InvId*, SetAffinity = *Nop*

- Vector = *Invalid*, CPUSetSize = { *Askew*, *Normal* }, CPUSetOnline = NA, CPUSetHuge = NA, CPUSet = *Valid*, CanSetAffinity = NA
- Vector = *Invalid*, CPUSetSize = *Huge*, CPUSetOnline = NA, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*, CanSetAffinity = NA

Status = *InvAddr*, SetAffinity = *Nop*

- Vector = *Valid*, CPUSetSize = { *Askew*, *Normal*, *Huge* }, CPUSetOnline = NA, CPUSetHuge = NA, CPUSet = *Null*, CanSetAffinity = { *Yes*, *No* }
- Vector = *Invalid*, CPUSetSize = { *Askew*, *Normal*, *Huge* }, CPUSetOnline = NA, CPUSetHuge = NA, CPUSet = *Null*, CanSetAffinity = NA

5.1.212 spec:/rtems/intr/req/vector-disable

spec:/rtems/intr/req/vector-disable

When the rtems_interrupt_vector_disable() (see: [spec:/rtems/intr/if/vector-disable](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_vector_disable

5.1.212.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

.IsEnabled

Yes While the interrupt vector associated with the vector parameter is enabled.

No While the interrupt vector associated with the vector parameter is disabled.

CanDisable

Yes While the interrupt vector associated with the vector parameter can be disabled.

Maybe While the interrupt vector associated with the vector parameter may be disabled.

No While the interrupt vector associated with the vector parameter cannot be disabled.

5.1.212.2 post-conditions

Status

Ok The return status of rtems_interrupt_vector_disable shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_interrupt_vector_disable shall be RTEMS_INVALID_ID.

Unsat The return status of rtems_interrupt_vector_disable shall be RTEMS_UNSATISFIED.

.IsEnabled

Nop The enabled status of the interrupt vector specified by vector shall not be modified by the rtems_interrupt_vector_disable call.

No The interrupt vector specified by vector shall be disabled.

Maybe The interrupt vector specified by vector may be disabled.

5.1.212.3 transition-map

Status = **Ok**, IsEnabled = **No**

- Vector = **Valid**, IsEnabled = { **Yes**, **No** }, CanDisable = **Yes**

Status = **Ok**, IsEnabled = **Maybe**

- Vector = **Valid**, IsEnabled = { **Yes**, **No** }, CanDisable = **Maybe**

Status = **Unsat**, IsEnabled = **Nop**

- Vector = **Valid**, IsEnabled = { **Yes**, **No** }, CanDisable = **No**

Status = **InvId**, IsEnabled = **Nop**

- Vector = **Invalid**, IsEnabled = NA, CanDisable = NA

5.1.213 spec:/rtems/intr/req/vector-enable

spec:/rtems/intr/req/vector-enable

When the rtems_interrupt_vector_enable() (see: [spec:/rtems/intr/if/vector-enable](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_vector_enable

5.1.213.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

.IsEnabled

Yes While the interrupt vector associated with the vector parameter is enabled.

No While the interrupt vector associated with the vector parameter is disabled.

CanEnable

Yes While the interrupt vector associated with the vector parameter can be enabled.

Maybe While the interrupt vector associated with the vector parameter may be enabled.

No While the interrupt vector associated with the vector parameter cannot be enabled.

5.1.213.2 post-conditions

Status

Ok The return status of rtems_interrupt_vector_enable shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_interrupt_vector_enable shall be RTEMS_INVALID_ID.

Unsat The return status of rtems_interrupt_vector_enable shall be RTEMS_UNSATISFIED.

.IsEnabled

Nop The enabled status of the interrupt vector specified by vector shall not be modified by the rtems_interrupt_vector_enable call.

Yes The interrupt vector specified by vector shall be enabled.

Maybe The interrupt vector specified by vector may be enabled.

5.1.213.3 transition-map

Status = *Ok*, IsEnabled = *Yes*

- Vector = *Valid*, IsEnabled = { *Yes*, *No* }, CanEnable = *Yes*

Status = *Ok*, IsEnabled = *Maybe*

- Vector = *Valid*, IsEnabled = { *Yes*, *No* }, CanEnable = *Maybe*

Status = *Unsat*, IsEnabled = *Nop*

- Vector = *Valid*, IsEnabled = { *Yes*, *No* }, CanEnable = *No*

Status = *InvId*, IsEnabled = NA

- Vector = *Invalid*, IsEnabled = NA, CanEnable = NA

5.1.214 spec:/rtems/intr/req/vector-is-enabled

spec:/rtems/intr/req/vector-is-enabled

When the rtems_interrupt_vector_is_enabled() (see: spec:/rtems/intr/if/vector-is-enabled) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicIntr - rtems_interrupt_vector_is_enabled

5.1.214.1 pre-conditions

Vector

Valid While the vector parameter is associated with an interrupt vector.

Invalid While the vector parameter is not associated with an interrupt vector.

Enabled

Obj While the enabled parameter references an object of type bool.

Null While the enabled parameter is equal to NULL.

.IsEnabled

Yes While the interrupt associated with the interrupt vector specified by vector was enabled for the processor executing the rtems_interrupt_vector_is_enabled call at some time point during the call.

No While the interrupt associated with the interrupt vector specified by vector was disabled for the processor executing the rtems_interrupt_vector_is_enabled call at some time point during the call.

5.1.214.2 post-conditions

Status

Ok The return status of rtems_interrupt_vector_is_enabled shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_interrupt_vector_is_enabled shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_interrupt_vector_is_enabled shall be RTEMS_INVALID_ID.

.IsEnabled

Nop Objects referenced by the enabled parameter in past calls to rtems_interrupt_vector_is_enabled shall not be accessed by the rtems_interrupt_vector_is_enabled call.

Yes The value of the object referenced by the enabled parameter shall be set to true.

No The value of the object referenced by the enabled parameter shall be set to false.

5.1.214.3 transition-map

Status = *Ok*, IsEnabled = *Yes*

- Vector = *Valid*, Enabled = *Obj*, IsEnabled = *Yes*

Status = *Ok*, IsEnabled = *No*

- Vector = *Valid*, Enabled = *Obj*, IsEnabled = *No*

Status = *InvId*, IsEnabled = *Nop*

- Vector = *Invalid*, Enabled = *Obj*, IsEnabled = NA

Status = *InvAddr*, IsEnabled = *Nop*

- Vector = *Valid*, Enabled = *Null*, IsEnabled = { *Yes*, *No* }

- Vector = *Invalid*, Enabled = *Null*, IsEnabled = NA

5.1.215 spec:/rtems/io/req/bsp-output-char

spec:/rtems/io/req/bsp-output-char

When the function referenced by BSP_output_char is called, it shall output the character specified by the parameter to the kernel character output device.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIKernelCharIO - BSP_output_char

5.1.216 spec:/rtems/io/req/bsp-poll-char-dequeue

spec:/rtems/io/req/bsp-poll-char-dequeue

While the BSP_poll_char pointer references a function, while the kernel character input device used by the function has a character enqueued, the function shall dequeue the least recently enqueued character and return it as an unsigned character value.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIKernelCharIO - BSP_poll_char

5.1.217 spec:/rtems/io/req/bsp-poll-char-empty

spec:/rtems/io/req/bsp-poll-char-empty

While the BSP_poll_char pointer references a function, while the kernel character input device used by the function has no character enqueued, the function shall return minus one.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIKernelCharIO - BSP_poll_char

5.1.218 spec:/rtems/io/req/getchark

spec:/rtems/io/req/getchark

When the getchark() (see: [spec:/rtems/io/if/getchark](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPICKernelCharIO - getchark

5.1.218.1 pre-conditions

PollChar

Valid While BSP_poll_char references a function.

Null While BSP_poll_char is equal to NULL.

5.1.218.2 post-conditions

Result

PollChar The return value of getchark shall be the return value of the function referenced by BSP_poll_char.

MinusOne The return value of getchark shall be minus one.

Calls

Once The function referenced by BSP_poll_char shall be called exactly once to get the return value for getchark.

5.1.218.3 transition-map

Result = *PollChar*, Calls = *Once*

- PollChar = *Valid*

Result = *MinusOne*, Calls = NA

- PollChar = *Null*

5.1.219 spec:/rtems/io/req/put-char

spec:/rtems/io/req/put-char

When the rtems_put_char() (see: spec:/rtems/io/if/put-char) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPICKernelCharIO - rtems_put_char

5.1.219.1 pre-conditions

Char

Nl While the c parameter is equal to NL.

Other While the c parameter is not equal to NL.

5.1.219.2 post-conditions

Output

CrNl The function referenced by BSP_output_char shall be called with a CR character followed by a call with a NL character.

Other The function referenced by BSP_output_char shall be called with the character specified by c.

5.1.219.3 transition-map

Output = *CrNl*

- Char = *Nl*

Output = *Other*

- Char = *Other*

5.1.220 spec:/rtems/io/req/putc

spec:/rtems/io/req/putc

When the rtems_putc() (see: spec:/rtems/io/if/putc) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIKernelCharIO - rtems_putc

5.1.220.1 pre-conditions

Char

Nl While the c parameter is equal to NL.

Other While the c parameter is not equal to NL.

5.1.220.2 post-conditions

Output

CrNl The function referenced by BSP_output_char shall be called with a CR character followed by a call with a NL character.

Other The function referenced by BSP_output_char shall be called with the character specified by c.

5.1.220.3 transition-map

Output = *CrNl*

- Char = *Nl*

Output = *Other*

- Char = *Other*

5.1.221 spec:/rtems/malloc/req/calloc

spec:/rtems/malloc/req/calloc

When the rtems_calloc() (see: [spec:/rtems/malloc/if/calloc](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: malloc_8h - rtems_calloc

5.1.221.1 pre-conditions

ElementCount

Huge While the nelem parameter is not equal to zero, while the nelem parameter is too large to allocate a memory area with the specified size.

Zero While the nelem parameter is equal to zero.

Valid While the nelem parameter is not equal to zero, while the nelem parameter is small enough to allocate a memory area with the specified size.

ElementSize

Huge While the elsize parameter is not equal to zero, while the elsize parameter is too large to allocate a memory area with the specified size.

Zero While the elsize parameter is equal to zero.

Valid While the elsize parameter is not equal to zero, while the elsize parameter is small enough to allocate a memory area with the specified size.

5.1.221.2 post-conditions

Status

Null The return value of rtems_calloc() shall be equal to NULL.

AreaBegin The return value of rtems_calloc() shall be equal to the begin address of the allocated memory area.

Alignment

Valid The begin address of the allocated memory area shall be an integral multiple of the heap alignment of the target architecture.

Size

Valid The size of the allocated memory area shall greater than or equal to the product of the nelem and elsize parameters.

Content

Zero The content of the allocated memory area shall be cleared to zero.

5.1.221.3 transition-map

Status = *AreaBegin*, Alignment = *Valid*, Size = *Valid*, Content = *Zero*

- ElementCount = *Valid*, ElementSize = *Valid*

Status = *Null*, Alignment = *Valid*, Size = NA, Content = NA

- ElementCount = { *Huge*, *Zero* }, ElementSize = { *Huge*, *Zero*, *Valid* }
- ElementCount = *Valid*, ElementSize = { *Huge*, *Zero* }

5.1.222 spec:/rtems/malloc/req/malloc

spec:/rtems/malloc/req/malloc

When the rtems_malloc() (see: spec:/rtems/malloc/if/malloc) directive is called.

rationale: N/A

functional-type: action

Traced design component: alloconly_8c - rtems_malloc

5.1.222.1 pre-conditions

Size

Huge While the size parameter is not equal to zero, while the size parameter is too large to allocate a memory area with the specified size.

Zero While the size parameter is equal to zero.

Valid While the size parameter is not equal to zero, while the size parameter is small enough to allocate a memory area with the specified size.

5.1.222.2 post-conditions

Status

Null The return value of rtems_malloc() shall be equal to NULL.

AreaBegin The return value of rtems_malloc() shall be equal to the begin address of the allocated memory area.

Alignment

Valid The begin address of the allocated memory area shall be an integral multiple of the heap alignment of the target architecture.

Size

Valid The size of the allocated memory area shall greater than or equal to the `size` parameter.

5.1.222.3 transition-map

Status = *AreaBegin*, Alignment = *Valid*, Size = *Valid*

- Size = *Valid*

Status = *Null*, Alignment = *Valid*, Size = NA

- Size = { *Huge*, *Zero* }

5.1.223 spec:/rtems/message/req/broadcast

spec:/rtems/message/req/broadcast

When the rtems_message_queue_broadcast() (see: spec:/rtems/message/if/broadcast) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_broadcast

5.1.223.1 pre-conditions

SendBuffer

Valid While the buffer parameter references a memory area where the message to be sent is stored.

Null While the buffer parameter is NULL.

Count

Valid While the count parameter references an uint32_t object.

Null While the count parameter is NULL.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

MsgSize

Zero While the size parameter is 0.

SomeSize While the size parameter has a value between 0 and the maximum message size.

MaxSize While the size parameter has a value of the maximum message size.

TooLarge While the size parameter has a value greater than the maximum message size.

MsgQueue

Empty While there is no message in the message queue.

Several While there are messages in the message queue.

Receivers

Waiting While one or more receivers are waiting to receive a message.

None While no receiver is waiting to receive a message.

Storage

Nop While the memory area to which a pointer is provided as member storage_area of type rtems_message_queue_config when the message queue is constructed by rtems_message_queue_construct is altered only by the RTEMS operating system.

5.1.223.2 post-conditions

Status

Ok The return status of rtems_message_queue_broadcast shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_message_queue_broadcast shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_message_queue_broadcast shall be RTEMS_INVALID_ADDRESS.

InvSize The return status of rtems_message_queue_broadcast shall be RTEMS_INVALID_SIZE.

Count

Zero The value of the object referenced by the count parameter shall be set to 0 after the return of the rtems_message_queue_broadcast call.

Set The value of the object referenced by the count parameter shall be set to the number of tasks unblocked (see unblock) by the call to directive rtems_message_queue_broadcast after the return of the rtems_message_queue_broadcast call.

Nop The value of the object referenced by the count parameter in past call to rtems_message_queue_broadcast shall not be accessed by the rtems_message_queue_broadcast call (see also Nop).

MsgQueue

Nop Objects referenced by the id parameter in the past call to rtems_message_queue_broadcast shall not be accessed by that call (see also Nop).

Receivers

Unblocked The call to the rtems_message_queue_broadcast directive shall unblock all receivers waiting for a message at the message queue.

Note: Currently, rtems_message_queue_broadcast unblocks receivers in a none-atomic way. Meaning, it will not only unblock those receivers it finds waiting at the queue when rtems_message_queue_broadcast is invoked but also any new receivers which start waiting for messages after rtems_message_queue_broadcast is invoked and before it returns. This may lead to infinite unblocking loops.

Nop The receivers waiting for a message at the message queue shall not be affected by the call to the rtems_message_queue_broadcast directive.

RecSize

Message The values of the objects referenced by the size parameter in all calls to rtems_message_queue_receive which are unblocked (see unblock) by the rtems_message_queue_broadcast call shall be set to the same value as provided by parameter size of the rtems_message_queue_broadcast call after the return of the rtems_message_queue_broadcast call.

Nop Objects referenced by the size parameter in past calls to rtems_message_queue_receive shall not be accessed by the rtems_message_queue_broadcast call (see also Nop).

RecBuffer

Message Bytes 0 till size - 1 of the object referenced by the buffer parameter in all calls to rtems_message_queue_receive which are unblocked (see unblock) by the rtems_message_queue_broadcast call shall be set to the same values as bytes 0 till size - 1 of the object referenced by parameter buffer of the rtems_message_queue_broadcast call after the return of the rtems_message_queue_receive call.

Nop Objects referenced by the buffer parameter in past calls to rtems_message_queue_receive shall not be accessed by the rtems_message_queue_broadcast call (see also Nop).

5.1.223.3 skip-reasons

NoWait The message queue must be empty for an receiver to wait for a message.

5.1.223.4 transition-map

Status = *InvSize*, Count = *Nop*, MsgQueue = *Nop*, Receivers = *Nop*, RecSize = *Nop*, RecBuffer = *Nop*

- SendBuffer = *Valid*, Count = *Valid*, Id = *Valid*, MsgSize = *TooLarge*, MsgQueue = *Empty*, Receivers = *Waiting*, Storage = *Nop*

Status = *InvSize*, Count = *Nop*, MsgQueue = *Nop*, Receivers = NA, RecSize = NA, RecBuffer = NA

- SendBuffer = *Valid*, Count = *Valid*, Id = *Valid*, MsgSize = *TooLarge*, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Storage = *Nop*

Status = *Ok*, Count = *Set*, MsgQueue = *Nop*, Receivers = *Unblocked*, RecSize = *Message*, RecBuffer = *Message*

- SendBuffer = *Valid*, Count = *Valid*, Id = *Valid*, MsgSize = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = *Empty*, Receivers = *Waiting*, Storage = *Nop*

Status = *InvId*, Count = *Nop*, MsgQueue = *Nop*, Receivers = *Nop*, RecSize = *Nop*, RecBuffer = *Nop*

- SendBuffer = *Valid*, Count = *Valid*, Id = *Invalid*, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = *Empty*, Receivers = *Waiting*, Storage = *Nop*

Status = *Ok*, Count = *Zero*, MsgQueue = *Nop*, Receivers = NA, RecSize = NA, RecBuffer = NA

- SendBuffer = *Valid*, Count = *Valid*, Id = *Valid*, MsgSize = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Storage = *Nop*

Status = *InvId*, Count = *Nop*, MsgQueue = *Nop*, Receivers = NA, RecSize = NA, RecBuffer = NA

- SendBuffer = *Valid*, Count = *Valid*, Id = *Invalid*, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Storage = *Nop*

Status = *InvAddr*, Count = *Nop*, MsgQueue = *Nop*, Receivers = *Nop*, RecSize = *Nop*, RecBuffer = *Nop*

- SendBuffer = *Valid*, Count = *Null*, Id = { *Valid*, *Invalid* }, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = *Empty*, Receivers = *Waiting*, Storage = *Nop*

- SendBuffer = *Null*, Count = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = *Empty*, Receivers = *Waiting*, Storage = *Nop*

Status = *InvAddr*, Count = *Nop*, MsgQueue = *Nop*, Receivers = NA, RecSize = NA, RecBuffer = NA

- SendBuffer = *Valid*, Count = *Null*, Id = { *Valid*, *Invalid* }, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Storage = *Nop*

- SendBuffer = *Null*, Count = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Storage = *Nop*

NoWait

- SendBuffer = { *Valid*, *Null* }, Count = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, MsgSize = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = *Several*, Receivers = *Waiting*, Storage = *Nop*

5.1.224 spec:/rtems/message/req/construct

spec:/rtems/message/req/construct

When the rtems_message_queue_construct() (see: [spec:/rtems/message/if/construct](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_construct

5.1.224.1 pre-conditions

Config

Valid While the config parameter references an object of type rtems_message_queue_config.

Null While the config parameter is NULL.

Name

Valid While the name of the message queue configuration is valid.

Invalid While the name of the message queue configuration is invalid.

Id

Id While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

MaxPending

Valid While the maximum number of pending messages of the message queue configuration is valid.

Zero While the maximum number of pending messages of the message queue configuration is zero.

Big While the maximum number of pending messages of the message queue configuration is big enough so that a calculation to get the message buffer storage area size overflows.

MaxSize

Valid While the maximum message size of the message queue configuration is valid.

Zero While the maximum message size of the message queue configuration is zero.

Big While the maximum message size of the message queue configuration is big enough so that a calculation to get the message buffer storage area size overflows.

Free

Yes While the system has at least one inactive message queue object available.

No While the system has no inactive message queue object available.

Area

Valid While the message buffer storage area begin pointer of the message queue configuration is valid.

Null While the message buffer storage area begin pointer of the message queue configuration is NULL.

AreaSize

Valid While the message buffer storage area size of the message queue configuration is valid.

Invalid While the message buffer storage area size of the message queue configuration is invalid.

StorageFree

Null While the storage free member of the message queue configuration is equal to NULL.

Handler While the storage free member of the message queue configuration references a storage free handler.

5.1.224.2 post-conditions

Status

Ok The return status of rtems_message_queue_construct shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_message_queue_construct shall be RTEMS_INVALID_ADDRESS.

InvName The return status of rtems_message_queue_construct shall be RTEMS_INVALID_NAME.

InvNum The return status of rtems_message_queue_construct shall be RTEMS_INVALID_NUMBER.

InvSize The return status of rtems_message_queue_construct shall be RTEMS_INVALID_SIZE.

TooMany The return status of rtems_message_queue_construct shall be RTEMS_TOO_MANY.

Unsat The return status of rtems_message_queue_construct shall be RTEMS_UNSATISFIED.

Name

Valid The unique object name shall identify the message queue constructed by the rtems_message_queue_construct call.

Invalid The unique object name shall not identify a message queue.

IdObj

Set The value of the object referenced by the id parameter shall be set to the object identifier of the constructed message queue after the return of the rtems_message_queue_construct call.

Nop Objects referenced by the id parameter in past calls to rtems_message_queue_construct shall not be accessed by the rtems_message_queue_construct call.

StorageFree

Free The storage free handler of the message queue configuration specified by the config parameter shall be used to free the message queue storage area.

Nop No operation shall be performed to free the message queue storage area.

5.1.224.3 transition-map

Status = *Ok*, Name = *Valid*, IdObj = *Set*, StorageFree = *Nop*

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = *Valid*, MaxSize = *Valid*, Free = *Yes*, Area = *Valid*, AreaSize = *Valid*, StorageFree = *Null*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, StorageFree = *Free*

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = *Valid*, MaxSize = *Valid*, Free = *Yes*, Area = *Valid*, AreaSize = *Valid*, StorageFree = *Handler*

Status = *Unsat*, Name = *Invalid*, IdObj = *Nop*, StorageFree = NA

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = *Valid*, MaxSize = *Valid*, Free = *Yes*, Area = *Valid*, AreaSize = *Invalid*, StorageFree = { *Null*, *Handler* }
- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = *Valid*, MaxSize = *Valid*, Free = *Yes*, Area = *Null*, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

Status = *TooMany*, Name = *Invalid*, IdObj = *Nop*, StorageFree = NA

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = { *Valid*, *Big* }, MaxSize = { *Valid*, *Big* }, Free = *No*, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

Status = *InvSize*, Name = *Invalid*, IdObj = *Nop*, StorageFree = NA

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = { *Valid*, *Big* }, MaxSize = *Zero*, Free = { *Yes*, *No* }, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = { *Valid*, *Big* }, MaxSize = *Big*, Free = *Yes*, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

Status = *InvNum*, Name = *Invalid*, IdObj = *Nop*, StorageFree = NA

- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = *Zero*, MaxSize = { *Valid*, *Zero*, *Big* }, Free = { *Yes*, *No* }, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }
- Config = *Valid*, Name = *Valid*, Id = *Id*, MaxPending = *Big*, MaxSize = *Valid*, Free = *Yes*, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

Status = *InvName*, Name = *Invalid*, IdObj = *Nop*, StorageFree = NA

- Config = *Valid*, Name = *Invalid*, Id = { *Id*, *Null* }, MaxPending = { *Valid*, *Zero*, *Big* }, MaxSize = { *Valid*, *Zero*, *Big* }, Free = { *Yes*, *No* }, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

Status = *InvAddr*, Name = *Invalid*, IdObj = *Nop*, StorageFree = NA

- Config = *Valid*, Name = *Valid*, Id = *Null*, MaxPending = { *Valid*, *Zero*, *Big* }, MaxSize = { *Valid*, *Zero*, *Big* }, Free = { *Yes*, *No* }, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }
- Config = *Null*, Name = { *Valid*, *Invalid* }, Id = { *Id*, *Null* }, MaxPending = { *Valid*, *Zero*, *Big* }, MaxSize = { *Valid*, *Zero*, *Big* }, Free = { *Yes*, *No* }, Area = { *Valid*, *Null* }, AreaSize = { *Valid*, *Invalid* }, StorageFree = { *Null*, *Handler* }

5.1.225 spec:/rtems/message/req/delete

spec:/rtems/message/req/delete

When the rtems_message_queue_delete() (see: [spec:/rtems/message/if/delete](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_delete

5.1.225.1 pre-conditions

Id

NoObj While the id parameter is not associated with a message queue.

MsgQueue While the id parameter is associated with a message queue.

5.1.225.2 post-conditions

Status

Ok The return status of rtems_message_queue_delete shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_message_queue_delete shall be RTEMS_INVALID_ID.

Name

Valid The unique object name shall identify a message queue.

Invalid The unique object name shall not identify a message queue.

Flush

Yes Tasks waiting at the message queue shall be unblocked.

No Tasks waiting at the message queue shall remain blocked.

5.1.225.3 transition-map

Status = *InvId*, Name = *Valid*, Flush = *No*

- Id = *NoObj*

Status = *Ok*, Name = *Invalid*, Flush = *Yes*

- Id = *MsgQueue*

5.1.226 spec:/rtems/message/req/flush-pending

spec:/rtems/message/req/flush-pending

When the rtems_message_queue_flush() (see: spec:/rtems/message/if/flush) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_flush

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_get_number_pending

5.1.226.1 pre-conditions

Count

Valid While the count parameter references an uint32_t object.

Null While the count parameter is NULL.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

MsgQueue

Empty While there is no message in the message queue.

Several While there are messages in the message queue.

Receivers

Waiting While one or more receivers are waiting to receive a message.

None While no receiver is waiting to receive a message.

Directive

Flush While the directive rtems_message_queue_flush is called.

Pending While the directive rtems_message_queue_get_number_pending is called.

Storage

Nop While the memory area to which a pointer is provided as member storage_area of type rtems_message_queue_config when the message queue is constructed by rtems_message_queue_construct is altered only by the RTEMS operating system.

5.1.226.2 post-conditions

Status

Ok The return status of the called directive (rtems_message_queue_flush or rtems_message_queue_get_number_pending) shall be RTEMS_SUCCESSFUL.

InvId The return status of the called directive (rtems_message_queue_flush or rtems_message_queue_get_number_pending) shall be RTEMS_INVALID_ID.

InvAddr The return status of the called directive (rtems_message_queue_flush or rtems_message_queue_get_number_pending) shall be RTEMS_INVALID_ADDRESS.

Count

Zero The value of the object referenced by the count parameter shall be 0 after the return of the rtems_message_queue_flush or rtems_message_queue_get_number_pending call.

Set The rtems_message_queue_get_number_pending directive shall set the value of the object referenced by the count parameter to the number of messages present in the message queue at a point in time during the single execution of the rtems_message_queue_get_number_pending directive.

The rtems_message_queue_flush directive shall set the value of the object referenced by the count parameter to the number of messages it removed from the message queue during the single execution of the rtems_message_queue_flush directive.

Nop The value of the object referenced by the count parameter in past call to rtems_message_queue_flush or rtems_message_queue_get_number_pending shall not be accessed by the rtems_message_queue_flush or rtems_message_queue_get_number_pending call (see also Nop).

MsgQueue

Empty The message queue shall contain no messages after the last call to id.

Nop Objects referenced by the id parameter in the past call to rtems_message_queue_flush or rtems_message_queue_get_number_pending shall not be changed by that call (see also Nop).

Receivers

Nop The receivers waiting for a message at the message queue shall not be affected by the call to the rtems_message_queue_flush or rtems_message_queue_get_number_pending directive.

5.1.226.3 skip-reasons

NoWait The message queue must be empty for an receiver to wait for a message.

5.1.226.4 transition-map

Status = *Ok*, Count = *Zero*, MsgQueue = *Empty*, Receivers = *Nop*

- Count = *Valid*, Id = *Valid*, MsgQueue = *Empty*, Receivers = *Waiting*, Directive = *Flush*, Storage = *Nop*

Status = *Ok*, Count = *Zero*, MsgQueue = *Nop*, Receivers = *Nop*

- Count = *Valid*, Id = *Valid*, MsgQueue = *Empty*, Receivers = *Waiting*, Directive = *Pending*, Storage = *Nop*

Status = *Ok*, Count = *Zero*, MsgQueue = *Empty*, Receivers = NA

- Count = *Valid*, Id = *Valid*, MsgQueue = *Empty*, Receivers = *None*, Directive = *Flush*, Storage = *Nop*

Status = *Ok*, Count = *Zero*, MsgQueue = *Nop*, Receivers = NA

- Count = *Valid*, Id = *Valid*, MsgQueue = *Empty*, Receivers = *None*, Directive = *Pending*, Storage = *Nop*

Status = *Ok*, Count = *Set*, MsgQueue = *Empty*, Receivers = NA

- Count = *Valid*, Id = *Valid*, MsgQueue = *Several*, Receivers = *None*, Directive = *Flush*, Storage = *Nop*

Status = *Ok*, Count = *Set*, MsgQueue = *Nop*, Receivers = NA

- Count = *Valid*, Id = *Valid*, MsgQueue = *Several*, Receivers = *None*, Directive = *Pending*, Storage = *Nop*

Status = *InvId*, Count = *Nop*, MsgQueue = *Nop*, Receivers = *Nop*

- Count = *Valid*, Id = *Invalid*, MsgQueue = *Empty*, Receivers = *Waiting*, Directive = { *Flush*, *Pending* }, Storage = *Nop*

Status = *InvId*, Count = *Nop*, MsgQueue = *Nop*, Receivers = NA

- Count = *Valid*, Id = *Invalid*, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Directive = { *Flush*, *Pending* }, Storage = *Nop*

Status = *InvAddr*, Count = *Nop*, MsgQueue = *Nop*, Receivers = *Nop*

- Count = *Null*, Id = { *Valid*, *Invalid* }, MsgQueue = *Empty*, Receivers = *Waiting*, Directive = { *Flush*, *Pending* }, Storage = *Nop*

Status = *InvAddr*, Count = *Nop*, MsgQueue = *Nop*, Receivers = NA

- Count = *Null*, Id = { *Valid*, *Invalid* }, MsgQueue = { *Empty*, *Several* }, Receivers = *None*, Directive = { *Flush*, *Pending* }, Storage = *Nop*

NoWait

- Count = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, MsgQueue = *Several*, Receivers = *Waiting*, Directive = { *Flush*, *Pending* }, Storage = *Nop*

5.1.227 spec:/rtems/message/req/ident

spec:/rtems/message/req/ident

The rtems_message_queue_ident directive shall identify an Classic API message queue class object by its name as specified by /rtems/req/ident.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_ident

5.1.228 spec:/rtems/message/req/receive

spec:/rtems/message/req/receive

When the rtems_message_queue_receive() (see: spec:/rtems/message/if/receive) directive is called.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- spec:/score/tq/req/enqueue-fifo
- spec:/score/tq/req/enqueue-priority

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_receive

Traced design component: RTEMSAPIClassicTypes - RTEMS_NO_TIMEOUT

5.1.228.1 pre-conditions

Buffer

Valid While the buffer parameter references a memory area able to store a message up to the maximum size permitted in this message queue.

Null While the buffer parameter is NULL.

Size

Valid While the size parameter references an object of type size_t.

Null While the size parameter is NULL.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

DuringWait

Nop While no rtems_message_queue_delete directive is called successfully on the message queue during the time one or more tasks are waiting to receive messages.

Deleted While rtems_message_queue_delete is called successfully on the message queue while one or more tasks are waiting to receive messages.

TaskQueue

Fifo While the member attributes of type rtems_message_queue_config contains value RTEMS_FIFO when the message queue is constructed.

Note: RTEMS_GLOBAL is not part of the space profile because no remote nodes are supported.

Priority While the member attributes of type rtems_message_queue_config contains value RTEMS_PRIORITY when the message queue is constructed.

Note: RTEMS_GLOBAL is not part of the space profile because no remote nodes are supported.

Wait

No While the option_set parameter indicates the RTEMS_NO_WAIT option.

Timeout While the option_set parameter indicates the RTEMS_WAIT option, while the timeout parameter is not equal to RTEMS_NO_TIMEOUT.

Forever While the option_set parameter indicates the RTEMS_WAIT option, while the timeout parameter is equal to RTEMS_NO_TIMEOUT.

MsgQueue

Empty While there is no message in the message queue.

One While there is exactly one message in the message queue.

Several While there are more than one message in the message queue.

Storage

Nop While the memory area to which a pointer is provided as member storage_area of type rtems_message_queue_config when the message queue is constructed by rtems_message_queue_construct is altered only by the RTEMS operating system.

5.1.228.2 post-conditions

Status

Ok The return status of rtems_message_queue_receive shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_message_queue_receive shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_message_queue_receive shall be RTEMS_INVALID_ADDRESS.

Unsat The return status of rtems_message_queue_receive shall be RTEMS_UNSATISFIED.

Timeout The return status of rtems_message_queue_receive shall be RTEMS_TIMEOUT.

Deleted The return status of rtems_message_queue_receive shall be RTEMS_OBJECT_WAS_DELETED.

Delay

None The rtems_message_queue_receive call shall return immediately.

Ticks The rtems_message_queue_receive call shall return after the timeout period in ticks.

Forever The rtems_message_queue_receive call shall not return.

Size

First The value of the object referenced by the size parameter shall be set to the size of the first message (the same value as provided by parameter size of the rtems_message_queue_send or rtems_message_queue_urgent directive which added the message to the queue) after the return of the rtems_message_queue_receive call.

Nop Objects referenced by the size parameter in past calls to rtems_message_queue_receive shall not be accessed by the rtems_message_queue_receive call (see also Nop).

Msg

First The bytes 0 till size - 1 of the object referenced by the option_set parameter shall contain a copy of the content of the first message (all bytes unchanged and in the same order as provided by parameter buffer of the rtems_message_queue_send or rtems_message_queue_urgent directive which added the message to the queue) after the return of the rtems_message_queue_receive call.

Nop Objects referenced by the option_set parameter in past calls to rtems_message_queue_receive shall not be accessed by the rtems_message_queue_receive call (see also Nop).

MsgQueue

Empty The message queue shall be empty after the return of the rtems_message_queue_receive call.

OneLess The first message shall be removed from the message queue after the return of the rtems_message_queue_receive call.

Nop Objects referenced by the id parameter in past calls to rtems_message_queue_receive shall not be accessed by the rtems_message_queue_receive call (see also Nop).

Tasks

Fifo Where the thread queue uses the FIFO discipline, the calling thread shall be enqueued in FIFO order.

Priority Where the thread queue uses the priority discipline, the calling thread shall be enqueued in priority order.

5.1.228.3 skip-reasons

NoWait The case *Deleted* can only occur when at least one `rtems_message_queue_receive` call is waiting on an empty message queue.

5.1.228.4 transition-map

Status = *Timeout*, Delay = *Ticks*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = *Fifo*

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = *Fifo*, Wait = *Timeout*, MsgQueue = *Empty*, Storage = *Nop*

Status = NA, Delay = *Forever*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = *Fifo*

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = *Fifo*, Wait = *Forever*, MsgQueue = *Empty*, Storage = *Nop*

Status = *Timeout*, Delay = *Ticks*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = *Priority*

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = *Priority*, Wait = *Timeout*, MsgQueue = *Empty*, Storage = *Nop*

Status = NA, Delay = *Forever*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = *Priority*

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = *Priority*, Wait = *Forever*, MsgQueue = *Empty*, Storage = *Nop*

Status = *Unsat*, Delay = *None*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = NA

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = { *Fifo*, *Priority* }, Wait = *No*, MsgQueue = *Empty*, Storage = *Nop*

Status = *Deleted*, Delay = *None*, Size = *Nop*, Msg = *Nop*, MsgQueue = NA, Tasks = NA

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Deleted*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *Timeout*, *Forever* }, MsgQueue = *Empty*, Storage = *Nop*

Status = *Ok*, Delay = *None*, Size = *First*, Msg = *First*, MsgQueue = *Empty*, Tasks = NA

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = *One*, Storage = *Nop*

Status = *Ok*, Delay = *None*, Size = *First*, Msg = *First*, MsgQueue = *OneLess*, Tasks = NA

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Nop*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = *Several*, Storage = *Nop*

Status = *InvId*, Delay = *None*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = NA

- Buffer = *Valid*, Size = *Valid*, Id = *Invalid*, DuringWait = *Nop*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*

Status = *InvId*, Delay = *None*, Size = *Nop*, Msg = *Nop*, MsgQueue = NA, Tasks = NA

- Buffer = *Valid*, Size = *Valid*, Id = *Invalid*, DuringWait = *Deleted*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*

Status = *InvAddr*, Delay = *None*, Size = *Nop*, Msg = *Nop*, MsgQueue = *Nop*, Tasks = NA

- Buffer = *Valid*, Size = *Null*, Id = { *Valid*, *Invalid* }, DuringWait = *Nop*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*
- Buffer = *Null*, Size = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, DuringWait = *Nop*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*

Status = *InvAddr*, Delay = *None*, Size = *Nop*, Msg = *Nop*, MsgQueue = NA, Tasks = NA

- Buffer = *Valid*, Size = *Null*, Id = { *Valid*, *Invalid* }, DuringWait = *Deleted*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*
- Buffer = *Null*, Size = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, DuringWait = *Deleted*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *No*, *Timeout*, *Forever* }, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*

NoWait

- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Deleted*, TaskQueue = { *Fifo*, *Priority* }, Wait = *No*, MsgQueue = { *Empty*, *One*, *Several* }, Storage = *Nop*
- Buffer = *Valid*, Size = *Valid*, Id = *Valid*, DuringWait = *Deleted*, TaskQueue = { *Fifo*, *Priority* }, Wait = { *Timeout*, *Forever* }, MsgQueue = { *One*, *Several* }, Storage = *Nop*

5.1.229 spec:/rtems/message/req/urgent-send

spec:/rtems/message/req/urgent-send

When the rtems_message_queue_send() (see: spec:/rtems/message/if/send) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_send

Traced design component: RTEMSAPIClassicMessage - rtems_message_queue_urgent

5.1.229.1 pre-conditions

Buffer

Valid While the buffer parameter references a memory area where the message to be sent is stored.

Null While the buffer parameter is NULL.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Size

Zero While the size parameter is 0.

SomeSize While the size parameter has a value between 0 and the maximum message size.

MaxSize While the size parameter has a value of the maximum message size.

TooLarge While the size parameter has a value greater than the maximum message size.

MsgQueue

Empty While there is no message in the message queue.

One While there is exactly one message in the message queue.

Several While there are more than one and less than maximum pending messages in the message queue.

Full While there are maximum pending messages in the message queue.

Receiver

Waiting While a receiver is waiting to receive a message.

No While no receiver is waiting to receive a message.

Directive

Send While the directive rtems_message_queue_send is called.

Urgent While the directive rtems_message_queue_urgent is called.

Storage

Nop While the memory area to which a pointer is provided as member storage_area of type rtems_message_queue_config when the message queue is constructed by rtems_message_queue_construct is altered only by the RTEMS operating system.

5.1.229.2 post-conditions

Status

Ok The return status of the called directive (rtems_message_queue_send or rtems_message_queue_urgent) shall be RTEMS_SUCCESSFUL

InvId The return status of the called directive (rtems_message_queue_send or rtems_message_queue_urgent) shall be RTEMS_INVALID_ID.

InvAddr The return status of the called directive (rtems_message_queue_send or rtems_message_queue_urgent) shall be RTEMS_INVALID_ADDRESS.

InvSize The return status of the called directive (rtems_message_queue_send or rtems_message_queue_urgent) shall be RTEMS_INVALID_SIZE.

TooMany The return status of the called directive (rtems_message_queue_send or rtems_message_queue_urgent) shall be RTEMS_TOO_MANY.

MsgQueue

Empty The message queue shall be empty after the return of the rtems_message_queue_send or rtems_message_queue_urgent call.

One The message queue shall contain only the send message after the return of the rtems_message_queue_send or rtems_message_queue_urgent call.

Prepend The message queue shall contain the message send by the last call to rtems_message_queue_urgent as first message followed by all the messages which were in the message queue before that call (in the same order and each message with the same content and size).

Append The message queue shall contain the message send by the last call to rtems_message_queue_send as last message preceded by all the messages which were in the message queue before that call (in the same order and each message with the same content and size).

Nop Objects referenced by the id parameter in past call to rtems_message_queue_send or rtems_message_queue_urgent shall not be accessed by that call (see also Nop).

Receiver

GotMsg The receiver shall receive the message send by the last call to the rtems_message_queue_send or rtems_message_queue_urgent directive.

Waiting The receiver shall still wait to receive a message after the last call to the rtems_message_queue_send or rtems_message_queue_urgent directive.

5.1.229.3 skip-reasons

NoWait The message queue must be empty for an receiver to wait for a message.

5.1.229.4 transition-map

Status = *InvSize*, MsgQueue = *Nop*, Receiver = *Waiting*

- Buffer = *Valid*, Id = *Valid*, Size = *TooLarge*, MsgQueue = *Empty*, Receiver = *Waiting*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *Ok*, MsgQueue = *Empty*, Receiver = *GotMsg*

- Buffer = *Valid*, Id = *Valid*, Size = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = *Empty*, Receiver = *Waiting*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *Ok*, MsgQueue = *One*, Receiver = NA

RTEMS Qualification Software Requirement Specification [sparc/gr712rc/smp/4]

Release 3

ESA Contract No. 4000125572/18/NL/GLC/as

- Buffer = *Valid*, Id = *Valid*, Size = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = *Empty*, Receiver = *No*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *Ok*, MsgQueue = *Append*, Receiver = NA

- Buffer = *Valid*, Id = *Valid*, Size = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = { *One*, *Several* }, Receiver = *No*, Directive = *Send*, Storage = *Nop*

Status = *Ok*, MsgQueue = *Prepend*, Receiver = NA

- Buffer = *Valid*, Id = *Valid*, Size = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = { *One*, *Several* }, Receiver = *No*, Directive = *Urgent*, Storage = *Nop*

Status = *TooMany*, MsgQueue = *Nop*, Receiver = NA

- Buffer = *Valid*, Id = *Valid*, Size = { *Zero*, *SomeSize*, *MaxSize* }, MsgQueue = *Full*, Receiver = *No*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *InvSize*, MsgQueue = *Nop*, Receiver = NA

- Buffer = *Valid*, Id = *Valid*, Size = *TooLarge*, MsgQueue = { *Empty*, *One*, *Several*, *Full* }, Receiver = *No*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *InvId*, MsgQueue = *Nop*, Receiver = *Waiting*

- Buffer = *Valid*, Id = *Invalid*, Size = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = *Empty*, Receiver = *Waiting*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *InvAddr*, MsgQueue = *Nop*, Receiver = *Waiting*

- Buffer = *Null*, Id = { *Valid*, *Invalid* }, Size = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = *Empty*, Receiver = *Waiting*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *InvId*, MsgQueue = *Nop*, Receiver = NA

- Buffer = *Valid*, Id = *Invalid*, Size = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = { *Empty*, *One*, *Several*, *Full* }, Receiver = *No*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

Status = *InvAddr*, MsgQueue = *Nop*, Receiver = NA

- Buffer = *Null*, Id = { *Valid*, *Invalid* }, Size = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = { *Empty*, *One*, *Several*, *Full* }, Receiver = *No*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

NoWait

- Buffer = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Size = { *Zero*, *SomeSize*, *MaxSize*, *TooLarge* }, MsgQueue = { *One*, *Several*, *Full* }, Receiver = *Waiting*, Directive = { *Send*, *Urgent* }, Storage = *Nop*

5.1.230 spec:/rtems/mode/req/interrupt-level

spec:/rtems/mode/req/interrupt-level

When the /rtems/mode/if/interrupt-level macro is used, it shall expand to a constant expression which computes the bitwise and of the value specified by the _interrupt_level parameter and RTEMS_INTERRUPT_MASK.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicModes - RTEMS_INTERRUPT_LEVEL

5.1.231 spec:/rtems/object/req/build-name

spec:/rtems/object/req/build-name

When the rtems_build_name directive is called, the directive shall return the return value of a call to rtems_build_name.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicObject - rtems_build_name

5.1.232 spec:/rtems/object/req/build-name-macro

spec:/rtems/object/req/build-name-macro

The result of rtems_build_name shall be equal to (_c1 modulo 256) * 16777216 + (_c2 modulo 256) * 65536 + (_c3 modulo 256) * 256 + (_c4 modulo 256).

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicObject - rtems_build_name

5.1.233 spec:/rtems/object/req/get-local-node

spec:/rtems/object/req/get-local-node

The result of rtems_object_get_local_node shall be equal to the local node number.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicObject - rtems_object_get_local_node

5.1.234 spec:/rtems/part/req/alignment

spec:/rtems/part/req/alignment

The RTEMS_PARTITION_ALIGNMENT constant shall expand to a constant expression which evaluates to the value of CPU_SIZEOF_POINTER.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicPart - RTEMS_PARTITION_ALIGNMENT

5.1.235 spec:/rtems/part/req/buffers

spec:/rtems/part/req/buffers

The count of buffers available for use from a partition shall be exactly the buffer area length divided by the buffer size (integer division) specified at partition creation.

rationale: N/A

functional-type: function

5.1.236 spec:/rtems/part/req/create

spec:/rtems/part/req/create

When the rtems_partition_create() (see: [spec:/rtems/part/if/create](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicPart - rtems_partition_create

5.1.236.1 pre-conditions

Name

Valid While the name parameter is valid.

Invalid While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

Start

Valid While the starting_address parameter is valid.

Null While the starting_address parameter is NULL.

BadAlign While the starting_address parameter is misaligned.

Length

Valid While the length parameter is valid.

Zero While the length parameter is zero.

Invalid While the length parameter is less than the buffer size.

Size

Valid While the buffer_size parameter is valid.

Zero While the buffer_size parameter is zero.

Skew The buffer_size parameter shall not an integral multiple of the pointer size.

Small The buffer_size parameter shall greater than zero and an integral multiple of the pointer size and less than the size of two pointers.

Free

Yes While the system has at least one inactive partition object available.

No While the system has no inactive partition object available.

5.1.236.2 post-conditions

Status

- Ok** The return status of rtems_partition_create shall be RTEMS_SUCCESSFUL.
- InvAddr** The return status of rtems_partition_create shall be RTEMS_INVALID_ADDRESS.
- InvName** The return status of rtems_partition_create shall be RTEMS_INVALID_NAME.
- InvSize** The return status of rtems_partition_create shall be RTEMS_INVALID_SIZE.
- TooMany** The return status of rtems_partition_create shall be RTEMS_TOO_MANY.

Name

- Valid** The unique object name shall identify the partition created by the rtems_partition_create call.
- Invalid** The unique object name shall not identify a partition.

IdVar

- Set** The value of the object referenced by the id parameter shall be set to the object identifier of the created partition after the return of the rtems_partition_create call.
- Nop** Objects referenced by the id parameter in past calls to rtems_partition_create shall not be accessed by the rtems_partition_create call.

5.1.236.3 transition-map

Status = *Ok*, Name = *Valid*, IdVar = *Set*

- Name = *Valid*, Id = *Valid*, Start = *Valid*, Length = *Valid*, Size = *Valid*, Free = *Yes*

Status = *TooMany*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Start = *Valid*, Length = *Valid*, Size = *Valid*, Free = *No*

Status = *InvSize*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Start = { *Valid*, *BadAlign* }, Length = *Valid*, Size = { *Zero*, *Skew*, *Small* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Start = { *Valid*, *BadAlign* }, Length = { *Zero*, *Invalid* }, Size = { *Valid*, *Zero*, *Skew*, *Small* }, Free = { *Yes*, *No* }

Status = *InvAddr*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Start = *Null*, Length = { *Valid*, *Zero*, *Invalid* }, Size = { *Valid*, *Zero*, *Skew*, *Small* }, Free = { *Yes*, *No* }

- Name = *Valid*, Id = *Valid*, Start = *BadAlign*, Length = *Valid*, Size = *Valid*, Free = { Yes, No }
- Name = *Valid*, Id = *Null*, Start = { *Valid*, *Null*, *BadAlign* }, Length = { *Valid*, *Zero*, *Invalid* }, Size = { *Valid*, *Zero*, *Skew*, *Small* }, Free = { Yes, No }

Status = *InvName*, Name = *Invalid*, IdVar = *Nop*

- Name = *Invalid*, Id = { *Valid*, *Null* }, Start = { *Valid*, *Null*, *BadAlign* }, Length = { *Valid*, *Zero*, *Invalid* }, Size = { *Valid*, *Zero*, *Skew*, *Small* }, Free = { Yes, No }

5.1.237 spec:/rtems/part/req/delete

spec:/rtems/part/req/delete

When the rtems_partition_delete() (see: [spec:/rtems/part/if/delete](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicPart - rtems_partition_delete

5.1.237.1 pre-conditions

Id

NoObj While the id parameter is not associated with a partition.

Part While the id parameter is associated with a partition.

InUse

Yes While the partition has at least one buffer in use.

No While the partition does not have a buffer in use.

5.1.237.2 post-conditions

Status

Ok The status shall be RTEMS_SUCCESSFUL. The deleted partition object shall be inactive.

InvId The status shall be RTEMS_INVALID_ID.

InUse The status shall be RTEMS_RESOURCE_IN_USE.

5.1.237.3 transition-map

Status = *InUse*

- Id = *Part*, InUse = *Yes*

Status = *Ok*

- Id = *Part*, InUse = *No*

Status = *InvId*

- Id = *NoObj*, InUse = { *Yes*, *No* }

5.1.238 spec:/rtems/part/req/fifo

spec:/rtems/part/req/fifo

A partition shall maintain free buffers in FIFO order.

rationale: N/A

functional-type: function

5.1.239 spec:/rtems/part/req/get-buffer

spec:/rtems/part/req/get-buffer

When the rtems_partition_get_buffer() (see: spec:/rtems/part/if/get-buffer) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicPart - rtems_partition_get_buffer

5.1.239.1 pre-conditions

Id

NoObj While the id parameter is not associated with a partition.

Part While the id parameter is associated with a partition.

Buf

Valid While the buffer parameter references an object of type void *.

Null While the buffer parameter is NULL.

Avail

Yes While the partition has at least one free buffer available.

No While the partition does not have a buffer available.

5.1.239.2 post-conditions

Status

Ok The return status of rtems_partition_get_buffer shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_partition_get_buffer shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_partition_get_buffer shall be RTEMS_INVALID_ADDRESS.

Unsat The return status of rtems_partition_get_buffer shall be RTEMS_UNSATISFIED.

BufVar

Set The value of the object referenced by the starting_address parameter shall be set to the object identifier of the begin address of the returned buffer after the return of the rtems_partition_create call.

Nop Objects referenced by the buffer parameter in past calls to rtems_partition_get_buffer shall not be accessed by the rtems_partition_get_buffer call.

5.1.239.3 transition-map

Status = *Ok*, BufVar = *Set*

- Id = *Part*, Buf = *Valid*, Avail = *Yes*

Status = *Unsat*, BufVar = *Nop*

- Id = *Part*, Buf = *Valid*, Avail = *No*

Status = *InvId*, BufVar = *Nop*

- Id = *NoObj*, Buf = *Valid*, Avail = { *Yes*, *No* }

Status = *InvAddr*, BufVar = *Nop*

- Id = { *NoObj*, *Part* }, Buf = *Null*, Avail = { *Yes*, *No* }

5.1.240 spec:/rtems/part/req/ident

spec:/rtems/part/req/ident

The rtems_partition_ident directive shall identify an Classic API partition class object by its name as specified by /rtems/req/ident.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicPart - rtems_partition_ident

5.1.241 spec:/rtems/part/req/return-buffer

spec:/rtems/part/req/return-buffer

When the rtems_partition_return_buffer() (see: [spec:/rtems/part/if/return-buffer](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicPart - rtems_partition_return_buffer

5.1.241.1 pre-conditions

Id

NoObj While the id parameter is invalid.

Part While the id parameter is associated with a partition.

Buf

Valid While the buffer parameter references a buffer previously returned by rtems_partition_get_buffer.

BadAlign While the buffer parameter is an address inside the buffer area of the partition, while the address is not on a valid buffer boundary.

BelowArea While the buffer parameter is an address below the buffer area of the partition.

AboveArea While the buffer parameter is an address above the buffer area of the partition.

5.1.241.2 post-conditions

Status

Ok The return status of rtems_partition_return_buffer shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_partition_return_buffer shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_partition_return_buffer shall be RTEMS_INVALID_ADDRESS.

Buf

Free The buffer obtained from the partition shall be made available for re-use by the rtems_partition_return_buffer call.

InUse The buffer obtained from the partition shall be still in use after the rtems_partition_return_buffer call.

5.1.241.3 transition-map

Status = *Ok*, Buf = *Free*

- Id = *Part*, Buf = *Valid*

Status = *InvAddr*, Buf = *InUse*

- Id = *Part*, Buf = { *BadAlign*, *BelowArea*, *AboveArea* }

Status = *InvId*, Buf = *InUse*

- Id = *NoObj*, Buf = { *Valid*, *BadAlign*, *BelowArea*, *AboveArea* }

5.1.242 spec:/rtems/ratemon/req/cancel

spec:/rtems/ratemon/req/cancel

When the rtems_rate_monotonic_cancel() (see: spec:/rtems/ratemon/if/cancel) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicRatemon - rtems_rate_monotonic_cancel

5.1.242.1 pre-conditions

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Caller

OwnerTask While the task invoking rtems_rate_monotonic_cancel is the task which created the period - the owner task.

OtherTask While the task invoking rtems_rate_monotonic_cancel is not the owner task.

State

Inactive While the id parameter references an period object in inactive state.

Active While the id parameter references an period object in active state.

Expired While the id parameter references an period object in expired state.

Postponed

Zero While the period is not in expired state.

One While there is one postponed job.

Several While there are two or more postponed jobs.

5.1.242.2 post-conditions

Status

Ok The return status of rtems_rate_monotonic_cancel shall be RTEMS_SUCCESSFUL

InvId The return status of rtems_rate_monotonic_cancel shall be RTEMS_INVALID_ID.

NotOwn The return status of rtems_rate_monotonic_cancel shall be RTEMS_NOT_OWNER_OF_RESOURCE.

State

Inactive The state of the period shall be inactive after the return of the rtems_rate_monotonic_cancel call.

Nop Objects referenced by the id parameter in past calls to rtems_rate_monotonic_cancel shall not be accessed by the rtems_rate_monotonic_cancel call (see also Nop).

Postponed

Zero There shall be no postponed jobs after the return of the rtems_rate_monotonic_cancel call.

Nop Objects referenced by the id parameter in past calls to rtems_rate_monotonic_cancel shall not be accessed by the rtems_rate_monotonic_cancel call (see also Nop).

Scheduler

Called The last call of the rtems_rate_monotonic_cancel function shall execute the cancel_job scheduler operation of the home scheduler.

Nop The last call of the rtems_rate_monotonic_cancel function shall not execute any scheduler operation.

5.1.242.3 skip-reasons

NotInInactiveState postponed jobs do not exist in inactive state.

NeverInExpiredState There must be postponed jobs in expired state.

5.1.242.4 transition-map

Status = *Ok*, State = *Inactive*, Postponed = NA, Scheduler = *Called*

- Id = *Valid*, Caller = *OwnerTask*, State = *Inactive*, Postponed = NA

Status = *Ok*, State = *Inactive*, Postponed = *Zero*, Scheduler = *Called*

- Id = *Valid*, Caller = *OwnerTask*, State = *Active*, Postponed = { *Zero*, *One*, *Several* }
- Id = *Valid*, Caller = *OwnerTask*, State = *Expired*, Postponed = { *One*, *Several* }

Status = *NotOwn*, State = *Nop*, Postponed = *Nop*, Scheduler = *Nop*

- Id = *Valid*, Caller = *OtherTask*, State = *Inactive*, Postponed = NA
- Id = *Valid*, Caller = *OtherTask*, State = *Active*, Postponed = { *Zero*, *One*, *Several* }
- Id = *Valid*, Caller = *OtherTask*, State = *Expired*, Postponed = { *One*, *Several* }

Status = *InvId*, State = *Nop*, Postponed = *Nop*, Scheduler = *Nop*

- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, State = *Inactive*, Postponed = NA
- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, State = *Active*, Postponed = { *Zero*, *One*, *Several* }
- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, State = *Expired*, Postponed = { *One*, *Several* }

NotInInactiveState

- Id = { *Valid*, *Invalid* }, Caller = { *OwnerTask*, *OtherTask* }, State = *Inactive*, Postponed = { *One*, *Several* }

NeverInExpiredState

- Id = { *Valid*, *Invalid* }, Caller = { *OwnerTask*, *OtherTask* }, State = *Expired*, Postponed = *Zero*

5.1.243 spec:/rtems/ratemon/req/create

spec:/rtems/ratemon/req/create

When the rtems_rate_monotonic_create() (see: [spec:/rtems/ratemon/if/create](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicRatemon - rtems_rate_monotonic_create

5.1.243.1 pre-conditions

Name

Valid While the name parameter is valid.

Invalid While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

Free

Yes While the system has at least one inactive period object available.

No While the system has no inactive period object available.

5.1.243.2 post-conditions

Status

Ok The return status of rtems_rate_monotonic_create shall be RTEMS_SUCCESSFUL.

InvName The return status of rtems_rate_monotonic_create shall be RTEMS_INVALID_NAME.

InvAddr The return status of rtems_rate_monotonic_create shall be RTEMS_INVALID_ADDRESS.

TooMany The return status of rtems_rate_monotonic_create shall be RTEMS_TOO_MANY.

Name

Valid The unique object name shall identify the period created by the rtems_rate_monotonic_create call.

Invalid The unique object name shall not identify a period.

IdVar

- Set** The value of the object referenced by the id parameter shall be set to the object identifier of the created period after the return of the rtems_rate_monotonic_create call.
- Nop** Objects referenced by the id parameter in past calls to rtems_rate_monotonic_create shall not be accessed by the rtems_rate_monotonic_create call.

5.1.243.3 transition-map

Status = *Ok*, Name = *Valid*, IdVar = *Set*

- Name = *Valid*, Id = *Valid*, Free = *Yes*

Status = *TooMany*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Free = *No*

Status = *InvAddr*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Null*, Free = { *Yes*, *No* }

Status = *InvName*, Name = *Invalid*, IdVar = *Nop*

- Name = *Invalid*, Id = { *Valid*, *Null* }, Free = { *Yes*, *No* }

5.1.244 spec:/rtems/ratemon/req/delete

spec:/rtems/ratemon/req/delete

When the rtems_rate_monotonic_delete() (see: spec:/rtems/ratemon/if/delete) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicRatemon - rtems_rate_monotonic_delete

5.1.244.1 pre-conditions

Id

NoObj While the id parameter is not associated with a period.

Period While the id parameter is associated with a period.

5.1.244.2 post-conditions

Status

Ok The return status of rtems_rate_monotonic_delete shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_rate_monotonic_delete shall be RTEMS_INVALID_ID.

Name

Valid The unique object name shall identify a period.

Invalid The unique object name shall not identify a period.

5.1.244.3 transition-map

Status = *InvId*, Name = *Valid*

- Id = *NoObj*

Status = *Ok*, Name = *Invalid*

- Id = *Period*

5.1.245 spec:/rtems/ratemon/req/get-status

spec:/rtems/ratemon/req/get-status

When the rtems_rate_monotonic_get_status() (see: spec:/rtems/ratemon/if/get-status) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicRatemon - rtems_rate_monotonic_get_status

5.1.245.1 pre-conditions

StatusAddr

Valid While the status parameter references an object of type rtems_rate_monotonic_period_status.

Null While the status parameter is NULL.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

State

Inactive While the id parameter references an period object in inactive state.

Active While the id parameter references an period object in active state.

Expired While the id parameter references an period object in expired state.

Elapsed

Time While a certain time of the CLOCK_MONOTONIC has elapsed.

Consumed

CpuTime While the owner task has consumed a certain amount of CPU time.

Postponed

Zero While the period is not in expired state.

One While there is one postponed job.

Several While there are two or more postponed job.

5.1.245.2 post-conditions

Status

Ok The return status of rtems_rate_monotonic_get_status shall be RTEMS_SUCCESSFUL

InvAddr The return status of rtems_rate_monotonic_get_status shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_rate_monotonic_get_status shall be RTEMS_INVALID_ID.

Owner

OwnerTask The value of the member owner of the object referenced by the status parameter shall be set to the object identifier of the owner task of the period after the return of the rtems_rate_monotonic_get_status call.

Nop Objects referenced by the status parameter in past call to rtems_rate_monotonic_get_status shall not be accessed by the rtems_rate_monotonic_get_status call (see also Nop).

State

Inactive The value of the member state of the object referenced by the status parameter shall be set to RATE_MONOTONIC_INACTIVE after the return of the rtems_rate_monotonic_get_status call. (See also inactive)

Active The value of the member state of the object referenced by the status parameter shall be set to RATE_MONOTONIC_ACTIVE after the return of the rtems_rate_monotonic_get_status call. (See also active)

Expired The value of the member state of the object referenced by the status parameter shall be set to RATE_MONOTONIC_EXPIRED after the return of the rtems_rate_monotonic_get_status call. (See also expired)

Nop Objects referenced by the status parameter in past calls to rtems_rate_monotonic_get_status shall not be accessed by the rtems_rate_monotonic_get_status call (see also Nop).

Elapsed

Time The value of the member since_last_period of the object referenced by the status parameter shall be set to the time elapsed.

Zero The value of the member since_last_period of the object referenced by the status parameter shall be set to 0.

Nop Objects referenced by the status parameter in past calls to rtems_rate_monotonic_get_status shall not be accessed by the rtems_rate_monotonic_get_status call (see also Nop).

Consumed

CpuTime The value of the member executed_since_last_period of the object referenced by the status parameter shall be set to the CPU time consumed by the owner task.

Zero The value of the member since_last_period of the object referenced by the status parameter shall be set to 0.

Nop Objects referenced by the status parameter in past calls to rtems_rate_monotonic_get_status shall not be accessed by the rtems_rate_monotonic_get_status call (see also Nop).

Postponed

Zero The value of the member postponed_jobs_count of the object referenced by the status parameter shall be set to 0 after the return of the rtems_rate_monotonic_get_status call.

One The value of the member postponed_jobs_count of the object referenced by the status parameter shall be set to the number of postponed jobs (here 1) after the return of the rtems_rate_monotonic_get_status call.

Several The value of the member postponed_jobs_count of the object referenced by the status parameter shall be set to the number of postponed jobs after the return of the rtems_rate_monotonic_get_status call.

Nop Objects referenced by the status parameter in past calls to rtems_rate_monotonic_get_status shall not be accessed by the rtems_rate_monotonic_get_status call (see also Nop).

5.1.245.3 skip-reasons

NotInInactiveState postponed jobs do not exist in inactive state.

NeverInExpiredState There must be postponed jobs in expired state.

5.1.245.4 transition-map

Status = *Ok*, Owner = *OwnerTask*, State = *Inactive*, Elapsed = *Zero*, Consumed = *Zero*, Postponed = NA

- StatusAddr = *Valid*, Id = *Valid*, State = *Inactive*, Elapsed = NA, Consumed = NA, Postponed = NA

Status = *Ok*, Owner = *OwnerTask*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Zero*

- StatusAddr = *Valid*, Id = *Valid*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Zero*

Status = *Ok*, Owner = *OwnerTask*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *One*

- StatusAddr = *Valid*, Id = *Valid*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *One*

Status = *Ok*, Owner = *OwnerTask*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Several*

- StatusAddr = *Valid*, Id = *Valid*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Several*

Status = *Ok*, Owner = *OwnerTask*, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *One*

- StatusAddr = *Valid*, Id = *Valid*, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *One*

Status = *Ok*, Owner = *OwnerTask*, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Several*

- StatusAddr = *Valid*, Id = *Valid*, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Several*

Status = *InvId*, Owner = *Nop*, State = *Nop*, Elapsed = *Nop*, Consumed = *Nop*, Postponed = *Nop*

- StatusAddr = *Valid*, Id = *Invalid*, State = *Inactive*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = NA
- StatusAddr = *Valid*, Id = *Invalid*, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = { *Zero*, *One*, *Several* }
- StatusAddr = *Valid*, Id = *Invalid*, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = { *One*, *Several* }

Status = *InvAddr*, Owner = *Nop*, State = *Nop*, Elapsed = *Nop*, Consumed = *Nop*, Postponed = *Nop*

- StatusAddr = *Null*, Id = { *Valid*, *Invalid* }, State = *Inactive*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = NA
- StatusAddr = *Null*, Id = { *Valid*, *Invalid* }, State = *Active*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = { *Zero*, *One*, *Several* }
- StatusAddr = *Null*, Id = { *Valid*, *Invalid* }, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = { *One*, *Several* }

NotInInactiveState

- StatusAddr = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, State = *Inactive*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = { *One*, *Several* }

NeverInExpiredState

- StatusAddr = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, State = *Expired*, Elapsed = *Time*, Consumed = *CpuTime*, Postponed = *Zero*

5.1.246 spec:/rtems/ratemon/req/ident

spec:/rtems/ratemon/req/ident

The rtems_rate_monotonic_ident directive shall identify an Classic API rate monotonic class object by its name as specified by /rtems/req/ident-local.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicRatemon - rtems_rate_monotonic_ident

5.1.247 spec:/rtems/ratemon/req/period

spec:/rtems/ratemon/req/period

When the rtems_rate_monotonic_period() (see: spec:/rtems/ratemon/if/period) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicRatemon - rtems_rate_monotonic_period

Traced design component: RTEMSAPIClassicRatemon - RTEMS_PERIOD_STATUS

5.1.247.1 pre-conditions

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Caller

OwnerTask While the task invoking rtems_rate_monotonic_period is the task which created the period - the owner task.

OtherTask While the task invoking rtems_rate_monotonic_period is not the owner task.

Length

Ticks While the length parameter is a number larger than 0.

Note:

- RTEMS_PERIOD_STATUS == 0
- The length parameter of all calls to rtems_rate_monotonic_period must have the same value (see interval).

Status While the length parameter is RTEMS_PERIOD_STATUS.

State

Inactive While the id parameter references an period object in inactive state.

Active While the id parameter references an period object in active state.

Expired While the id parameter references an period object in expired state.

Postponed

Zero While there is no postponed job.

One While there is one postponed job.

Several While there are two or more postponed jobs.

InactiveCause

New While rtems_rate_monotonic_period has never been invoked with result RTEMS_SUCCESSFUL on the period object referenced by the id parameter since that period object has been created.

Canceled While rtems_rate_monotonic_period has never been invoked with result RTEMS_SUCCESSFUL on the period object referenced by the id parameter since that period object has been canceled using rtems_rate_monotonic_cancel.

5.1.247.2 post-conditions

Status

- Ok** The return status of rtems_rate_monotonic_period shall be RTEMS_SUCCESSFUL.
- InvId** The return status of rtems_rate_monotonic_period shall be RTEMS_INVALID_ID.
- NotOwn** The return status of rtems_rate_monotonic_period shall be RTEMS_NOT_OWNER_OF_RESOURCE.
- NotDef** The return status of rtems_rate_monotonic_period shall be RTEMS_NOT_DEFINED.
- TimeOut** The return status of rtems_rate_monotonic_period shall be RTEMS_TIMEOUT.

State

- Inactive** The state of the period shall be inactive after the return of the rtems_rate_monotonic_period call.
- Active** The state of the period shall be RATE_MONOTONIC_ACTIVE after the return of the rtems_rate_monotonic_period call.
- Expired** The state of the period shall be RATE_MONOTONIC_EXPIRED after the return of the rtems_rate_monotonic_period call.
- Nop** Objects referenced by the id parameter in past calls to rtems_rate_monotonic_period shall not be accessed by the rtems_rate_monotonic_period call (see also Nop).

Postponed

- Zero** There shall be no postponed jobs after the return of the rtems_rate_monotonic_period call.
- OneOrMore** The number of postponed jobs shall be the number of deadlines passed minus the number of returned calls to rtems_rate_monotonic_period.
- The last call to rtems_rate_monotonic_period where the state changes from inactive to RATE_MONOTONIC_ACTIVE is counted as the first returned call. The first deadline occurred at a point in time during that call to rtems_rate_monotonic_period.
- Nop** Objects referenced by the id parameter in past calls to rtems_rate_monotonic_period shall not be accessed by the rtems_rate_monotonic_period call (see also Nop).

Delay

- None** The last call to rtems_rate_monotonic_period shall return without delay.
- TillDeadline** The last call to rtems_rate_monotonic_period shall block the owner task till the next deadline and return afterwards.

Scheduler

Called The last call of the rtems_rate_monotonic_period function shall execute the release_job scheduler operation of the home scheduler.

Nop The last call of the rtems_rate_monotonic_period function shall not execute any scheduler operation.

5.1.247.3 skip-reasons

NeverInExpiredState There must be postponed jobs in expired state.

5.1.247.4 transition-map

Status = *Ok*, State = *Active*, Postponed = *Zero*, Delay = *TillDeadline*, Scheduler = *Called*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Ticks*, State = *Active*, Postponed = *Zero*, InactiveCause = NA

Status = *TimeOut*, State = *Active*, Postponed = *Zero*, Delay = *None*, Scheduler = *Called*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Ticks*, State = { *Active*, *Expired* }, Postponed = *One*, InactiveCause = NA

Status = *TimeOut*, State = *Active*, Postponed = *OneOrMore*, Delay = *None*, Scheduler = *Called*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Ticks*, State = { *Active*, *Expired* }, Postponed = *Several*, InactiveCause = NA

Status = *TimeOut*, State = *Nop*, Postponed = *Nop*, Delay = *None*, Scheduler = *Nop*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Status*, State = *Expired*, Postponed = { *One*, *Several* }, InactiveCause = NA

Status = *Ok*, State = *Active*, Postponed = *Zero*, Delay = *None*, Scheduler = *Called*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Ticks*, State = *Inactive*, Postponed = NA, InactiveCause = *New*
- Id = *Valid*, Caller = *OwnerTask*, Length = *Ticks*, State = *Inactive*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = *Canceled*

Status = *NotDef*, State = *Nop*, Postponed = *Nop*, Delay = *None*, Scheduler = *Nop*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Status*, State = *Inactive*, Postponed = NA, InactiveCause = *New*
- Id = *Valid*, Caller = *OwnerTask*, Length = *Status*, State = *Inactive*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = *Canceled*

Status = *Ok*, State = *Nop*, Postponed = *Nop*, Delay = *None*, Scheduler = *Nop*

- Id = *Valid*, Caller = *OwnerTask*, Length = *Status*, State = *Active*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = NA

Status = *NotOwn*, State = *Nop*, Postponed = *Nop*, Delay = *None*, Scheduler = *Nop*

- Id = *Valid*, Caller = *OtherTask*, Length = { *Ticks*, *Status* }, State = *Inactive*, Postponed = NA, InactiveCause = *New*
- Id = *Valid*, Caller = *OtherTask*, Length = { *Ticks*, *Status* }, State = *Inactive*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = *Canceled*
- Id = *Valid*, Caller = *OtherTask*, Length = { *Ticks*, *Status* }, State = *Active*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = NA
- Id = *Valid*, Caller = *OtherTask*, Length = { *Ticks*, *Status* }, State = *Expired*, Postponed = { *One*, *Several* }, InactiveCause = NA

Status = *InvId*, State = *Nop*, Postponed = *Nop*, Delay = *None*, Scheduler = *Nop*

- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, Length = { *Ticks*, *Status* }, State = *Inactive*, Postponed = NA, InactiveCause = *New*
- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, Length = { *Ticks*, *Status* }, State = *Inactive*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = *Canceled*
- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, Length = { *Ticks*, *Status* }, State = *Active*, Postponed = { *Zero*, *One*, *Several* }, InactiveCause = NA
- Id = *Invalid*, Caller = { *OwnerTask*, *OtherTask* }, Length = { *Ticks*, *Status* }, State = *Expired*, Postponed = { *One*, *Several* }, InactiveCause = NA

NeverInExpiredState

- Id = { *Valid*, *Invalid* }, Caller = { *OwnerTask*, *OtherTask* }, Length = { *Ticks*, *Status* }, State = *Expired*, Postponed = *Zero*, InactiveCause = NA

5.1.248 spec:/rtems/ratemon/req/timeout

spec:/rtems/ratemon/req/timeout

When the rate monotonic period timer expires.

rationale: N/A

functional-type: action

5.1.248.1 pre-conditions

WaitFor

PeriodSelf While the owner task of the period waits for the period.

PeriodOther While the owner task of the period waits for another period.

Other While the owner task of the period does not wait for a period.

WaitState

Blocked While the owner task is in the blocked wait state.

IntendToBlock While the owner task is in the intend to block wait state.

PostponedJobs

Zero While the count of postponed jobs is equal to zero.

NotZeroOrMax While the count of postponed jobs is not equal to zero or `UINT32_MAX`.

Max While the count of postponed jobs is equal to `UINT32_MAX`.

5.1.248.2 post-conditions

PostponedJobs

Nop The count of postponed jobs of the period shall not be modified.

PlusOne The count of postponed jobs of the period shall be incremented by one.

ReleaseJob

Yes The owner task of the period shall release a job with a deadline equal to the clock tick plus the next period length by the timeout operation.

No The owner task of the period shall not release a job by the timeout operation.

Unblock

Yes The owner task of the period shall be unblocked by the timeout operation.

No The owner task of the period shall not be unblocked by the timeout operation.

PeriodState

Active The period state shall be active.

Expired The period state shall be expired.

Timer

Active The timeout timer of the period shall expire at the current clock tick plus the next period length.

Uptime

Nop The period initiated `CLOCK_MONOTONIC` value shall not be modified.

Set The period initiated `CLOCK_MONOTONIC` value shall be set to the `CLOCK_MONOTONIC` at some time point during the timeout operation.

CPUUsage

Nop The period initiated CPU usage of the owner task value shall not be modified.

Set The period initiated CPU usage of the owner task value shall be set to the CPU usage of the owner task at some time point during the timeout operation.

5.1.248.3 skip-reasons

WaitForPeriodZeroPostponedJobs A thread can only wait for a period if its postponed jobs counter is zero.

5.1.248.4 transition-map

PostponedJobs = *Nop*, ReleaseJob = *Yes*, Unblock = *Yes*, PeriodState = *Active*, Timer = *Active*, Uptime = *Set*, CPUUsage = *Set*

- WaitFor = *PeriodSelf*, WaitState = *Blocked*, PostponedJobs = *Zero*

PostponedJobs = *Nop*, ReleaseJob = *Yes*, Unblock = *No*, PeriodState = *Active*, Timer = *Active*, Uptime = *Set*, CPUUsage = *Set*

- WaitFor = *PeriodSelf*, WaitState = *IntendToBlock*, PostponedJobs = *Zero*

PostponedJobs = *Nop*, ReleaseJob = *No*, Unblock = *No*, PeriodState = *Expired*, Timer = *Active*, Uptime = *Nop*, CPUUsage = *Nop*

- WaitFor = { *PeriodOther*, *Other* }, WaitState = NA, PostponedJobs = *Max*

PostponedJobs = *PlusOne*, ReleaseJob = *No*, Unblock = *No*, PeriodState = *Expired*, Timer = *Active*, Uptime = *Nop*, CPUUsage = *Nop*

- WaitFor = { *PeriodOther*, *Other* }, WaitState = NA, PostponedJobs = { *Zero*, *NotZeroOrMax* }

WaitForPeriodZeroPostponedJobs

- WaitFor = *PeriodSelf*, WaitState = { *Blocked*, *IntendToBlock* }, PostponedJobs = { *NotZeroOrMax*, *Max* }

5.1.249 spec:/rtems/req/ident

spec:/rtems/req/ident

When the specified object identification directive is called.

rationale: N/A

functional-type: action

This requirement refines *spec:/rtems/req/group*.

Traced design component: RTEMSAPIClassicObject - RTEMS_SEARCH_ALL_NODES

Traced design component: RTEMSAPIClassicObject - RTEMS_SEARCH_LOCAL_NODE

Traced design component: RTEMSAPIClassicObject - RTEMS_SEARCH_OTHER_NODES

5.1.249.1 pre-conditions

Name

Invalid While the name parameter is not associated with an active object of the specified class .

Valid While the name parameter is associated with an active object of the specified class .

Node

Local While the node parameter is the local node number.

Remote While the node parameter is a remote node number.

Invalid While the node parameter is an invalid node number.

SearchAll While the node parameter is RTEMS_SEARCH_ALL_NODES.

SearchOther While the node parameter is RTEMS_SEARCH_OTHER_NODES.

SearchLocal While the node parameter is RTEMS_SEARCH_LOCAL_NODE.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

5.1.249.2 post-conditions

Status

Ok The return status shall be RTEMS_SUCCESSFUL.

InvAddr The return status shall be RTEMS_INVALID_ADDRESS.

InvName The return status shall be RTEMS_INVALID_NAME.

InvNode The return status shall be RTEMS_INVALID_NODE.

Id

Nop The value of the object identifier referenced by the id parameter shall be the value before the action.

Null While the id is NULL.

LocalObj The value of the object identifier referenced by the id parameter shall be the identifier of a local object of the specified class with a name equal to the name parameter. If more than one local object of the specified class with such a name exists, then it shall be the identifier of the object with the lowest object index.

RemoteObj The value of the object identifier referenced by the id parameter shall be the identifier of a remote object of the specified class on a eligible node defined by the node parameter with a name equal to the name parameter. If more than one local object of the specified class with such a name exists, then it shall be the identifier of the object with the lowest object index. Otherwise, if more than one object of the specified class with such a name exists on remote eligible nodes, then it shall be the identifier of the object with the lowest node index and the lowest object index on this node.

5.1.249.3 transition-map

Status = *Ok*, Id = *LocalObj*

- Name = *Valid*, Node = { *Local*, *SearchAll*, *SearchLocal* }, Id = *Valid*

Status = *InvName*, Id = *Nop*

- Name = *Invalid*, Node = { *Local*, *Remote*, *Invalid*, *SearchAll*, *SearchOther*, *SearchLocal* }, Id = *Valid*
- Name = *Valid*, Node = { *Remote*, *Invalid*, *SearchOther* }, Id = *Valid*

Status = *InvAddr*, Id = *Null*

- Name = { *Invalid*, *Valid* }, Node = { *Local*, *Remote*, *Invalid*, *SearchAll*, *SearchOther*, *SearchLocal* }, Id = *Null*

5.1.250 spec:/rtems/req/ident-local

spec:/rtems/req/ident-local

When the specified object identification directive is called.

rationale: N/A

functional-type: action

This requirement refines *spec:/rtems/req/group*.

5.1.250.1 pre-conditions

Name

Invalid While the name parameter is not associated with an active object of the specified class .

Valid While the name parameter is associated with an active object of the specified class .

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

5.1.250.2 post-conditions

Status

Ok The return status shall be RTEMS_SUCCESSFUL.

InvAddr The return status shall be RTEMS_INVALID_ADDRESS.

InvName The return status shall be RTEMS_INVALID_NAME.

Id

Nop The value of the object identifier referenced by the id parameter shall be the value before the action.

Null While the id is NULL.

Id The value of the object identifier referenced by the id parameter shall be the identifier of a local object of the specified class with a name equal to the name parameter. If more than one local object of the specified class with such a name exists, then it shall be the identifier of the object with the lowest object index.

5.1.250.3 transition-map

Status = *InvName*, Id = *Nop*

- Name = *Invalid*, Id = *Valid*

Status = *Ok*, Id = *Id*

- Name = *Valid*, Id = *Valid*

Status = *InvAddr*, Id = *Null*

- Name = { *Invalid*, *Valid* }, Id = *Null*

5.1.251 spec:/rtems/scheduler/req/add-processor

spec:/rtems/scheduler/req/add-processor

When the rtems_scheduler_add_processor() (see: spec:/rtems/scheduler/if/add-processor) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_add_processor

5.1.251.1 pre-conditions

HasReady

Ready While the scheduler has at least one ready thread.

Empty While the scheduler has no ready threads.

Id

Invalid While the scheduler_id parameter is not associated with a scheduler.

Scheduler While the scheduler_id parameter is associated with a scheduler.

CPUIndex

Valid While the cpu_index parameter is less than the configured processor maximum.

Invalid While the cpu_index parameter is greater than or equal to the configured processor maximum.

CPUSTate

Idle While the processor associated with the cpu_index parameter is configured to be used by a scheduler, while the processor associated with the cpu_index parameter is online, while the processor associated with the cpu_index parameter is not owned by a scheduler.

InUse While the processor associated with the cpu_index parameter is owned by a scheduler.

NotOnline While the processor associated with the cpu_index parameter is not online.

NotUsable While the processor associated with the cpu_index parameter is not configured to be used by a scheduler.

5.1.251.2 post-conditions

Status

Ok The return status of rtems_scheduler_add_processor shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_scheduler_add_processor shall be RTEMS_INVALID_ID.

NotConf The return status of rtems_scheduler_add_processor shall be RTEMS_NOT_CONFIGURED.

IncStat The return status of rtems_scheduler_add_processor shall be RTEMS_INCORRECT_STATE.

InUse The return status of rtems_scheduler_add_processor shall be RTEMS_RESOURCE_IN_USE.

Added

Yes The processor specified by the `cpu_index` parameter shall be added to the scheduler specified by the `scheduler_id` by the `rtems_scheduler_add_processor` call.

Nop No processor shall be added to a scheduler by the `rtems_scheduler_add_processor` call.

5.1.251.3 skip-reasons

OnlyOneCPU Where the system is build with SMP support disabled, the system has exactly one processor and there is no other processor available to add to a scheduler.

5.1.251.4 transition-map

Status = *Ok*, Added = *Yes*

- HasReady = { *Ready*, *Empty* }, Id = *Scheduler*, CPUIndex = *Valid*, CPUState = *Idle*

Status = *InUse*, Added = *Nop*

- HasReady = { *Ready*, *Empty* }, Id = *Scheduler*, CPUIndex = *Valid*, CPUState = *InUse*

Status = *IncStat*, Added = *Nop*

- HasReady = { *Ready*, *Empty* }, Id = *Scheduler*, CPUIndex = *Valid*, CPUState = *NotOnline*

Status = *NotConf*, Added = *Nop*

- HasReady = { *Ready*, *Empty* }, Id = *Scheduler*, CPUIndex = *Valid*, CPUState = *NotUsable*
- HasReady = { *Ready*, *Empty* }, Id = *Scheduler*, CPUIndex = *Invalid*, CPUState = NA

Status = *InvId*, Added = *Nop*

- HasReady = { *Ready*, *Empty* }, Id = *Invalid*, CPUIndex = *Valid*, CPUState = { *Idle*, *InUse*, *NotOnline*, *NotUsable* }
- HasReady = { *Ready*, *Empty* }, Id = *Invalid*, CPUIndex = *Invalid*, CPUState = NA

5.1.252 spec:/rtems/scheduler/req/get-maximum-priority

spec:/rtems/scheduler/req/get-maximum-priority

When the `rtems_scheduler_get_maximum_priority()` (see: spec:/rtems/scheduler/if/get-maximum-priority) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - `rtems_scheduler_get_maximum_priority`

5.1.252.1 pre-conditions

Id

Invalid While the scheduler_id parameter is not associated with a scheduler.

Scheduler While the scheduler_id parameter is associated with a scheduler.

Prio

Valid While the priority parameter references an object of type rtems_task_priority.

Null While the priority parameter is equal to NULL.

5.1.252.2 post-conditions

Status

Ok The return status of rtems_scheduler_get_maximum_priority shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_scheduler_get_maximum_priority shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_scheduler_get_maximum_priority shall be RTEMS_INVALID_ID.

PrioObj

Set The value of the object referenced by the priority parameter shall be set to the maximum priority value of the scheduler specified by the scheduler_id parameter after the return of the rtems_scheduler_get_maximum_priority call.

Nop Objects referenced by the priority parameter in past calls to rtems_scheduler_get_maximum_priority shall not be accessed by the rtems_scheduler_get_maximum_priority call.

5.1.252.3 transition-map

Status = *InvId*, PrioObj = *Nop*

- Id = *Invalid*, Prio = *Valid*

Status = *Ok*, PrioObj = *Set*

- Id = *Scheduler*, Prio = *Valid*

Status = *InvAddr*, PrioObj = *Nop*

- Id = { *Invalid*, *Scheduler* }, Prio = *Null*

5.1.253 spec:/rtems/scheduler/req/get-processor

spec:/rtems/scheduler/req/get-processor

When the rtems_scheduler_get_processor directive is called, the directive shall return the return value of a call to rtems_scheduler_get_processor.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_get_processor

5.1.254 spec:/rtems/scheduler/req/get-processor-maximum

spec:/rtems/scheduler/req/get-processor-maximum

When the rtems_scheduler_get_processor_maximum directive is called, the directive shall return the return value of a call to rtems_scheduler_get_processor_maximum.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_get_processor_maximum

5.1.255 spec:/rtems/scheduler/req/get-processor-maximum-smp-only

spec:/rtems/scheduler/req/get-processor-maximum-smp-only

The return value of the rtems_scheduler_get_processor_maximum directive shall be the minimum of the processors available at the target and the CONFIGURE_MAXIMUM_PROCESSORS application configuration option value.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_get_processor_maximum

5.1.256 spec:/rtems/scheduler/req/get-processor-set

spec:/rtems/scheduler/req/get-processor-set

When the rtems_scheduler_get_processor_set() (see: spec:/rtems/scheduler/if/get-processor-set) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_get_processor_set

5.1.256.1 pre-conditions

Id

Invalid While the scheduler_id parameter is not associated with a scheduler.

Scheduler While the scheduler_id parameter is associated with a scheduler.

CPUSetSize

Valid While the cpusetsize parameter is an integral multiple of the size of long, while the cpusetsize parameter specifies a processor set which is large enough to contain the processor set of the scheduler.

TooSmall While the cpusetsize parameter is an integral multiple of the size of long, while the cpusetsize parameter specifies a processor set which is not large enough to contain the processor set of the scheduler.

Askew While the cpusetsize parameter is not an integral multiple of the size of long.

CPUSet

Valid While the cpuset parameter references an object of type cpu_set_t.

Null While the cpuset parameter is equal to NULL.

5.1.256.2 post-conditions

Status

Ok The return status of rtems_scheduler_get_processor_set shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_scheduler_get_processor_set shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_scheduler_get_processor_set shall be RTEMS_INVALID_ID.

InvSize The return status of `rtems_scheduler_get_processor_set` shall be `RTEMS_INVALID_SIZE`.

CPUSetVar

Set The value of the object referenced by the `cpuset` parameter shall be set to the processor set owned by the scheduler specified by the `scheduler_id` parameter at some point during the call after the return of the `rtems_scheduler_get_processor_set` call.

Nop Objects referenced by the `cpuset` parameter in past calls to `rtems_scheduler_get_processor_set` shall not be accessed by the `rtems_scheduler_get_processor_set` call.

5.1.256.3 transition-map

Status = *Ok*, CPUSetVar = *Set*

- Id = *Scheduler*, CPUSetSize = *Valid*, CPUSet = *Valid*

Status = *InvSize*, CPUSetVar = *Nop*

- Id = *Scheduler*, CPUSetSize = { *TooSmall*, *Askew* }, CPUSet = *Valid*

Status = *InvId*, CPUSetVar = *Nop*

- Id = *Invalid*, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Valid*

Status = *InvAddr*, CPUSetVar = *Nop*

- Id = { *Invalid*, *Scheduler* }, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Null*

5.1.257 spec:/rtems/scheduler/req/get-processor-smp-only

spec:/rtems/scheduler/req/get-processor-smp-only

The return value of the `rtems_scheduler_get_processor` directive shall be the index of a processor which executed at least one instruction of the directive call.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicScheduler - `rtems_scheduler_get_processor`

5.1.258 spec:/rtems/scheduler/req/ident

spec:/rtems/scheduler/req/ident

When the rtems_scheduler_ident() (see: [spec:/rtems/scheduler/if/ident](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_ident

5.1.258.1 pre-conditions

Name

Invalid While the name parameter is not associated with a scheduler.

Valid While the name parameter is associated with a scheduler.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is equal to NULL.

5.1.258.2 post-conditions

Status

Ok The return status of rtems_scheduler_ident shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_scheduler_ident shall be RTEMS_INVALID_ADDRESS.

InvName The return status of rtems_scheduler_ident shall be RTEMS_INVALID_NAME.

IdVar

Set The value of the object referenced by the id parameter shall be set to the identifier of the scheduler with the lowest scheduler index and a name equal to the name parameter after the return of the rtems_scheduler_ident call.

Nop Objects referenced by the id parameter in past calls to rtems_scheduler_ident shall not be accessed by the rtems_scheduler_ident call.

5.1.258.3 transition-map

Status = *InvName*, IdVar = *Nop*

- Name = *Invalid*, Id = *Valid*

Status = *Ok*, IdVar = *Set*

- Name = *Valid*, Id = *Valid*

Status = *InvAddr*, IdVar = *Nop*

- Name = { *Invalid*, *Valid* }, Id = *Null*

5.1.259 spec:/rtems/scheduler/req/ident-by-processor

spec:/rtems/scheduler/req/ident-by-processor

When the rtems_scheduler_ident_by_processor() (see: [spec:/rtems/scheduler/if/ident-by-processor](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_ident_by_processor

5.1.259.1 pre-conditions

CPUOwnedByScheduler

Yes While the processor specified by the cpu_index parameter is owned by a scheduler.

No While the processor specified by the cpu_index parameter is not owned by a scheduler.

CPUIndex

Invalid While the cpu_index parameter greater than or equal to the processor maximum.

Valid While the cpu_index parameter less than the processor maximum.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is equal to NULL.

5.1.259.2 post-conditions

Status

Ok The return status of rtems_scheduler_ident_by_processor shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_scheduler_ident_by_processor shall be RTEMS_INVALID_ADDRESS.

InvName The return status of rtems_scheduler_ident_by_processor shall be RTEMS_INVALID_NAME.

IncStat The return status of rtems_scheduler_ident_by_processor shall be RTEMS_INVALID_NAME.

IdVar

Set The value of the object referenced by the id parameter shall be set to the identifier of the scheduler which owned the processor specified by the cpu_index parameter at some point during the call after the return of the rtems_scheduler_ident_by_processor call.

Nop Objects referenced by the id parameter in past calls to rtems_scheduler_ident_by_processor shall not be accessed by the rtems_scheduler_ident_by_processor call.

5.1.259.3 skip-reasons

NoSecondCPU Where the system is build with SMP support disabled, the system has exactly one processor available and this processor is always owned by a scheduler.

5.1.259.4 transition-map

Status = *Ok*, IdVar = *Set*

- CPUOwnedByScheduler = *Yes*, CPUIndex = *Valid*, Id = *Valid*

Status = *IncStat*, IdVar = *Nop*

- CPUOwnedByScheduler = *No*, CPUIndex = *Valid*, Id = *Valid*

Status = *InvName*, IdVar = *Nop*

- CPUOwnedByScheduler = *NA*, CPUIndex = *Invalid*, Id = *Valid*

Status = *InvAddr*, IdVar = *Nop*

- CPUOwnedByScheduler = *NA*, CPUIndex = *Invalid*, Id = *Null*
- CPUOwnedByScheduler = { *Yes*, *No* }, CPUIndex = *Valid*, Id = *Null*

5.1.260 spec:/rtems/scheduler/req/ident-by-processor-set

spec:/rtems/scheduler/req/ident-by-processor-set

When the rtems_scheduler_ident_by_processor_set() (see: spec:/rtems/scheduler/if/ident-by-processor-set) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_ident_by_processor_set

5.1.260.1 pre-conditions

CPUOwnedByScheduler

Yes While the highest numbered online processor specified by the processor set is owned by a scheduler.

No While the highest numbered online processor specified by the processor set is not owned by a scheduler.

CPUSetObj

Invalid While the processor set contains no online processor.

Valid While the processor set contains at least one online processor.

CPUSetSize

Valid While the cpusetsize parameter is an integral multiple of the size of long.

Invalid While the cpusetsize parameter is not an integral multiple of the size of long.

CPUSet

Valid While the cpuset parameter references an object of type cpu_set_t.

Null While the cpuset parameter is equal to NULL.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is equal to NULL.

5.1.260.2 post-conditions

Status

Ok The return status of rtems_scheduler_ident_by_processor_set shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_scheduler_ident_by_processor_set shall be RTEMS_INVALID_ADDRESS.

InvSize The return status of rtems_scheduler_ident_by_processor_set shall be RTEMS_INVALID_SIZE.

InvName The return status of rtems_scheduler_ident_by_processor_set shall be RTEMS_INVALID_NAME.

IncStat The return status of rtems_scheduler_ident_by_processor_set shall be RTEMS_INVALID_NAME.

IdVar

Set The value of the object referenced by the id parameter shall be set to the identifier of the scheduler which owned the highest numbered online processor specified by the cpusetsize cpuset parameters at some point during the call after the return of the rtems_scheduler_ident_by_processor_set call.

Nop Objects referenced by the id parameter in past calls to rtems_scheduler_ident_by_processor_set shall not be accessed by the rtems_scheduler_ident_by_processor_set call.

5.1.260.3 skip-reasons

OnlyOneCPU Where the system is build with SMP support disabled, the system has exactly one processor available and this processor is always owned by a scheduler.

5.1.260.4 transition-map

Status = *Ok*, IdVar = *Set*

- CPUOwnedByScheduler = *Yes*, CPUSetObj = *Valid*, CPUSetSize = *Valid*, CPUSet = *Valid*, Id = *Valid*

Status = *IncStat*, IdVar = *Nop*

- CPUOwnedByScheduler = *No*, CPUSetObj = *Valid*, CPUSetSize = *Valid*, CPUSet = *Valid*, Id = *Valid*

Status = *InvName*, IdVar = *Nop*

- CPUOwnedByScheduler = NA, CPUSetObj = *Invalid*, CPUSetSize = *Valid*, CPUSet = *Valid*, Id = *Valid*

Status = *InvSize*, IdVar = *Nop*

- CPUOwnedByScheduler = NA, CPUSetObj = *Invalid*, CPUSetSize = *Invalid*, CPUSet = *Valid*, Id = *Valid*
- CPUOwnedByScheduler = { *Yes*, *No* }, CPUSetObj = *Valid*, CPUSetSize = *Invalid*, CPUSet = *Valid*, Id = *Valid*

Status = *InvAddr*, IdVar = *Nop*

- CPUOwnedByScheduler = NA, CPUSetObj = *Invalid*, CPUSetSize = { *Valid*, *Invalid* }, CPUSet = *Valid*, Id = *Null*
- CPUOwnedByScheduler = NA, CPUSetObj = *Invalid*, CPUSetSize = { *Valid*, *Invalid* }, CPUSet = *Null*, Id = { *Valid*, *Null* }
- CPUOwnedByScheduler = { *Yes*, *No* }, CPUSetObj = *Valid*, CPUSetSize = { *Valid*, *Invalid* }, CPUSet = *Valid*, Id = *Null*
- CPUOwnedByScheduler = { *Yes*, *No* }, CPUSetObj = *Valid*, CPUSetSize = { *Valid*, *Invalid* }, CPUSet = *Null*, Id = { *Valid*, *Null* }

5.1.261 spec:/rtems/scheduler/req/remove-processor

spec:/rtems/scheduler/req/remove-processor

When the rtems_scheduler_remove_processor() (see: spec:/rtems/scheduler/if/remove-processor) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicScheduler - rtems_scheduler_remove_processor

5.1.261.1 pre-conditions

Id

Invalid While the scheduler_id parameter is not associated with a scheduler.

Scheduler While the scheduler_id parameter is associated with a scheduler.

CPUIndex

Valid While the cpu_index parameter is less than the configured processor maximum.

Invalid While the cpu_index parameter is greater than or equal to the configured processor maximum.

Owned

Yes While the processor specified by the cpu_index parameter is owned by the scheduler specified by the scheduler_id parameter.

No While the processor specified by the `cpu_index` parameter is not owned by the scheduler specified by the `scheduler_id` parameter.

Last

Yes While the processor specified by the `cpu_index` parameter is the last processor owned by the scheduler specified by the `scheduler_id` parameter.

No While the processor specified by the `cpu_index` parameter is not the last processor owned by the scheduler specified by the `scheduler_id` parameter.

Home

Yes While at least one non-idle task exists which uses the scheduler specified by the `scheduler_id` parameter as its home scheduler.

No While no non-idle task exists which uses the scheduler specified by the `scheduler_id` parameter as its home scheduler.

RequiredByAffinity

Yes While at least one non-idle task which uses the scheduler specified by the `scheduler_id` parameter as its home scheduler exists those processor affinity set requires the processor specified by the `cpu_index` parameter.

No While no non-idle task which uses the scheduler specified by the `scheduler_id` parameter as its home scheduler exists those processor affinity set requires the processor specified by the `cpu_index` parameter.

UsedBy

Idle While the processor specified by the `cpu_index` parameter is used by an idle task.

Task While the processor specified by the `cpu_index` parameter is used by a task task which uses the scheduler specified by the `scheduler_id` parameter as its home scheduler.

TaskIdle While the processor specified by the `cpu_index` parameter is used by an idle task on behalf of a task task which uses the scheduler specified by the `scheduler_id` parameter as its home scheduler.

Helping While the processor specified by the `cpu_index` parameter is used by a task task which uses the scheduler specified by the `scheduler_id` parameter as a helping scheduler.

5.1.261.2 post-conditions

Status

Ok The return status of `rtems_scheduler_remove_processor` shall be `RTEMS_SUCCESSFUL`.

InvId The return status of `rtems_scheduler_remove_processor` shall be `RTEMS_INVALID_ID`.

InvNum The return status of `rtems_scheduler_remove_processor` shall be `RTEMS_INVALID_NUMBER`.

InUse The return status of `rtems_scheduler_remove_processor` shall be `RTEMS_RESOURCE_IN_USE`.

Removed

Yes The processor specified by the `cpu_index` parameter shall be removed from the scheduler specified by the `scheduler_id` by the `rtems_scheduler_remove_processor` call.

Nop No processor shall be removed from a scheduler by the `rtems_scheduler_remove_processor` call.

5.1.261.3 skip-reasons

OnlyOneCPU Where the system is build with SMP support disabled, the system has exactly one processor and there is no processor available to remove from a scheduler. In addition, the scheduler helping protocol is not available.

NoHomeNoTaskUser The processor can only be used by a task if a task uses the scheduler as its home scheduler.

LastIsRequired The last processor is required by a task which uses the scheduler as its home scheduler.

5.1.261.4 transition-map

Status = *InUse*, Removed = *Nop*

- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = { *Yes*, *No* }, Home = *Yes*, RequiredByAffinity = *Yes*, UsedBy = { *Idle*, *Task*, *TaskIdle*, *Helping* }
- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = *Yes*, Home = *No*, RequiredByAffinity = NA, UsedBy = *Helping*

Status = *Ok*, Removed = *Yes*

- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = *Yes*, Home = *No*, RequiredByAffinity = NA, UsedBy = *Idle*
- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = *No*, Home = *Yes*, RequiredByAffinity = *No*, UsedBy = { *Idle*, *Task*, *TaskIdle*, *Helping* }
- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = *No*, Home = *No*, RequiredByAffinity = NA, UsedBy = { *Idle*, *Helping* }

Status = *InvNum*, Removed = *Nop*

- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *No*, Last = NA, Home = NA, RequiredByAffinity = NA, UsedBy = NA
- Id = *Scheduler*, CPUIndex = *Invalid*, Owned = NA, Last = NA, Home = NA, RequiredByAffinity = NA, UsedBy = NA

Status = *InvId*, Removed = *Nop*

- Id = *Invalid*, CPUIndex = { *Valid*, *Invalid* }, Owned = NA, Last = NA, Home = NA, RequiredByAffinity = NA, UsedBy = NA

NoHomeNoTaskUser

- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = { *Yes*, *No* }, Home = *No*, RequiredByAffinity = NA, UsedBy = { *Task*, *TaskIdle* }

LastIsRequired

- Id = *Scheduler*, CPUIndex = *Valid*, Owned = *Yes*, Last = *Yes*, Home = *Yes*, RequiredByAffinity = *No*, UsedBy = { *Idle*, *Task*, *TaskIdle*, *Helping* }

5.1.262 spec:/rtems/sem/req/create

spec:/rtems/sem/req/create

When the rtems_semaphore_create() (see: spec:/rtems/sem/if/create) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_create

5.1.262.1 pre-conditions

Name

Valid While the name parameter is valid.

Invalid While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

Count

Zero While the count parameter is zero.

One While the count parameter is one.

GtOne While the count parameter is greater than one.

Binary

Yes While the attribute_set parameter specifies the binary semaphore class.

No While the attribute_set parameter does not specify the binary semaphore class.

Simple

Yes While the attribute_set parameter specifies the simple binary semaphore class.

No While the attribute_set parameter does not specify the simple binary semaphore class.

Inherit

Yes While the attribute_set parameter specifies the priority inheritance locking protocol.

No While the attribute_set parameter does not specify the priority inheritance locking protocol.

Ceiling

Yes While the attribute_set parameter specifies the priority ceiling locking protocol.

No While the attribute_set parameter does not specify the priority ceiling locking protocol.

MrsP

Yes While the attribute_set parameter specifies the MrsP locking protocol.

No While the attribute_set parameter does not specify the MrsP locking protocol.

Disc

FIFO While the attribute_set parameter specifies the FIFO task wait queue discipline or the default task wait queue discipline.

Prio While the attribute_set parameter specifies the priority task wait queue discipline.

Prio

LeCur While the priority_ceiling parameter is a valid task priority less than or equal to the current priority of the calling task with respect to the scheduler of the calling task at some point during the directive call.

GtCur While the priority_ceiling parameter is a valid task priority greater than the current priority of the calling task with respect to the scheduler of the calling task at some point during the directive call.

Invalid The priority_ceiling parameter shall not be a valid task priority with respect to the scheduler of the calling task at some point during the directive call.

Free

Yes While the system has at least one inactive semaphore object available.

No While the system has no inactive semaphore object available.

5.1.262.2 post-conditions

Status

- Ok** The return status of rtems_semaphore_create shall be RTEMS_SUCCESSFUL.
- InvName** The return status of rtems_semaphore_create shall be RTEMS_INVALID_NAME.
- InvAddr** The return status of rtems_semaphore_create shall be RTEMS_INVALID_ADDRESS.
- InvNum** The return status of rtems_semaphore_create shall be RTEMS_INVALID_NUMBER.
- InvPrio** The return status of rtems_semaphore_create shall be RTEMS_INVALID_PRIORITY.
- NotDef** The return status of rtems_semaphore_create shall be RTEMS_NOT_DEFINED.
- TooMany** The return status of rtems_semaphore_create shall be RTEMS_TOO_MANY.

Name

Valid The unique object name shall identify the semaphore created by the rtems_semaphore_create call.

Invalid The unique object name shall not identify a semaphore.

IdVar

Set The value of the object referenced by the id parameter shall be set to the object identifier of the created semaphore after the return of the rtems_semaphore_create call.

Nop Objects referenced by the id parameter in past calls to rtems_semaphore_create shall not be accessed by the rtems_semaphore_create call.

Variant

Cnt The semaphore created by the rtems_semaphore_create call shall be a counting semaphore.

Bin The semaphore created by the rtems_semaphore_create call shall be a binary semaphore not using a locking protocol.

PI The semaphore created by the rtems_semaphore_create call shall be a binary semaphore using the priority inheritance locking protocol.

PC The semaphore created by the rtems_semaphore_create call shall be a binary semaphore using the priority ceiling locking protocol.

SB The semaphore created by the rtems_semaphore_create call shall be a simple binary semaphore.

MrsP The semaphore created by the rtems_semaphore_create call shall be a binary semaphore using the MrsP locking protocol.

Disc

FIFO The semaphore created by the rtems_semaphore_create call shall use the FIFO task wait queue discipline.

Prio The semaphore created by the rtems_semaphore_create call shall use the priority task wait queue discipline.

Count

Initial The semaphore created by the rtems_semaphore_create call shall have an initial count equal to the value of the count parameter.

Owner

Caller The semaphore created by the rtems_semaphore_create call shall be initially owned by the calling task.

No The semaphore created by the rtems_semaphore_create call shall not initially have an owner.

Prio

Ceiling The current priority of the task which called rtems_semaphore_create shall be equal to the value of the priority_ceiling parameter.

Nop The current priority of the task which called rtems_semaphore_create shall not be modified by the rtems_semaphore_create call.

5.1.262.3 transition-map

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *PC*, Disc = *Prio*, Count = NA, Owner = *Caller*, Prio = *Ceiling*

- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *Prio*, Prio = *LeCur*, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *MrsP*, Disc = *Prio*, Count = NA, Owner = *Caller*, Prio = *Ceiling*

- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *Prio*, Prio = *LeCur*, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *PC*, Disc = *Prio*, Count = NA, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *MrsP*, Disc = *Prio*, Count = NA, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *Prio*, Prio = { *LeCur*, *GtCur* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *PI*, Disc = *Prio*, Count = NA, Owner = *Caller*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *Bin*, Disc = *FIFO*, Count = NA, Owner = *Caller*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *Bin*, Disc = *Prio*, Count = NA, Owner = *Caller*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *PI*, Disc = *Prio*, Count = NA, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *Bin*, Disc = *FIFO*, Count = NA, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *Bin*, Disc = *Prio*, Count = NA, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *InvPrio*, Name = *Invalid*, IdVar = *Nop*, Variant = NA, Disc = NA, Count = NA, Owner = NA, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *Prio*, Prio = { *GtCur*, *Invalid* }, Free = *Yes*
- Name = *Valid*, Id = *Valid*, Count = *Zero*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *Prio*, Prio = { *GtCur*, *Invalid* }, Free = *Yes*
- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *Prio*, Prio = *Invalid*, Free = *Yes*
- Name = *Valid*, Id = *Valid*, Count = *One*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *Prio*, Prio = *Invalid*, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *SB*, Disc = *FIFO*, Count = *Initial*, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *No*, Simple = *Yes*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *SB*, Disc = *Prio*, Count = *Initial*, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *No*, Simple = *Yes*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *Cnt*, Disc = *FIFO*, Count = *Initial*, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *No*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *Ok*, Name = *Valid*, IdVar = *Set*, Variant = *Cnt*, Disc = *Prio*, Count = *Initial*, Owner = *No*, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *No*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *Yes*

Status = *InvNum*, Name = *Invalid*, IdVar = *Nop*, Variant = NA, Disc = NA, Count = NA, Owner = NA, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = *GtOne*, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = *GtOne*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = *GtOne*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = *GtOne*, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = *GtOne*, Binary = *No*, Simple = *Yes*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }

Status = *TooMany*, Name = *Invalid*, IdVar = *Nop*, Variant = NA, Disc = NA, Count = NA, Owner = NA, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *No*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *No*
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *No*
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *Prio*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *No*
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *No*
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One* }, Binary = *No*, Simple = { *Yes*, *No* }, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *No*

- Name = *Valid*, Id = *Valid*, Count = *GtOne*, Binary = *No*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *No*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = *No*

Status = *NotDef*, Name = *Invalid*, IdVar = *Nop*, Variant = NA, Disc = NA, Count = NA, Owner = NA, Prio = *Nop*

- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *Yes*, Inherit = { *Yes*, *No* }, Ceiling = { *Yes*, *No* }, MrsP = { *Yes*, *No* }, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *Yes*, MrsP = { *Yes*, *No* }, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *No*, MrsP = *Yes*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *Yes*, Ceiling = *No*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *Yes*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *Yes*, Simple = *No*, Inherit = *No*, Ceiling = *Yes*, MrsP = *No*, Disc = *FIFO*, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *No*, Simple = { *Yes*, *No* }, Inherit = *Yes*, Ceiling = { *Yes*, *No* }, MrsP = { *Yes*, *No* }, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *No*, Simple = { *Yes*, *No* }, Inherit = *No*, Ceiling = *Yes*, MrsP = { *Yes*, *No* }, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *No*, Simple = { *Yes*, *No* }, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }
- Name = *Valid*, Id = *Valid*, Count = { *Zero*, *One*, *GtOne* }, Binary = *No*, Simple = { *Yes*, *No* }, Inherit = *No*, Ceiling = *No*, MrsP = *Yes*, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }

Status = *InvAddr*, Name = *Invalid*, IdVar = *Nop*, Variant = NA, Disc = NA, Count = NA, Owner = NA, Prio = *Nop*

- Name = *Valid*, Id = *Null*, Count = { *Zero*, *One*, *GtOne* }, Binary = { *Yes*, *No* }, Simple = { *Yes*, *No* }, Inherit = { *Yes*, *No* }, Ceiling = { *Yes*, *No* }, MrsP = { *Yes*, *No* }, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }

Status = *InvName*, Name = *Invalid*, IdVar = *Nop*, Variant = NA, Disc = NA, Count = NA, Owner = NA, Prio = *Nop*

- Name = *Invalid*, Id = { *Valid*, *Null* }, Count = { *Zero*, *One*, *GtOne* }, Binary = { *Yes*, *No* }, Simple = { *Yes*, *No* }, Inherit = { *Yes*, *No* }, Ceiling = { *Yes*, *No* }, MrsP = { *Yes*, *No* }, Disc = { *FIFO*, *Prio* }, Prio = { *LeCur*, *GtCur*, *Invalid* }, Free = { *Yes*, *No* }

5.1.263 spec:/rtems/sem/req/delete

spec:/rtems/sem/req/delete

When the rtems_semaphore_delete() (see: [spec:/rtems/sem/if/delete](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_delete

5.1.263.1 pre-conditions

Id

NoObj While the id parameter is not associated with a semaphore.

Counting While the id parameter is associated with a counting semaphore.

Simple While the id parameter is associated with a simple binary semaphore.

Binary While the id parameter is associated with a binary semaphore.

PrioCeiling While the id parameter is associated with a priority ceiling semaphore.

PrioInherit While the id parameter is associated with a priority inheritance semaphore.

MrsP While the id parameter is associated with a MrsP semaphore.

Discipline

FIFO While the semaphore uses the FIFO task wait queue discipline.

Priority While the semaphore uses the priority task wait queue discipline.

State

GtZeroOrNoOwner While the semaphore has a count greater than zero or no owner.

Zero While the semaphore has a count of zero or an owner.

Blocked While the semaphore has tasks blocked on the semaphore.

5.1.263.2 post-conditions

Status

Ok The return status of rtems_semaphore_delete shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_semaphore_delete shall be RTEMS_INVALID_ID.

InUse The return status of rtems_semaphore_delete shall be RTEMS_RESOURCE_IN_USE.

Name

Valid The unique object name shall identify a semaphore.

Invalid The unique object name shall not identify a semaphore.

Flush

FIFO Tasks waiting at the semaphore shall be unblocked in FIFO order.

Priority Tasks waiting at the semaphore shall be unblocked in priority order.

No Tasks waiting at the semaphore shall remain blocked.

5.1.263.3 skip-reasons

NeedsPriorityDiscipline Binary semaphores with a locking protocol are required to use the priority task wait queue discipline.

5.1.263.4 transition-map

Status = *Ok*, Name = *Invalid*, Flush = *FIFO*

- Id = { *Counting*, *Simple* }, Discipline = *FIFO*, State = *Blocked*

Status = *Ok*, Name = *Invalid*, Flush = *Priority*

- Id = { *Counting*, *Simple* }, Discipline = *Priority*, State = *Blocked*

Status = *InvId*, Name = *Valid*, Flush = *No*

- Id = *NoObj*, Discipline = { *FIFO*, *Priority* }, State = { *GtZeroOrNoOwner*, *Zero*, *Blocked* }

Status = *InUse*, Name = *Valid*, Flush = *No*

- Id = *Binary*, Discipline = { *FIFO*, *Priority* }, State = { *Zero*, *Blocked* }
- Id = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *Priority*, State = { *Zero*, *Blocked* }

Status = *Ok*, Name = *Invalid*, Flush = *No*

- Id = { *Counting*, *Simple* }, Discipline = { *FIFO*, *Priority* }, State = { *GtZeroOrNoOwner*, *Zero* }
- Id = *Binary*, Discipline = { *FIFO*, *Priority* }, State = *GtZeroOrNoOwner*

- Id = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *Priority*, State = *GtZeroOrNoOwner*
NeedsPriorityDiscipline
- Id = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *FIFO*, State = { *GtZeroOrNoOwner*, *Zero*, *Blocked* }

5.1.264 spec:/rtems/sem/req/flush

spec:/rtems/sem/req/flush

When the rtems_semaphore_flush() (see: [spec:/rtems/sem/if/flush](#)) directive is called.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- [spec:/score/tq/req/flush-fifo](#)
- [spec:/score/tq/req/flush-priority](#)
- [spec:/score/tq/req/flush-priority-inherit](#)

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_flush

5.1.264.1 pre-conditions

Class

Counting While the semaphore object is a counting semaphore.

Simple While the semaphore object is a simple binary semaphore.

Binary While the semaphore object is a binary semaphore.

PrioCeiling While the semaphore object is a priority ceiling semaphore.

PrioInherit While the semaphore object is a priority inheritance semaphore.

MrsP While the semaphore object is a MrsP semaphore.

Discipline

FIFO While the semaphore uses the FIFO task wait queue discipline.

Priority While the semaphore uses the priority task wait queue discipline.

Id

Valid While the id parameter is associated with the semaphore.

Invalid While the id parameter is not associated with a semaphore.

5.1.264.2 post-conditions

Action

InvId The return status of rtems_semaphore_flush shall be RTEMS_INVALID_ID.

NotDef The return status of rtems_semaphore_flush shall be RTEMS_NOT_DEFINED.

FlushFIFO The calling task shall flush the semaphore as specified by /score/tq/req/flush-fifo.

FlushPriority The calling task shall flush the semaphore as specified by /score/tq/req/flush-priority.

FlushPriorityCeiling The calling task shall flush the semaphore as specified by /score/tq/req/flush-priority.

FlushPriorityInherit The calling task shall flush the semaphore as specified by /score/tq/req/flush-priority-inherit.

5.1.264.3 skip-reasons

NeedsPriorityDiscipline Binary semaphores with a locking protocol are required to use the priority task wait queue discipline.

NoMrsP Where the system is build with SMP support disabled, the MrsP locking protocol is not available.

5.1.264.4 transition-map

Action = *FlushPriorityCeiling*

- Class = *PrioCeiling*, Discipline = *Priority*, Id = *Valid*

Action = *FlushPriorityInherit*

- Class = *PrioInherit*, Discipline = *Priority*, Id = *Valid*

Action = *NotDef*

- Class = *MrsP*, Discipline = *Priority*, Id = *Valid*

Action = *FlushFIFO*

- Class = { *Counting*, *Simple*, *Binary* }, Discipline = *FIFO*, Id = *Valid*

Action = *FlushPriority*

- Class = { *Counting*, *Simple*, *Binary* }, Discipline = *Priority*, Id = *Valid*

Action = *InvId*

- Class = NA, Discipline = NA, Id = *Invalid*

NeedsPriorityDiscipline

- Class = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *FIFO*, Id = *Valid*

5.1.265 spec:/rtems/sem/req/ident

spec:/rtems/sem/req/ident

The rtems_semaphore_ident directive shall identify an Classic API semaphore class object by its name as specified by /rtems/req/ident.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_ident

5.1.266 spec:/rtems/sem/req/mrsp-obtain

spec:/rtems/sem/req/mrsp-obtain

When a MrsP semaphore is obtained.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_obtain

5.1.266.1 pre-conditions

Home

Idle While an idle task executes on the processor owned by the home scheduler of the obtaining task.

Task While the obtaining task executes on the processor owned by the home scheduler of the obtaining task.

TaskIdle While an idle task on behalf of the obtaining task executes on the processor owned by the home scheduler of the obtaining task.

Second While the second task executes on the processor owned by the home scheduler of the obtaining task.

SecondIdle While an idle task on behalf of the second task executes on the processor owned by the home scheduler of the obtaining task.

Helping

Idle While an idle task executes on the processor owned by the helping scheduler of the obtaining task.

Task While the obtaining task executes on the processor owned by the helping scheduler of the obtaining task.

Helping While a helping task of the obtaining task executes on the processor owned by the helping scheduler of the obtaining task.

HelpingIdle While an idle task on behalf of a helping task of the obtaining task executes on the processor owned by the helping scheduler of the obtaining task.

Third While the third task executes on the processor owned by the helping scheduler of the obtaining task.

ThirdIdle While an idle task on behalf of the third task executes on the processor owned by the helping scheduler of the obtaining task.

PriorityHome

None While no ceiling priority with respect to the home scheduler of the obtaining task is already available to the task.

NewHigh While the ceiling priority of the semaphore with respect to the home scheduler of the obtaining task is higher than the ceiling priorities already available to the task.

NewEqual While the ceiling priority of the semaphore with respect to the home scheduler of the obtaining task is equal to the ceiling priorities already available to the task.

SecondHigh While the ceiling priority of the semaphore with respect to the home scheduler of the obtaining task is higher than the priority of the second task.

SecondEqual While the ceiling priority of the semaphore with respect to the home scheduler of the obtaining task is equal to the priority of the second task.

SecondLow While the ceiling priority of the semaphore with respect to the home scheduler of the obtaining task is lower than the priority of the second task.

PriorityHelping

None While no ceiling priority with respect to the helping scheduler of the obtaining task is already available to the task.

Helping While ceiling priorities with respect to the helping scheduler of the obtaining task are already available to the task.

ThirdHigh While the ceiling priority of the semaphore with respect to the helping scheduler of the obtaining task is higher than the priority of the third task.

ThirdEqual While the ceiling priority of the semaphore with respect to the helping scheduler of the obtaining task is equal to the priority of the third task.

ThirdLow While the ceiling priority of the semaphore with respect to the helping scheduler of the obtaining task is lower than the priority of the third task.

5.1.266.2 post-conditions

Home

Task The obtaining task shall execute on the processor owned by the home scheduler of the obtaining task.

TaskIdle An idle task on behalf of the obtaining task shall execute on the processor owned by the home scheduler of the obtaining task.

Second The second task shall execute on the processor owned by the home scheduler of the obtaining task.

SecondIdle An idle task on behalf of the second task shall execute on the processor owned by the home scheduler of the obtaining task.

Helping

Idle An idle task shall execute on the processor owned by the helping scheduler of the obtaining task.

Task The obtaining task shall execute on the processor owned by the helping scheduler of the obtaining task.

TaskIdle An idle task on behalf of the obtaining task shall execute on the processor owned by the helping scheduler of the obtaining task.

Helping The helping task shall execute on the processor owned by the helping scheduler of the obtaining task.

HelpingIdle An idle task on behalf of the helping task shall execute on the processor owned by the helping scheduler of the obtaining task.

Third The third task shall execute on the processor owned by the helping scheduler of the obtaining task.

ThirdIdle An idle task on behalf of the third task shall execute on the processor owned by the helping scheduler of the obtaining task.

5.1.266.3 skip-reasons

TaskExecutesExactlyOnce The task executes on either the home or the helping scheduler.

MrsPCeilingRequired The availability of MrsP ceiling priorities depends on the ownership of a MrsP semaphore and helping tasks associated with such a semaphore.

SecondPriorityNeedsSecondTask The second task is required to have a priority relative to the second task.

ThirdPriorityNeedsThirdTask The third task is required to have a priority relative to the third task.

5.1.266.4 transition-map

Home = *Task*, Helping = *Idle*

- Home = *Task*, Helping = *Idle*, PriorityHome = *None*, PriorityHelping = *None*

Home = *Task*, Helping = *Helping*

- Home = *Task*, Helping = *Helping*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = *Helping*

Home = *Task*, Helping = *HelpingIdle*

- Home = *Task*, Helping = *HelpingIdle*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = *Helping*

Home = *Second*, Helping = *Task*

- Home = *Second*, Helping = *Task*, PriorityHome = { *None*, *SecondEqual* }, PriorityHelping = *None*
- Home = *Second*, Helping = *Task*, PriorityHome = *SecondEqual*, PriorityHelping = *Helping*
- Home = *Second*, Helping = *Task*, PriorityHome = *SecondLow*, PriorityHelping = { *None*, *Helping* }

Home = *SecondIdle*, Helping = *Task*

- Home = *SecondIdle*, Helping = *Task*, PriorityHome = { *None*, *SecondEqual* }, PriorityHelping = *None*
- Home = *SecondIdle*, Helping = *Task*, PriorityHome = *SecondEqual*, PriorityHelping = *Helping*
- Home = *SecondIdle*, Helping = *Task*, PriorityHome = *SecondLow*, PriorityHelping = { *None*, *Helping* }

Home = *TaskIdle*, Helping = *Task*

- Home = *Idle*, Helping = *Task*, PriorityHome = *None*, PriorityHelping = *None*
- Home = *TaskIdle*, Helping = *Task*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = *Helping*
- Home = { *Second*, *SecondIdle* }, Helping = *Task*, PriorityHome = *SecondHigh*, PriorityHelping = { *None*, *Helping* }

Home = *Task*, Helping = *Third*

- Home = *Task*, Helping = *Third*, PriorityHome = *None*, PriorityHelping = { *None*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = *Third*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = { *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

Home = *Task*, Helping = *ThirdIdle*

- Home = *Task*, Helping = *ThirdIdle*, PriorityHome = *None*, PriorityHelping = { *None*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = *ThirdIdle*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = { *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

TaskExecutesExactlyOnce

- Home = *Idle*, Helping = { *Idle*, *Third* }, PriorityHome = *None*, PriorityHelping = *None*
- Home = *Idle*, Helping = *Third*, PriorityHome = *None*, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = { *Idle*, *Second*, *SecondIdle* }, Helping = *ThirdIdle*, PriorityHome = *None*, PriorityHelping = { *None*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = *Task*, PriorityHome = *None*, PriorityHelping = *None*
- Home = *Task*, Helping = *Task*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = *Helping*
- Home = *TaskIdle*, Helping = { *Helping*, *HelpingIdle* }, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = *Helping*
- Home = *TaskIdle*, Helping = { *Third*, *ThirdIdle* }, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = { *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = { *Second*, *SecondIdle* }, Helping = *Idle*, PriorityHome = { *None*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *None*
- Home = { *Second*, *SecondIdle* }, Helping = *Helping*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *None*, *Helping* }
- Home = { *Second*, *SecondIdle* }, Helping = *HelpingIdle*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *Helping*
- Home = { *Second*, *SecondIdle* }, Helping = *Third*, PriorityHome = *None*, PriorityHelping = { *None*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = { *Second*, *SecondIdle* }, Helping = { *Third*, *ThirdIdle* }, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

MrsPCeilingRequired

- Home = { *Idle*, *Task*, *Second*, *SecondIdle* }, Helping = *Idle*, PriorityHome = { *None*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *Helping*
- Home = *Idle*, Helping = { *Idle*, *Task*, *Third*, *ThirdIdle* }, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = { *Idle*, *Task*, *Second*, *SecondIdle* }, Helping = { *Task*, *Third*, *ThirdIdle* }, PriorityHome = *None*, PriorityHelping = *Helping*
- Home = { *Idle*, *Second*, *SecondIdle* }, Helping = { *Helping*, *HelpingIdle* }, PriorityHome = { *None*, *NewHigh*, *NewEqual* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

- Home = { *Idle*, *Second*, *SecondIdle* }, Helping = *HelpingIdle*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *None*
- Home = { *Task*, *Second*, *SecondIdle* }, Helping = *Idle*, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = { *Task*, *Helping*, *Third*, *ThirdIdle* }, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = *None*
- Home = *Task*, Helping = { *Helping*, *HelpingIdle* }, PriorityHome = *None*, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = { *Task*, *TaskIdle* }, Helping = *HelpingIdle*, PriorityHome = { *NewHigh*, *NewEqual*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *None*
- Home = *TaskIdle*, Helping = *Idle*, PriorityHome = { *None*, *NewHigh*, *NewEqual* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *TaskIdle*, Helping = *Idle*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *None*, *Helping* }
- Home = *TaskIdle*, Helping = { *Task*, *Helping*, *HelpingIdle*, *Third*, *ThirdIdle* }, PriorityHome = *None*, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *TaskIdle*, Helping = { *Task*, *Helping*, *Third*, *ThirdIdle* }, PriorityHome = { *NewHigh*, *NewEqual*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *None*
- Home = { *Second*, *SecondIdle* }, Helping = { *Task*, *Third*, *ThirdIdle* }, PriorityHome = { *NewHigh*, *NewEqual* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

SecondPriorityNeedsSecondTask

- Home = { *Idle*, *Task* }, Helping = *Idle*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *None*
- Home = { *Idle*, *Task* }, Helping = { *Task*, *Helping* }, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *None*, *Helping* }
- Home = { *Idle*, *Task* }, Helping = *HelpingIdle*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *Helping*
- Home = { *Idle*, *Task* }, Helping = { *Third*, *ThirdIdle* }, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *None*, *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *TaskIdle*, Helping = { *Task*, *Helping*, *HelpingIdle* }, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = *Helping*
- Home = *TaskIdle*, Helping = { *Third*, *ThirdIdle* }, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *Helping*, *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

ThirdPriorityNeedsThirdTask

- Home = { *Idle*, *Second*, *SecondIdle* }, Helping = { *Idle*, *Task* }, PriorityHome = { *None*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

- Home = { *Idle*, *Second*, *SecondIdle* }, Helping = { *Helping*, *HelpingIdle* }, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = *Idle*, PriorityHome = { *None*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = *Task*, PriorityHome = { *None*, *NewHigh*, *NewEqual*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *Task*, Helping = { *Helping*, *HelpingIdle* }, PriorityHome = { *NewHigh*, *NewEqual*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *TaskIdle*, Helping = *Idle*, PriorityHome = { *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }
- Home = *TaskIdle*, Helping = { *Task*, *Helping*, *HelpingIdle* }, PriorityHome = { *NewHigh*, *NewEqual*, *SecondHigh*, *SecondEqual*, *SecondLow* }, PriorityHelping = { *ThirdHigh*, *ThirdEqual*, *ThirdLow* }

5.1.267 spec:/rtems/sem/req/mrsp-prio-change-while-waiting

spec:/rtems/sem/req/mrsp-prio-change-while-waiting

While a thread is waiting to obtain a MrsP semaphore, while its current priority changed so that it is higher than the ceiling priority of the semaphore, the thread is allowed to become the new owner of the semaphore.

rationale: While a thread has a higher current priority than the ceiling priority of a MrsP semaphore, while the semaphore has an owner thread, if the thread tries to obtain the semaphore, then an error status is returned to notify about a priority ceiling violation. This error condition is not checked if the current priority is raised to a higher priority than the ceiling priority while the thread is enqueued on the thread queue of the semaphore. It would complicate the implementation considerable to check this error condition and restore the thread state so that an error status can be returned. The error check before the thread is enqueued helps to detect application design issues. For the implementation it does not matter if the current priority is higher than a ceiling priority.

functional-type: function

5.1.268 spec:/rtems/sem/req/mrsp-uniprocessor-scheduler

spec:/rtems/sem/req/mrsp-uniprocessor-scheduler

Where an uniprocessor scheduler is configured, the owner of a semaphore created with the RTEMS_MULTIPROCESSOR_RESOURCE_SHARING attribute shall not be made sticky.

rationale: N/A

functional-type: function

5.1.269 spec:/rtems/sem/req/obtain

spec:/rtems/sem/req/obtain

When the rtems_semaphore_obtain() (see: [spec:/rtems/sem/if/obtain](#)) directive is called.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- [spec:/score mtx/req/seize-try](#)
- [spec:/score mtx/req/seize-wait](#)
- [spec:/score sem/req/seize-try](#)
- [spec:/score sem/req/seize-wait](#)

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_obtain

Traced design component: RTEMSAPIClassicTypes - RTEMS_NO_TIMEOUT

5.1.269.1 pre-conditions

Class

Counting While the semaphore object is a counting semaphore.

Simple While the semaphore object is a simple binary semaphore.

Binary While the semaphore object is a binary semaphore.

PrioCeiling While the semaphore object is a priority ceiling semaphore.

PrioInherit While the semaphore object is a priority inheritance semaphore.

MrsP While the semaphore object is a MrsP semaphore.

Discipline

FIFO While the semaphore uses the FIFO task wait queue discipline.

Priority While the semaphore uses the priority task wait queue discipline.

Id

Valid While the id parameter is associated with the semaphore.

Invalid While the id parameter is not associated with a semaphore.

Wait

No While the option_set parameter indicates the RTEMS_NO_WAIT option.

Timeout While the option_set parameter indicates the RTEMS_WAIT option, while the timeout parameter is not equal to RTEMS_NO_TIMEOUT.

Forever While the option_set parameter indicates the RTEMS_WAIT option, while the timeout parameter is equal to RTEMS_NO_TIMEOUT.

5.1.269.2 post-conditions

Action

InvId The return status of rtems_semaphore_obtain shall be RTEMS_INVALID_ID.

SemSeizeTry The calling task shall try to seize the semaphore as specified by /score/sem/req/seize-try.

SemSeizeWait The calling task shall wait to seize the semaphore as specified by /score/sem/req/seize-wait.

MtxSeizeTry The calling task shall try to seize the mutex as specified by /score/mtx/req/seize-try where an enqueue blocks, a recursive seize is allowed, and no locking protocol is used.

MtxSeizeWait The calling task shall wait to seize the mutex as specified by /score/mtx/req/seize-wait where an enqueue blocks, a recursive seize is allowed, and no locking protocol is used.

InheritMtxSeizeTry The calling task shall try to seize the mutex as specified by /score/mtx/req/seize-try where an enqueue blocks, a recursive seize is allowed, and a priority inheritance protocol is used.

InheritMtxSeizeWait The calling task shall wait to seize the mutex as specified by /score/mtx/req/seize-wait where an enqueue blocks, a recursive seize is allowed, and a priority inheritance protocol is used.

CeilingMtxSeizeTry The calling task shall try to seize the mutex as specified by /score/mtx/req/seize-try where an enqueue blocks, a recursive seize is allowed, and a priority ceiling is used.

CeilingMtxSeizeWait The calling task shall wait to seize the mutex as specified by /score/mtx/req/seize-wait where an enqueue blocks, a recursive seize is allowed, and a priority ceiling is used.

MrsPMtxSeizeTry The calling task shall try to seize the mutex as specified by /score mtx/req/seize-try where an enqueue is sticky, a recursive seize returns an error status, and a priority ceiling is used.

MrsPMtxSeizeWait The calling task shall wait to seize the mutex as specified by /score mtx/req/seize-wait where an enqueue is sticky, a recursive seize returns an error status, and a priority ceiling is used.

5.1.269.3 skip-reasons

NeedsPriorityDiscipline Binary semaphores with a locking protocol are required to use the priority task wait queue discipline.

NoMrsP Where the system is build with SMP support disabled, the MrsP locking protocol is not available.

5.1.269.4 transition-map

Action = *CeilingMtxSeizeTry*

- Class = *PrioCeiling*, Discipline = *Priority*, Id = *Valid*, Wait = *No*

Action = *InheritMtxSeizeTry*

- Class = *PrioInherit*, Discipline = *Priority*, Id = *Valid*, Wait = *No*

Action = *MrsPMtxSeizeTry*

- Class = *MrsP*, Discipline = *Priority*, Id = *Valid*, Wait = *No*

Action = *MtxSeizeTry*

- Class = *Binary*, Discipline = { *FIFO*, *Priority* }, Id = *Valid*, Wait = *No*

Action = *CeilingMtxSeizeWait*

- Class = *PrioCeiling*, Discipline = *Priority*, Id = *Valid*, Wait = { *Timeout*, *Forever* }

Action = *InheritMtxSeizeWait*

- Class = *PrioInherit*, Discipline = *Priority*, Id = *Valid*, Wait = { *Timeout*, *Forever* }

Action = *MrsPMtxSeizeWait*

- Class = *MrsP*, Discipline = *Priority*, Id = *Valid*, Wait = { *Timeout*, *Forever* }

Action = *SemSeizeTry*

- Class = { *Counting*, *Simple* }, Discipline = { *FIFO*, *Priority* }, Id = *Valid*, Wait = *No*

Action = *MtxSeizeWait*

- Class = *Binary*, Discipline = { *FIFO*, *Priority* }, Id = *Valid*, Wait = { *Timeout*, *Forever* }

Action = *SemSeizeWait*

- Class = { *Counting*, *Simple* }, Discipline = { *FIFO*, *Priority* }, Id = *Valid*, Wait = { *Timeout*, *Forever* }

Action = *InvId*

- Class = { *Counting*, *Simple*, *Binary* }, Discipline = { *FIFO*, *Priority* }, Id = *Invalid*, Wait = { *No*, *Timeout*, *Forever* }
- Class = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *Priority*, Id = *Invalid*, Wait = { *No*, *Timeout*, *Forever* }

NeedsPriorityDiscipline

- Class = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *FIFO*, Id = { *Valid*, *Invalid* }, Wait = { *No*, *Timeout*, *Forever* }

5.1.270 spec:/rtems/sem/req/release

spec:/rtems/sem/req/release

When the rtems_semaphore_release() (see: spec:/rtems/sem/if/release) directive is called.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- spec:/score/mtx/req/surrender
- spec:/score/sem/req/surrender

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_release

5.1.270.1 pre-conditions

Class

Counting While the semaphore object is a counting semaphore.

Simple While the semaphore object is a simple binary semaphore.

Binary While the semaphore object is a binary semaphore.

PrioCeiling While the semaphore object is a priority ceiling semaphore.

PrioInherit While the semaphore object is a priority inheritance semaphore.

MrsP While the semaphore object is a MrsP semaphore.

Discipline

FIFO While the semaphore uses the FIFO task wait queue discipline.

Priority While the semaphore uses the priority task wait queue discipline.

Id

Valid While the id parameter is associated with the semaphore.

Invalid While the id parameter is not associated with a semaphore.

5.1.270.2 post-conditions

Action

InvId The return status of rtems_semaphore_release shall be RTEMS_INVALID_ID.

BinarySurrender The calling task shall surrender the binary semaphore as specified by /score/sem/req/surrender.

CountingSurrender The calling task shall surrender the counting semaphore as specified by /score/sem/req/surrender.

MtxSurrender The calling task shall surrender the mutex as specified by /score mtx/req/surrender where an enqueue blocks, a recursive seize is allowed, the owner is checked, and no locking protocol is used.

InheritMtxSurrender The calling task shall surrender the mutex as specified by /score/mtx/req/surrender where an enqueue blocks, a recursive seize is allowed, the owner is checked, and a priority inheritance protocol is used.

CeilingMtxSurrender The calling task shall surrender the mutex as specified by /score/mtx/req/surrender where an enqueue blocks, a recursive seize is allowed, the owner is checked, and a priority ceiling is used.

MrsPMtxSurrender The calling task shall surrender the mutex as specified by /score/mtx/req/surrender where an enqueue is sticky, a recursive seize returns an error status, the owner is checked, and a priority ceiling is used.

5.1.270.3 skip-reasons

NeedsPriorityDiscipline Binary semaphores with a locking protocol are required to use the priority task wait queue discipline.

NoMrsP Where the system is build with SMP support disabled, the MrsP locking protocol is not available.

5.1.270.4 transition-map

Action = *CeilingMtxSurrender*

- Class = *PrioCeiling*, Discipline = *Priority*, Id = *Valid*

Action = *InheritMtxSurrender*

- Class = *PrioInherit*, Discipline = *Priority*, Id = *Valid*

Action = *MrsPMtxSurrender*

- Class = *MrsP*, Discipline = *Priority*, Id = *Valid*

Action = *CountingSurrender*

- Class = *Counting*, Discipline = { *FIFO*, *Priority* }, Id = *Valid*

Action = *BinarySurrender*

- Class = *Simple*, Discipline = { *FIFO*, *Priority* }, Id = *Valid*

Action = *MtxSurrender*

- Class = *Binary*, Discipline = { *FIFO*, *Priority* }, Id = *Valid*

Action = *InvId*

- Class = { *Counting*, *Simple*, *Binary* }, Discipline = { *FIFO*, *Priority* }, Id = *Invalid*

- Class = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *Priority*, Id = *Invalid*

NeedsPriorityDiscipline

- Class = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *FIFO*, Id = { *Valid*, *Invalid* }

5.1.271 spec:/rtems/sem/req/set-priority

spec:/rtems/sem/req/set-priority

When the rtems_semaphore_set_priority() (see: spec:/rtems/sem/if/set-priority) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_set_priority

5.1.271.1 pre-conditions

Class

Counting While the semaphore object is a counting semaphore.

Simple While the semaphore object is a simple binary semaphore.

Binary While the semaphore object is a binary semaphore.

PrioCeilingNoOwner While the semaphore object is a priority ceiling semaphore, while the semaphore has no owner.

PrioCeilingOwner While the semaphore object is a priority ceiling semaphore, while the semaphore has an owner.

PrioInherit While the semaphore object is a priority inheritance semaphore.

MrsP While the semaphore object is a MrsP semaphore.

SemId

Valid While the semaphore_id parameter is associated with the semaphore.

Invalid While the semaphore_id parameter is not associated with a semaphore.

SchedId

Invalid While the scheduler_id parameter is not associated with a scheduler.

Create While the scheduler_id parameter is associated with the scheduler used to create the semaphore.

Other While the scheduler_id parameter is associated with a scheduler other than the one used to create the semaphore.

NewPrio

Current While the new_priority parameter is equal to RTEMS_CURRENT_PRIORITY.

Valid While the new_priority parameter is not equal to RTEMS_CURRENT_PRIORITY and valid with respect to the scheduler specified by the scheduler_id parameter.

Invalid While the new_priority parameter is invalid with respect to the scheduler specified by the scheduler_id parameter.

OldPrio

Valid While the old_priority parameter references an object of type rtems_task_priority.

Null While the old_priority parameter is NULL.

5.1.271.2 post-conditions

Status

Ok The return status of rtems_semaphore_set_priority shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_semaphore_set_priority shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_semaphore_set_priority shall be RTEMS_INVALID_ID.

InvPrio The return status of rtems_semaphore_set_priority shall be RTEMS_INVALID_PRIORITY.

NotDef The return status of rtems_semaphore_set_priority shall be RTEMS_NOT_DEFINED.

OwnerPrio

Nop The current priority of the owner task of the semaphore for the scheduler specified by the scheduler_id parameter shall not be modified by the rtems_semaphore_set_priority call.

New The current priority of the owner task of the semaphore for the scheduler specified by the scheduler_id parameter shall be less than or equal to the value of the new_priority parameter.

SemPrio

Set The priority used for the scheduler specified by the scheduler_id parameter of the semaphore associated with the identifier specified by the semaphore_id parameter shall be set to the priority specified by the new_priority parameter during the rtems_semaphore_set_priority call.

Nop Priorities used by semaphores shall not be modified by the rtems_semaphore_set_priority call.

OldPrioVar

Set The value of the object referenced by the old_priority parameter shall be set to the priority used for the scheduler specified by the scheduler_id parameter of the semaphore associated with the identifier specified by the semaphore_id parameter right before the priority is set by the rtems_semaphore_set_priority call.

Nop Objects referenced by the old_priority parameter in past calls to rtems_semaphore_set_priority shall not be accessed by the rtems_semaphore_set_priority call.

5.1.271.3 skip-reasons

NoOtherScheduler Where the system was built with SMP support disabled, exactly one scheduler is present in an application.

5.1.271.4 transition-map

Status = *Ok*, OwnerPrio = *Nop*, SemPrio = *Nop*, OldPrioVar = *Set*

- Class = *PrioCeilingOwner*, SemId = *Valid*, SchedId = *Create*, NewPrio = *Current*, OldPrio = *Valid*

Status = *Ok*, OwnerPrio = *New*, SemPrio = *Set*, OldPrioVar = *Set*

- Class = *PrioCeilingOwner*, SemId = *Valid*, SchedId = *Create*, NewPrio = *Valid*, OldPrio = *Valid*

Status = *InvPrio*, OwnerPrio = *Nop*, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = *PrioCeilingOwner*, SemId = *Valid*, SchedId = { *Create*, *Other* }, NewPrio = *Invalid*, OldPrio = *Valid*

Status = *NotDef*, OwnerPrio = *Nop*, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = *PrioCeilingOwner*, SemId = *Valid*, SchedId = *Other*, NewPrio = { *Current*, *Valid* }, OldPrio = *Valid*

Status = *Ok*, OwnerPrio = NA, SemPrio = *Nop*, OldPrioVar = *Set*

- Class = { *PrioCeilingNoOwner*, *MrsP* }, SemId = *Valid*, SchedId = *Create*, NewPrio = *Current*, OldPrio = *Valid*

- Class = *MrsP*, SemId = *Valid*, SchedId = *Other*, NewPrio = *Current*, OldPrio = *Valid*

Status = *Ok*, OwnerPrio = NA, SemPrio = *Set*, OldPrioVar = *Set*

- Class = { *PrioCeilingNoOwner*, *MrsP* }, SemId = *Valid*, SchedId = *Create*, NewPrio = *Valid*, OldPrio = *Valid*

- Class = *MrsP*, SemId = *Valid*, SchedId = *Other*, NewPrio = *Valid*, OldPrio = *Valid*

Status = *InvPrio*, OwnerPrio = NA, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = { *Counting*, *Simple*, *Binary*, *PrioCeilingNoOwner*, *PrioInherit*, *MrsP* }, SemId = *Valid*, SchedId = { *Create*, *Other* }, NewPrio = *Invalid*, OldPrio = *Valid*

Status = *InvId*, OwnerPrio = *Nop*, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = *PrioCeilingOwner*, SemId = { *Valid*, *Invalid* }, SchedId = *Invalid*, NewPrio = { *Current*, *Valid*, *Invalid* }, OldPrio = *Valid*

- Class = *PrioCeilingOwner*, SemId = *Invalid*, SchedId = { *Create*, *Other* }, NewPrio = { *Current*, *Valid*, *Invalid* }, OldPrio = *Valid*

Status = *NotDef*, OwnerPrio = NA, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = { *Counting*, *Simple*, *Binary*, *PrioInherit* }, SemId = *Valid*, SchedId = { *Create*, *Other* }, NewPrio = { *Current*, *Valid* }, OldPrio = *Valid*

- Class = *PrioCeilingNoOwner*, SemId = *Valid*, SchedId = *Other*, NewPrio = { *Current*, *Valid* }, OldPrio = *Valid*

Status = *InvAddr*, OwnerPrio = *Nop*, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = *PrioCeilingOwner*, SemId = { *Valid*, *Invalid* }, SchedId = { *Invalid*, *Create*, *Other* }, NewPrio = { *Current*, *Valid*, *Invalid* }, OldPrio = *Null*

Status = *InvId*, OwnerPrio = NA, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = { *Counting*, *Simple*, *Binary*, *PrioCeilingNoOwner*, *PrioInherit*, *MrsP* }, SemId = { *Valid*, *Invalid* }, SchedId = *Invalid*, NewPrio = { *Current*, *Valid*, *Invalid* }, OldPrio = *Valid*

- Class = { *Counting*, *Simple*, *Binary*, *PrioCeilingNoOwner*, *PrioInherit*, *MrsP* }, SemId = *Invalid*, SchedId = { *Create*, *Other* }, NewPrio = { *Current*, *Valid*, *Invalid* }, OldPrio = *Valid*

Status = *InvAddr*, OwnerPrio = NA, SemPrio = *Nop*, OldPrioVar = *Nop*

- Class = { *Counting*, *Simple*, *Binary*, *PrioCeilingNoOwner*, *PrioInherit*, *MrsP* }, SemId = { *Valid*, *Invalid* }, SchedId = { *Invalid*, *Create*, *Other* }, NewPrio = { *Current*, *Valid*, *Invalid* }, OldPrio = *Null*

5.1.272 spec:/rtems/sem/req/timeout

spec:/rtems/sem/req/timeout

When a semaphore obtain timeout happens.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- *spec:/score/tq/req/timeout*
- *spec:/score/tq/req/timeout-mrsp*
- *spec:/score/tq/req/timeout-priority-inherit*

Traced design component: RTEMSAPIClassicSem - rtems_semaphore_obtain

5.1.272.1 pre-conditions

Class

Counting While the semaphore object is a counting semaphore.

Simple While the semaphore object is a simple binary semaphore.

Binary While the semaphore object is a binary semaphore.

PrioCeiling While the semaphore object is a priority ceiling semaphore.

PrioInherit While the semaphore object is a priority inheritance semaphore.

MrsP While the semaphore object is a MrsP semaphore.

Discipline

FIFO While the semaphore uses the FIFO task wait queue discipline.

Priority While the semaphore uses the priority task wait queue discipline.

5.1.272.2 post-conditions

Action

Timeout The semaphore obtain timeout actions shall be done as specified by /score/tq/req/timeout.

TimeoutMrsP The semaphore obtain timeout actions shall be done as specified by /score/tq/req/timeout-mrsp.

TimeoutPriorityInherit The semaphore obtain timeout actions shall be done as specified by /score/tq/req/timeout-priority-inherit.

5.1.272.3 skip-reasons

NeedsPriorityDiscipline Binary semaphores with a locking protocol are required to use the priority task wait queue discipline.

NoMrsP Where the system is build with SMP support disabled, the MrsP locking protocol is not available.

5.1.272.4 transition-map

Action = *TimeoutPriorityInherit*

- Class = *PrioInherit*, Discipline = *Priority*

Action = *TimeoutMrsP*

- Class = *MrsP*, Discipline = *Priority*

Action = *Timeout*

- Class = { *Counting*, *Simple*, *Binary* }, Discipline = { *FIFO*, *Priority* }
- Class = *PrioCeiling*, Discipline = *Priority*

NeedsPriorityDiscipline

- Class = { *PrioCeiling*, *PrioInherit*, *MrsP* }, Discipline = *FIFO*

5.1.273 spec:/rtems/signal/req/catch

spec:/rtems/signal/req/catch

When the rtems_signal_catch() (see: spec:/rtems/signal/if/catch) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSignal - rtems_signal_catch

5.1.273.1 pre-conditions

Pending

Yes Where the system has more than one processor, while the calling task has pending signals.

Where the system has exactly one processor, while the calling task has no pending signals.

No While the calling task has no pending signals.

Handler

Invalid While the asr_handler parameter is NULL.

Valid While the asr_handler parameter is a valid ASR handler.

Preempt

Yes While the mode_set parameter specifies that preemption is enabled.

No While the mode_set parameter specifies that preemption is disabled.

Timeslice

Yes While the mode_set parameter specifies that timeslicing is enabled.

No While the mode_set parameter specifies that timeslicing is disabled.

ASR

Yes While the mode_set parameter specifies that ASR processing is enabled.

No While the mode_set parameter specifies that ASR processing is disabled.

IntLvl

Zero While the mode_set parameter specifies an interrupt level of zero.

Positive While the mode_set parameter specifies an interrupt level greater than or equal to one and less than or equal to CPU_MODES_INTERRUPT_MASK.

5.1.273.2 post-conditions

Status

Ok The return status of rtems_signal_catch shall be RTEMS_SUCCESSFUL.

NotImplIntLvl The return status of rtems_signal_catch shall be RTEMS_NOT_IMPLEMENTED.

NotImplIntLvlSMP Where the system needs inter-processor interrupts, the return status of rtems_signal_catch shall be RTEMS_NOT_IMPLEMENTED.

Where the system does not need inter-processor interrupts, the return status of rtems_signal_catch shall be RTEMS_SUCCESSFUL.

NotImplNoPreempt Where the scheduler does not support the no-preempt mode, the return status of rtems_signal_catch shall be RTEMS_NOT_IMPLEMENTED.

Where the scheduler does support the no-preempt mode, the return status of rtems_signal_catch shall be RTEMS_SUCCESSFUL.

ASRInfo

NopIntLvl The ASR information of the caller of rtems_signal_catch shall not be changed by the rtems_signal_catch call.

NopIntLvlSMP Where the system needs inter-processor interrupts, the ASR information of the caller of rtems_signal_catch shall not be changed by the rtems_signal_catch call.

Where the system does not need inter-processor interrupts, the ASR processing for the caller of rtems_signal_catch shall be done using the handler specified by asr_handler in the mode specified by mode_set.

NopNoPreempt Where the scheduler does not support the no-preempt mode, the ASR information of the caller of rtems_signal_catch shall not be changed by the rtems_signal_catch call.

Where the scheduler does support the no-preempt mode, the ASR processing for the caller of rtems_signal_catch shall be done using the handler specified by asr_handler in the mode specified by mode_set.

New The ASR processing for the caller of rtems_signal_catch shall be done using the handler specified by asr_handler in the mode specified by mode_set.

Inactive The ASR processing for the caller of rtems_signal_catch shall be deactivated.

The pending signals of the caller of rtems_signal_catch shall be cleared.

5.1.273.3 transition-map

Status = *Ok*, ASRInfo = *New*

- Pending = { *Yes*, *No* }, Handler = *Valid*, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*

Status = *NotImplNoPreempt*, ASRInfo = *NopNoPreempt*

- Pending = { *Yes*, *No* }, Handler = *Valid*, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*

Status = *NotImplIntLvlSMP*, ASRInfo = *NopIntLvlSMP*

- Pending = { *Yes*, *No* }, Handler = *Valid*, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*

Status = *Ok*, ASRInfo = *Inactive*

- Pending = { *Yes*, *No* }, Handler = *Invalid*, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* }

5.1.274 spec:/rtems/signal/req/send

spec:/rtems/signal/req/send

When the rtems_signal_send() (see: spec:/rtems/signal/if/send) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSignal - rtems_signal_send

5.1.274.1 pre-conditions

Task

NoObj While the id parameter is not associated with a task.

Self While the id parameter is associated with the calling task.

Other While the id parameter is associated with a task other than the calling task.

Set

Zero While the signal_set parameter is zero.

NonZero While the signal_set parameter is non-zero.

Handler

Invalid While the target task has no valid ASR handler installed.

Valid While the target task has a valid ASR handler installed.

ASR

Enabled While the target task has ASR processing enabled.

Disabled While the target task has ASR processing disabled.

Nested

Yes While the target task processes an asynchronous signal.

No While the target task does not process an asynchronous signal.

5.1.274.2 post-conditions

Status

Ok The return status of rtems_signal_send shall be RTEMS_SUCCESSFUL.

InvNum The return status of rtems_signal_send shall be RTEMS_INVALID_NUMBER.

InvId The return status of rtems_signal_send shall be RTEMS_INVALID_ID.

NotDef The return status of rtems_signal_send shall be RTEMS_NOT_DEFINED.

Handler

NoCall While the ASR processing is disabled, the ASR handler shall not be called.

DuringSend The ASR handler shall be called during the rtems_signal_send call.

AfterDispatch When the next thread dispatch of the target task of the rtems_signal_send call takes place, the ASR handler shall be called.

AfterEnable When the target task of the rtems_signal_send call re-enables ASR processing, the ASR handler shall be called.

Recursive

Yes The ASR handler shall be called recursively.

No The ASR handler shall not be called recursively.

5.1.274.3 transition-map

Status = *Ok*, Handler = *DuringSend*, Recursive = *Yes*

- Task = *Self*, Set = *NonZero*, Handler = *Valid*, ASR = *Enabled*, Nested = *Yes*

Status = *Ok*, Handler = *DuringSend*, Recursive = *No*

- Task = *Self*, Set = *NonZero*, Handler = *Valid*, ASR = *Enabled*, Nested = *No*

Status = *Ok*, Handler = *AfterDispatch*, Recursive = *Yes*

- Task = *Other*, Set = *NonZero*, Handler = *Valid*, ASR = *Enabled*, Nested = *Yes*

Status = *Ok*, Handler = *AfterDispatch*, Recursive = *No*

- Task = *Other*, Set = *NonZero*, Handler = *Valid*, ASR = *Enabled*, Nested = *No*

Status = *Ok*, Handler = *AfterEnable*, Recursive = *No*

- Task = { *Self*, *Other* }, Set = *NonZero*, Handler = *Valid*, ASR = *Disabled*, Nested = { *Yes*, *No* }

Status = *InvId*, Handler = *NoCall*, Recursive = *No*

- Task = *NoObj*, Set = *NonZero*, Handler = { *Invalid*, *Valid* }, ASR = { *Enabled*, *Disabled* }, Nested = { *Yes*, *No* }

Status = *NotDef*, Handler = *NoCall*, Recursive = *No*

- Task = { *Self*, *Other* }, Set = *NonZero*, Handler = *Invalid*, ASR = { *Enabled*, *Disabled* }, Nested = { *Yes*, *No* }

Status = *InvNum*, Handler = *NoCall*, Recursive = *No*

- Task = { *NoObj*, *Self*, *Other* }, Set = *Zero*, Handler = { *Invalid*, *Valid* }, ASR = { *Enabled*, *Disabled* }, Nested = { *Yes*, *No* }

5.1.275 spec:/rtems/signal/req/signal-constant

spec:/rtems/signal/req/signal-constant

Let $J \in \{0, 1, \dots, 31\}$. The J -th signal constant shall be a constant expression which evaluates to the integer representation of signal J .

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_0

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_1

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_2

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_3

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_4

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_5

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_6

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_7

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_8

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_9

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_10

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_11

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_12

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_13

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_14

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_15

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_16

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_17

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_18

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_19

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_20

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_21

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_22

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_23

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_24

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_25

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_26

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_27

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_28

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_29

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_30

Traced design component: RTEMSAPIClassicSignal - RTEMS_SIGNAL_31

5.1.276 spec:/rtems/signal/req/signal-count

spec:/rtems/signal/req/signal-count

The @ref RTEMSAPIClassicSignal shall support exactly 32 signals.

rationale: N/A

functional-type: function

This requirement refines [spec:/rtems/signal/req/group](#).

This requirement is refined by the following requirements:

- [spec:/rtems/signal/req/signal-number](#)

5.1.277 spec:/rtems/signal/req/signal-delivery

spec:/rtems/signal/req/signal-delivery

Let $J \in \{0, 1, \dots, 31\}$. While a task is able to catch signals, when signal J is sent to the task, it shall be caught by the task as exactly signal J .

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSignal - rtems_signal_send

5.1.278 spec:/rtems/signal/req/signal-number

spec:/rtems/signal/req/signal-number

Let $J \in \{0, 1, \dots, 31\}$. The signal J shall be represented by the integer 2^J .

rationale: N/A

functional-type: function

This requirement refines *spec:/rtems/signal/req/signal-count*.

5.1.279 spec:/rtems/signal/req/signal-set

spec:/rtems/signal/req/signal-set

Let $J \in \{0, 1, \dots, 31\}$. The rtems_signal_set integer type shall be able to represent the integer representation of signal J .

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSignal - rtems_signal_set

5.1.280 spec:/rtems/status/req/first

spec:/rtems/status/req/first

The RTEMS_STATUS_CODES_FIRST constant shall be equal to the minimum value of all rtems_status_code enumerators.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicStatus - RTEMS_STATUS_CODES_FIRST

5.1.281 spec:/rtems/status/req/is-equal

spec:/rtems/status/req/is-equal

When the rtems_are_statuses_equal() (see: spec:/rtems/status/if/is-equal) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicStatus - rtems_are_statuses_equal

5.1.281.1 pre-conditions

Status

Equal While the left_status_code parameter is equal to the right_status_code parameter.

NotEqual While the left_status_code parameter is not equal to the right_status_code parameter.

5.1.281.2 post-conditions

Result

True The return value of rtems_are_statuses_equal shall be true.

False The return value of rtems_are_statuses_equal shall be false.

5.1.281.3 transition-map

Result = *True*

- Status = *Equal*

Result = *False*

- Status = *NotEqual*

5.1.282 spec:/rtems/status/req/is-successful

spec:/rtems/status/req/is-successful

When the rtems_is_status_successful() (see: spec:/rtems/status/if/is-successful) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicStatus - rtems_is_status_successful

5.1.282.1 pre-conditions

Status

Successful While the status_code parameter is equal to RTEMS_SUCCESSFUL.

Other While the status_code parameter is not equal to RTEMS_SUCCESSFUL.

5.1.282.2 post-conditions

Result

True The return value of rtems_is_status_successful shall be true.

False The return value of rtems_is_status_successful shall be false.

5.1.282.3 transition-map

Result = *True*

- Status = *Successful*

Result = *False*

- Status = *Other*

5.1.283 spec:/rtems/status/req/last

spec:/rtems/status/req/last

The RTEMS_STATUS_CODES_LAST constant shall be equal to the maximum value of all rtems_status_code enumerators.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicStatus - RTEMS_STATUS_CODES_LAST

5.1.284 spec:/rtems/status/req/text

spec:/rtems/status/req/text

When the rtems_status_text() (see: spec:/rtems/status/if/text) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicStatus - rtems_status_text

5.1.284.1 pre-conditions

Code

AlreadySuspended While the status_code parameter value is equal to RTEMS_ALREADY_SUSPENDED.

CalledFromIsr While the status_code parameter value is equal to RTEMS_CALLED_FROM_ISR.

IllegalOnRemoteObject While the status_code parameter value is equal to RTEMS_ILLEGAL_ON_REMOTE_OBJECT.

IllegalOnSelf While the status_code parameter value is equal to RTEMS_ILLEGAL_ON_SELF.

IncorrectState While the status_code parameter value is equal to RTEMS_INCORRECT_STATE.

InternalError While the status_code parameter value is equal to RTEMS_INTERNAL_ERROR.

Interrupted While the status_code parameter value is equal to RTEMS_INTERRUPTED.

InvalidAddress While the status_code parameter value is equal to RTEMS_INVALID_ADDRESS.

InvalidClock While the status_code parameter value is equal to RTEMS_INVALID_CLOCK.

InvalidId While the status_code parameter value is equal to RTEMS_INVALID_ID.

InvalidName While the status_code parameter value is equal to RTEMS_INVALID_NAME.

InvalidNode While the status_code parameter value is equal to RTEMS_INVALID_NODE.

InvalidNumber While the status_code parameter value is equal to RTEMS_INVALID_NUMBER.

InvalidPriority While the status_code parameter value is equal to RTEMS_INVALID_PRIORITY.

InvalidSize While the status_code parameter value is equal to RTEMS_INVALID_SIZE.

IoError While the status_code parameter value is equal to RTEMS_IO_ERROR.

MpNotConfigured While the status_code parameter value is equal to RTEMS_MP_NOT_CONFIGURED.

NoMemory While the status_code parameter value is equal to RTEMS_NO_MEMORY.

NotConfigured While the status_code parameter value is equal to RTEMS_NOT_CONFIGURED.

NotDefined While the status_code parameter value is equal to RTEMS_NOT_DEFINED.

NotImplemented While the status_code parameter value is equal to RTEMS_NOT_IMPLEMENTED.

NotOwnerOfResource While the status_code parameter value is equal to RTEMS_NOT_OWNER_OF_RESOURCE.

ObjectWasDeleted While the status_code parameter value is equal to RTEMS_OBJECT_WAS_DELETED.

ProxyBlocking While the status_code parameter value is equal to RTEMS_PROXY_BLOCKING.

ResourceInUse While the status_code parameter value is equal to RTEMS_RESOURCE_IN_USE.

Successful While the status_code parameter value is equal to RTEMS_SUCCESSFUL.

TaskExitted While the status_code parameter value is equal to RTEMS_TASK_EXITTED.

Timeout While the status_code parameter value is equal to RTEMS_TIMEOUT.

TooMany While the status_code parameter value is equal to RTEMS_TOO_MANY.

Unsatisfied While the status_code parameter value is equal to RTEMS_UNSATISFIED.

NotAStatusCode While the status_code parameter is not equal to an enumerator of rtems_status_code.

5.1.284.2 post-conditions

Result

AlreadySuspended The return value of rtems_status_text shall reference a string which is equal to “RTEMS_ALREADY_SUSPENDED”.

CalledFromIsr The return value of rtems_status_text shall reference a string which is equal to “RTEMS_CALLED_FROM_ISR”.

IllegalOnRemoteObject The return value of rtems_status_text shall reference a string which is equal to “RTEMS_ILLEGAL_ON_REMOTE_OBJECT”.

IllegalOnSelf The return value of rtems_status_text shall reference a string which is equal to “RTEMS_ILLEGAL_ON_SELF”.

IncorrectState The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INCORRECT_STATE”.

InternalError The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INTERNAL_ERROR”.

Interrupted The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INTERRUPTED”.

InvalidAddress The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_ADDRESS”.

InvalidClock The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_CLOCK”.

InvalidId The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_ID”.

InvalidName The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_NAME”.

InvalidNode The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_NODE”.

InvalidNumber The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_NUMBER”.

InvalidPriority The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_PRIORITY”.

InvalidSize The return value of rtems_status_text shall reference a string which is equal to “RTEMS_INVALID_SIZE”.

IoError The return value of rtems_status_text shall reference a string which is equal to “RTEMS_IO_ERROR”.

MpNotConfigured The return value of rtems_status_text shall reference a string which is equal to “RTEMS_MP_NOT_CONFIGURED”.

NoMemory The return value of rtems_status_text shall reference a string which is equal to “RTEMS_NO_MEMORY”.

NotConfigured The return value of rtems_status_text shall reference a string which is equal to “RTEMS_NOT_CONFIGURED”.

NotDefined The return value of rtems_status_text shall reference a string which is equal to “RTEMS_NOT_DEFINED”.

NotImplemented The return value of rtems_status_text shall reference a string which is equal to “RTEMS_NOT_IMPLEMENTED”.

NotOwnerOfResource The return value of rtems_status_text shall reference a string which is equal to “RTEMS_NOT_OWNER_OF_RESOURCE”.

ObjectWasDeleted The return value of rtems_status_text shall reference a string which is equal to “RTEMS_OBJECT_WAS_DELETED”.

ProxyBlocking The return value of rtems_status_text shall reference a string which is equal to “RTEMS_PROXY_BLOCKING”.

ResourceInUse The return value of rtems_status_text shall reference a string which is equal to “RTEMS_RESOURCE_IN_USE”.

Successful The return value of rtems_status_text shall reference a string which is equal to “RTEMS_SUCCESSFUL”.

TaskExited The return value of rtems_status_text shall reference a string which is equal to “RTEMS_TASK_EXITTED”.

Timeout The return value of rtems_status_text shall reference a string which is equal to “RTEMS_TIMEOUT”.

TooMany The return value of rtems_status_text shall reference a string which is equal to “RTEMS_TOO_MANY”.

Unsatisfied The return value of rtems_status_text shall reference a string which is equal to “RTEMS_UNSATISFIED”.

NotAStatusCode The return value of rtems_status_text shall reference a string which is equal to “?”.

5.1.284.3 transition-map

Result = *AlreadySuspended*

- Code = *AlreadySuspended*

Result = *CalledFromIsr*

- Code = *CalledFromIsr*

Result = *IllegalOnRemoteObject*

- Code = *IllegalOnRemoteObject*

Result = *IllegalOnSelf*

- Code = *IllegalOnSelf*

Result = *IncorrectState*

- Code = *IncorrectState*

Result = *InternalError*

- Code = *InternalError*

Result = *Interrupted*

- Code = *Interrupted*

Result = *InvalidAddress*

- Code = *InvalidAddress*

Result = *InvalidClock*

- Code = *InvalidClock*

Result = *InvalidId*

- Code = *InvalidId*

Result = *InvalidName*

- Code = *InvalidName*

Result = *InvalidNode*

- Code = *InvalidNode*

Result = *InvalidNumber*

- Code = *InvalidNumber*

Result = *InvalidPriority*

- Code = *InvalidPriority*

Result = *InvalidSize*

- Code = *InvalidSize*

Result = *IoError*

- Code = *IoError*

Result = *MpNotConfigured*

- Code = *MpNotConfigured*

Result = *NoMemory*

- Code = *NoMemory*

Result = *NotConfigured*

- Code = *NotConfigured*

Result = *NotDefined*

- Code = *NotDefined*

Result = *NotImplemented*

- Code = *NotImplemented*

Result = *NotOwnerOfResource*

- Code = *NotOwnerOfResource*

Result = *ObjectWasDeleted*

- Code = *ObjectWasDeleted*

Result = *ProxyBlocking*

- Code = *ProxyBlocking*

Result = *ResourceInUse*

- Code = *ResourceInUse*

Result = *Successful*

- Code = *Successful*

Result = *TaskExitted*

- Code = *TaskExitted*

Result = *Timeout*

- Code = *Timeout*

Result = *TooMany*

- Code = *TooMany*

Result = *Unsatisfied*

- Code = *Unsatisfied*

Result = *NotAStatusCode*

- Code = *NotAStatusCode*

5.1.285 spec:/rtems/support/req/is-name-valid

spec:/rtems/support/req/is-name-valid

When the rtems_is_name_valid() (see: spec:/rtems/support/if/is-name-valid) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicSupport - rtems_is_name_valid

5.1.285.1 pre-conditions

Name

Valid While the name parameter is not equal to zero.

Invalid While the name parameter is equal to zero.

5.1.285.2 post-conditions

Result

True The return value of rtems_is_name_valid shall be true.

False The return value of rtems_is_name_valid shall be false.

5.1.285.3 transition-map

Result = *True*

- Name = *Valid*

Result = *False*

- Name = *Invalid*

5.1.286 spec:/rtems/support/req/name-to-characters-1

spec:/rtems/support/req/name-to-characters-1

When the call to rtems_name_to_characters returned, the object referenced by the c1 parameter shall be set the value of name divided by 16777216 modulo 256.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSupport - rtems_name_to_characters

5.1.287 spec:/rtems/support/req/name-to-characters-2

spec:/rtems/support/req/name-to-characters-2

When the call to rtems_name_to_characters returned, the object referenced by the c2 parameter shall be set the value of name divided by 65536 modulo 256.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSupport - rtems_name_to_characters

5.1.288 spec:/rtems/support/req/name-to-characters-3

spec:/rtems/support/req/name-to-characters-3

When the call to rtems_name_to_characters returned, the object referenced by the c3 parameter shall be set the value of name divided by 256 modulo 256.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSupport - rtems_name_to_characters

5.1.289 spec:/rtems/support/req/name-to-characters-4

spec:/rtems/support/req/name-to-characters-4

When the call to rtems_name_to_characters returned, the object referenced by the c4 parameter shall be set the value of name modulo 256.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicSupport - rtems_name_to_characters

5.1.290 spec:/rtems/task/req/configured-minimum-stack-size

spec:/rtems/task/req/configured-minimum-stack-size

The RTEMS_CONFIGURED_MINIMUM_STACK_SIZE constant shall expand to a constant expression which evaluates to the value of zero.

rationale: Using a value of zero will let rtems_task_create select the configured minimum task stack size.

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_CONFIGURED_MINIMUM_STACK_SIZE

5.1.291 spec:/rtems/task/req/construct

spec:/rtems/task/req/construct

When the rtems_task_construct() (see: [spec:/rtems/task/if/construct](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_construct

5.1.291.1 pre-conditions

CPUs

One Where the system does not need inter-processor interrupts, where the scheduler does support the no-preempt mode.

More Where the system needs inter-processor interrupts, where the scheduler does not support the no-preempt mode.

Config

Valid While the config parameter references an object of type rtems_task_config.

Null While the config parameter is NULL.

Name

Valid While the name of the task configuration is valid.

Invalid While the name of the task configuration is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

SystemTask

Yes While the attributes of the task configuration specifies a system task.

No While the attributes of the task configuration specifies an application task.

Priority

Valid While the initial priority of the task configuration is valid and non-zero.

Zero While the initial priority of the task configuration is zero.

Invalid While the initial priority of the task configuration is invalid.

Free

Yes While the system has at least one inactive task object available.

No While the system has no inactive task object available.

TLS

Enough While the maximum thread-local storage size of the task configuration is greater than or equal to the thread-local storage size.

TooSmall While the maximum thread-local storage size of the task configuration is less than the thread-local storage size.

Stack

Enough While the task stack size of the task configuration is greater than or equal to the configured minimum size.

TooSmall While the task stack size of the task configuration is less than the configured minimum size.

Create

Ok While none of the thread create user extensions fails.

Error While at least one of the thread create user extensions fails.

FloatingPoint

Yes While the attributes of the task configuration specifies a floating-point task.

No While the attributes of the task configuration specifies a non-floating-point task.

Preempt

Yes While the initial modes of the task configuration specify that preemption is enabled.

No While the initial modes of the task configuration specify that preemption is disabled.

Timeslice

Yes While the initial modes of the task configuration specify that timeslicing is enabled.

No While the initial modes of the task configuration specify that timeslicing is disabled.

ASR

Yes While the initial modes of the task configuration specify that ASR processing is enabled.

No While the initial modes of the task configuration specify that ASR processing is disabled.

IntLvl

Zero While the initial modes of the task configuration specify an interrupt level of zero.

Positive While the initial modes of the task configuration specify an interrupt level greater than zero and less than or equal to CPU_MODES_INTERRUPT_MASK.

5.1.291.2 post-conditions

Status

Ok The return status of rtems_task_construct shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_construct shall be RTEMS_INVALID_ADDRESS.

InvName The return status of rtems_task_construct shall be RTEMS_INVALID_NAME.

InvPrio The return status of rtems_task_construct shall be RTEMS_INVALID_PRIORITY.

InvSize The return status of rtems_task_construct shall be RTEMS_INVALID_SIZE.

TooMany The return status of rtems_task_construct shall be RTEMS_TOO_MANY.

Unsat The return status of rtems_task_construct shall be RTEMS_UNSATISFIED.

Name

Valid The unique object name shall identify the task constructed by the rtems_task_construct call.

Invalid The unique object name shall not identify a task.

IdObj

Set The value of the object referenced by the id parameter shall be set to the object identifier of the constructed task after the return of the rtems_task_construct call.

Nop Objects referenced by the id parameter in past calls to rtems_task_construct shall not be accessed by the rtems_task_construct call.

CreateNew

All The thread create user extensions shall be invoked for the task under construction during the rtems_task_construct call.

UpToFailing The thread create user extensions up to the failing extension shall be invoked for the task under construction during the rtems_task_construct call.

Nop The thread create user extensions shall not be invoked for the task under construction during the rtems_task_construct call.

DeleteNew

All The thread delete user extensions shall be invoked for the task under construction during the rtems_task_construct call.

Nop The thread delete user extensions shall not be invoked for the task under construction during the rtems_task_construct call.

KillZombies

Yes The registered zombie threads shall be killed before an attempt to allocate a TCB is made by the rtems_task_construct call.

No The registered zombie threads shall not be killed by the rtems_task_construct call.

StorageFree

Yes The storage free handler of the task configuration shall be invoked during the rtems_task_construct call.

No The storage free handler of the task configuration shall not be invoked during the rtems_task_construct call.

FloatingPoint

Yes Where threads have a dedicated floating-point context, the task constructed by the rtems_task_construct call shall be able to use the floating-point unit.

No Where threads have a dedicated floating-point context, the task constructed by the rtems_task_construct call shall not be able to use the floating-point unit.

Preempt

Yes Task preemption in the initial modes of the task constructed by the rtems_task_construct call shall be enabled.

No Task preemption in the initial modes of the task constructed by the rtems_task_construct call shall be disabled.

Timeslice

Yes Timeslicing in the initial modes of the task constructed by the rtems_task_construct call shall be enabled.

No Timeslicing in the initial modes of the task constructed by the rtems_task_construct call shall be disabled.

ASR

Yes ASR processing in the initial modes of the task constructed by the rtems_task_construct call shall be enabled.

No ASR processing in the initial modes of the task constructed by the rtems_task_construct call shall be disabled.

IntLvl

Zero The interrupt level in the initial modes of the task constructed by the `rtems_task_construct` call shall be zero.

Positive The interrupt level in the initial modes of the task constructed by the `rtems_task_construct` call shall be the interrupt level specified by the initial modes of the task configuration mapped to an target architecture-specific positive value.

5.1.291.3 skip-reasons

OnlyOneCPU Where the system is build with SMP support disabled, the system has exactly one processor.

5.1.291.4 transition-map

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *No*, Timeslice = *No*, ASR = *No*, IntLvl = *Positive*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *Yes*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *Yes*, ASR = *No*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *Yes*, IntLvl = *Zero*

Status = *Ok*, Name = *Valid*, IdObj = *Set*, CreateNew = *All*, DeleteNew = *Nop*, KillZombies = *Yes*, StorageFree = *No*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*
- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Ok*, FloatingPoint = *No*, Preempt = *Yes*, Timeslice = *No*, ASR = *No*, IntLvl = *Zero*

Status = *Unsat*, Name = *Invalid*, IdObj = *Nop*, CreateNew = *UpToFailing*, DeleteNew = *All*, KillZombies = *Yes*, StorageFree = *Yes*, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Error*, FloatingPoint = { *Yes*, *No* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* }
- CPUs = *One*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Error*, FloatingPoint = { *Yes*, *No* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* }
- CPUs = *More*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Error*, FloatingPoint = { *Yes*, *No* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*
- CPUs = *More*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = *Error*, FloatingPoint = { *Yes*, *No* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*

Status = *Unsat*, Name = *Invalid*, IdObj = *Nop*, CreateNew = *Nop*, DeleteNew = *All*, KillZombies = *Yes*, StorageFree = *Yes*, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = *More*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*
- CPUs = *More*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *Yes*, Priority = { *Valid*, *Zero* }, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* }
- CPUs = *More*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*
- CPUs = *More*, Config = *Valid*, Name = *Valid*, Id = *Valid*, SystemTask = *No*, Priority = *Valid*, Free = *Yes*, TLS = *Enough*, Stack = *Enough*, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*

= { Yes, No }, Preempt = No, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }

Status = InvSize, Name = Invalid, IdObj = Nop, CreateNew = Nop, DeleteNew = Nop, KillZombies = Yes, StorageFree = No, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = Yes, Priority = { Valid, Zero }, Free = Yes, TLS = Enough, Stack = TooSmall, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }
- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = Yes, Priority = { Valid, Zero }, Free = Yes, TLS = TooSmall, Stack = { Enough, TooSmall }, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }
- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = No, Priority = Valid, Free = Yes, TLS = Enough, Stack = TooSmall, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }
- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = No, Priority = Valid, Free = Yes, TLS = TooSmall, Stack = { Enough, TooSmall }, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }

Status = TooMany, Name = Invalid, IdObj = Nop, CreateNew = Nop, DeleteNew = Nop, KillZombies = Yes, StorageFree = No, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = Yes, Priority = { Valid, Zero }, Free = No, TLS = { Enough, TooSmall }, Stack = { Enough, TooSmall }, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }
- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = No, Priority = Valid, Free = No, TLS = { Enough, TooSmall }, Stack = { Enough, TooSmall }, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }

Status = InvPrio, Name = Invalid, IdObj = Nop, CreateNew = Nop, DeleteNew = Nop, KillZombies = No, StorageFree = No, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = Yes, Priority = Invalid, Free = { Yes, No }, TLS = { Enough, TooSmall }, Stack = { Enough, TooSmall }, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }
- CPUs = { One, More }, Config = Valid, Name = Valid, Id = Valid, SystemTask = No, Priority = { Zero, Invalid }, Free = { Yes, No }, TLS = { Enough, TooSmall }, Stack = { Enough, TooSmall }, Create = { Ok, Error }, FloatingPoint = { Yes, No }, Preempt = { Yes, No }, Timeslice = { Yes, No }, ASR = { Yes, No }, IntLvl = { Zero, Positive }

Status = *InvName*, Name = *Invalid*, IdObj = *Nop*, CreateNew = *Nop*, DeleteNew = *Nop*, Kill-Zombies = *No*, StorageFree = *No*, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Invalid*, Id = { *Valid*, *Null* }, SystemTask = { *Yes*, *No* }, Priority = { *Valid*, *Zero*, *Invalid* }, Free = { *Yes*, *No* }, TLS = { *Enough*, *TooSmall* }, Stack = { *Enough*, *TooSmall* }, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* } }

Status = *InvAddr*, Name = *Invalid*, IdObj = *Nop*, CreateNew = *Nop*, DeleteNew = *Nop*, Kill-Zombies = *No*, StorageFree = *No*, FloatingPoint = NA, Preempt = NA, Timeslice = NA, ASR = NA, IntLvl = NA

- CPUs = { *One*, *More* }, Config = *Valid*, Name = *Valid*, Id = *Null*, SystemTask = { *Yes*, *No* }, Priority = { *Valid*, *Zero*, *Invalid* }, Free = { *Yes*, *No* }, TLS = { *Enough*, *TooSmall* }, Stack = { *Enough*, *TooSmall* }, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* } }
- CPUs = { *One*, *More* }, Config = *Null*, Name = { *Valid*, *Invalid* }, Id = { *Valid*, *Null* }, SystemTask = { *Yes*, *No* }, Priority = { *Valid*, *Zero*, *Invalid* }, Free = { *Yes*, *No* }, TLS = { *Enough*, *TooSmall* }, Stack = { *Enough*, *TooSmall* }, Create = { *Ok*, *Error* }, FloatingPoint = { *Yes*, *No* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* } }

5.1.292 spec:/rtems/task/req/construct-home-scheduler

spec:/rtems/task/req/construct-home-scheduler

When a task is constructed, the home scheduler of the constructed task shall be set to the home scheduler of the task calling the rtems_task_construct directive at some time point during the directive call.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_construct

5.1.293 spec:/rtems/task/req/create-errors

spec:/rtems/task/req/create-errors

When the rtems_task_create() (see: spec:/rtems/task/if/create) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_create

5.1.293.1 pre-conditions

Name

Valid While the name parameter is valid.

Inv While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

SysTsk

Yes While the attribute_set parameter specifies a system task.

No While the attribute_set parameter specifies an application task.

Prio

Valid While the initial_priority parameter is valid and non-zero.

Zero While the initial_priority parameter is zero.

Inv While the initial_priority parameter is invalid.

Free

Yes While the system has at least one inactive task object available.

No While the system has no inactive task object available.

Stack

Normal While the initial_priority parameter is greater than or equal to the configured minimum size and less than or equal to the maximum stack size which can be allocated by the system.

Small While the initial_priority parameter is less than the configured minimum size.

Huge While the initial_priority parameter is greater than the maximum stack size which can be allocated by the system.

Ext

Ok While none of the task create extensions fails.

Err While at least one of the task create extensions fails.

5.1.293.2 post-conditions

Status

Ok The return status of rtems_task_create shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_create shall be RTEMS_INVALID_ADDRESS.

InvName The return status of rtems_task_create shall be RTEMS_INVALID_NAME.

InvPrio The return status of rtems_task_create shall be RTEMS_INVALID_PRIORITY.

TooMany The return status of rtems_task_create shall be RTEMS_TOO_MANY.

Unsat The return status of rtems_task_create shall be RTEMS_UNSATISFIED.

Name

Valid The unique object name shall identify the task created by the rtems_task_create call.

Invalid The unique object name shall not identify a task.

IdVar

Set The value of the object referenced by the id parameter shall be set to the object identifier of the created task after the return of the rtems_task_create call.

Nop Objects referenced by the id parameter in past calls to rtems_task_create shall not be accessed by the rtems_task_create call.

CreateExt

Yes The create user extensions shall be invoked during the rtems_task_create call.

No The create user extensions shall not be invoked during the rtems_task_create call.

DelExt

Yes The delete user extensions shall be invoked during the rtems_task_create call.

No The delete user extensions shall not be invoked during the rtems_task_create call.

5.1.293.3 transition-map

Status = *Ok*, Name = *Valid*, IdVar = *Set*, CreateExt = *Yes*, DelExt = *No*

- Name = *Valid*, Id = *Valid*, SysTsk = *Yes*, Prio = { *Valid*, *Zero* }, Free = *Yes*, Stack = { *Normal*, *Small* }, Ext = *Ok*
- Name = *Valid*, Id = *Valid*, SysTsk = *No*, Prio = *Valid*, Free = *Yes*, Stack = { *Normal*, *Small* }, Ext = *Ok*

Status = *Unsat*, Name = *Invalid*, IdVar = *Nop*, CreateExt = *Yes*, DelExt = *Yes*

- Name = *Valid*, Id = *Valid*, SysTsk = *Yes*, Prio = { *Valid*, *Zero* }, Free = *Yes*, Stack = { *Normal*, *Small* }, Ext = *Err*
- Name = *Valid*, Id = *Valid*, SysTsk = *No*, Prio = *Valid*, Free = *Yes*, Stack = { *Normal*, *Small* }, Ext = *Err*

Status = *Unsat*, Name = *Invalid*, IdVar = *Nop*, CreateExt = *No*, DelExt = *No*

- Name = *Valid*, Id = *Valid*, SysTsk = *Yes*, Prio = { *Valid*, *Zero* }, Free = *Yes*, Stack = *Huge*, Ext = { *Ok*, *Err* }
- Name = *Valid*, Id = *Valid*, SysTsk = *No*, Prio = *Valid*, Free = *Yes*, Stack = *Huge*, Ext = { *Ok*, *Err* }

Status = *TooMany*, Name = *Invalid*, IdVar = *Nop*, CreateExt = *No*, DelExt = *No*

- Name = *Valid*, Id = *Valid*, SysTsk = *Yes*, Prio = { *Valid*, *Zero* }, Free = *No*, Stack = { *Normal*, *Small*, *Huge* }, Ext = { *Ok*, *Err* }
- Name = *Valid*, Id = *Valid*, SysTsk = *No*, Prio = *Valid*, Free = *No*, Stack = { *Normal*, *Small*, *Huge* }, Ext = { *Ok*, *Err* }

Status = *InvPrio*, Name = *Invalid*, IdVar = *Nop*, CreateExt = *No*, DelExt = *No*

- Name = *Valid*, Id = *Valid*, SysTsk = *Yes*, Prio = *Inv*, Free = { *Yes*, *No* }, Stack = { *Normal*, *Small*, *Huge* }, Ext = { *Ok*, *Err* }
- Name = *Valid*, Id = *Valid*, SysTsk = *No*, Prio = { *Zero*, *Inv* }, Free = { *Yes*, *No* }, Stack = { *Normal*, *Small*, *Huge* }, Ext = { *Ok*, *Err* }

Status = *InvAddr*, Name = *Invalid*, IdVar = *Nop*, CreateExt = *No*, DelExt = *No*

- Name = *Valid*, Id = *Null*, SysTsk = { *Yes*, *No* }, Prio = { *Valid*, *Zero*, *Inv* }, Free = { *Yes*, *No* }, Stack = { *Normal*, *Small*, *Huge* }, Ext = { *Ok*, *Err* }

Status = *InvName*, Name = *Invalid*, IdVar = *Nop*, CreateExt = *No*, DelExt = *No*

- Name = *Inv*, Id = { *Valid*, *Null* }, SysTsk = { *Yes*, *No* }, Prio = { *Valid*, *Zero*, *Inv* }, Free = { *Yes*, *No* }, Stack = { *Normal*, *Small*, *Huge* }, Ext = { *Ok*, *Err* }

5.1.294 spec:/rtems/task/req/create-home-scheduler

spec:/rtems/task/req/create-home-scheduler

When a task is created, the home scheduler of the created task shall be set to the home scheduler of the task calling the rtems_task_create directive at some time point during the directive call.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_create

5.1.295 spec:/rtems/task/req/delete

spec:/rtems/task/req/delete

When the rtems_task_delete() (see: spec:/rtems/task/if/delete) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_delete

5.1.295.1 pre-conditions

Id

Executing While the id parameter is associated with the calling task.

Other While the id parameter is associated with a task other than the calling task.

Invalid While the id parameter is not associated with a task.

Context

Task While the rtems_task_delete directive is called from within task context.

Interrupt While the rtems_task_delete directive is called from within interrupt context.

ThreadDispatch

Disabled While thread dispatching is disabled for the calling task.

Enabled While thread dispatching is enabled for the calling task.

CallerPriority

Higher While the current priority of the task calling `rtems_task_delete` is higher than the real priority of the task specified by the `id` parameter.

LowerEqual While the current priority of the task calling `rtems_task_delete` is lower than or equal to the real priority of the task specified by the `id` parameter.

Dormant

No While the task specified by the `id` parameter is not dormant.

Yes While the task specified by the `id` parameter is dormant.

Suspended

Yes While the task specified by the `id` parameter is suspended.

No While the task specified by the `id` parameter is not suspended.

Restarting

No While the task specified by the `id` parameter is not restarting.

Yes While the task specified by the `id` parameter is restarting.

Terminating

No While the task specified by the `id` parameter is not terminating.

Yes While the task specified by the `id` parameter is terminating.

Protected

Yes While thread life of the task specified by the `id` parameter is protected.

No While thread life of the task specified by the `id` parameter is not protected.

State

Enqueued While the task specified by the `id` parameter is enqueued on a wait queue.

Ready While the task specified by the `id` parameter is a ready task or a scheduled task.

Blocked While the task specified by the `id` parameter is blocked.

Timer

Inactive While timer of the task specified by the `id` parameter is inactive.

Active While timer of the task specified by the `id` parameter is active.

5.1.295.2 post-conditions

Status

- Ok** The return status of rtems_task_delete shall be RTEMS_SUCCESSFUL.
- InvId** The return status of rtems_task_delete shall be RTEMS_INVALID_ID.
- CalledFromISR** The return status of rtems_task_delete shall be RTEMS_CALLED_FROM_ISR.
- NoReturn** The rtems_task_delete call shall not return.

FatalError

- Yes** The fatal error with a fatal source of INTERNAL_ERROR_CORE and a fatal code of INTERNAL_ERROR_BAD_THREAD_DISPATCH_DISABLE_LEVEL shall occur through the rtems_task_delete call.
- Nop** No fatal error shall occur through the rtems_task_delete call.

Zombie

- Yes** The task specified by the id parameter shall be in the zombie state after the rtems_task_delete call.
- No** The task specified by the id parameter shall not be in the zombie state after the rtems_task_delete call.

RealPriority

- Raised** The real priority of the task specified by the id parameter shall be raised to the current priority of the caller of rtems_task_delete.
- Nop** The real priority of the task specified by the id parameter shall not be changed by the rtems_task_delete call.

RestartExtensions

- Nop** The thread delete user extensions shall not be invoked by the rtems_task_delete call.

TerminateExtensions

- Yes** The thread terminate user extensions shall be invoked by the rtems_task_delete call.
- Nop** The thread terminate user extensions shall not be invoked by the rtems_task_delete call.

Dormant

- Yes** The task specified by the id parameter shall be dormant after the rtems_task_delete call.
- No** The task specified by the id parameter shall not be dormant after the rtems_task_delete call.

Suspended

Yes The task specified by the id parameter shall be suspended after the rtems_task_delete call.

No The task specified by the id parameter shall not be suspended after the rtems_task_delete call.

Restarting

Yes The task specified by the id parameter shall be restarting after the rtems_task_delete call.

No The task specified by the id parameter shall not be restarting after the rtems_task_delete call.

Terminating

Yes The task specified by the id parameter shall be terminating after the rtems_task_delete call.

No The task specified by the id parameter shall not be terminating after the rtems_task_delete call.

Protected

Yes The thread life of the task specified by the id parameter be protected after the rtems_task_delete call.

No The thread life of the task specified by the id parameter shall not be protected after the rtems_task_delete call.

State

Enqueued The task specified by the id parameter shall be enqueued on a wait queue and blocked.

Ready The task specified by the id parameter shall not be enqueued on a wait queue and not blocked.

Blocked The task specified by the id parameter shall be not enqueued on a wait queue and blocked.

Timer

Active The timer of the task specified by the id parameter shall be active after the rtems_task_delete call.

Inactive The timer of the task specified by the id parameter shall be inactive after the rtems_task_delete call.

5.1.295.3 skip-reasons

ExecutingIsNotDormant An executing thread was started and thus is never dormant.

ExecutingIsNotBlocked An executing thread is not blocked.

NotBlockedHasInactiveTimer The timer of a not blocked thread is inactive.

ThreadDispatchDisabled While ISRs or nested requests are processed, the thread dispatching is disabled.

5.1.295.4 transition-map

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *Yes*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Enqueued*, Timer = *Active*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Inactive*

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*
- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Enqueued*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Inactive*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Blocked*, Timer = *Active*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Nop*, Zombie = *Yes*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Yes*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Nop*, Zombie = *Yes*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Yes*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = *Ready*, Timer = *Inactive*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protecting = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protecting = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Raised*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protecting = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *Higher*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protecting = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protecting = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *Ok*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Executing*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = *No*, Suspended = *No*, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Ready*, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Enqueued*, Timer = { *Inactive*, *Active* }

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = { *Ready*, *Blocked* }, Timer = *Inactive*
- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = *LowerEqual*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *No*, *Yes* }, Protected = *No*, State = *Blocked*, Timer = *Active*

Status = *NoReturn*, FatalError = *Yes*, Zombie = *Yes*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *Yes*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = *Yes*, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Status = *NoReturn*, FatalError = *Yes*, Zombie = *Yes*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *Yes*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = *No*, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Status = *Ok*, FatalError = *Nop*, Zombie = *Yes*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *Yes*, Suspended = *Yes*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = *Yes*, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Status = *Ok*, FatalError = *Nop*, Zombie = *Yes*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *Yes*, Suspended = *No*, Restarting = *No*, Terminating = *Yes*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Task*, ThreadDispatch = *Enabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = *No*, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *Yes*, Suspended = *Yes*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = *Yes*, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Status = *CalledFromISR*, FatalError = *Nop*, Zombie = *No*, RealPriority = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = *Yes*, Suspended = *No*, Restarting = *No*, Terminating = *No*, Protected = *No*, State = *Ready*, Timer = *Inactive*

- Id = *Other*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = *No*, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Status = *InvId*, FatalError = *Nop*, Zombie = NA, RealPriority = NA, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Dormant = NA, Suspended = NA, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

- Id = *Invalid*, Context = *Task*, ThreadDispatch = { *Disabled*, *Enabled* }, CallerPriority = NA, Dormant = NA, Suspended = NA, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA
- Id = *Invalid*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = NA, Dormant = NA, Suspended = NA, Restarting = NA, Terminating = NA, Protected = NA, State = NA, Timer = NA

Executing IsNotDormant

- Id = *Executing*, Context = *Task*, ThreadDispatch = { *Disabled*, *Enabled* }, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = { *No*, *Yes* }, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = { *Enqueued*, *Ready*, *Blocked* }, Timer = { *Inactive*, *Active* }
- Id = *Executing*, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = { *No*, *Yes* }, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = { *Enqueued*, *Ready*, *Blocked* }, Timer = { *Inactive*, *Active* }

Executing IsNotBlocked

- Id = *Executing*, Context = *Task*, ThreadDispatch = { *Disabled*, *Enabled* }, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *No*, *Yes* }, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = { *Enqueued*, *Blocked* }, Timer = { *Inactive*, *Active* }

NotBlockedHasInactiveTimer

- Id = { *Executing*, *Other* }, Context = *Task*, ThreadDispatch = { *Disabled*, *Enabled* }, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *No*, *Yes* }, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = *Ready*, Timer = *Active*
- Id = { *Executing*, *Other* }, Context = *Interrupt*, ThreadDispatch = *Disabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *No*, *Yes* }, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = *Ready*, Timer = *Active*

ThreadDispatchDisabled

- Id = { *Executing*, *Other*, *Invalid* }, Context = *Interrupt*, ThreadDispatch = *Enabled*, CallerPriority = { *Higher*, *LowerEqual* }, Dormant = { *No*, *Yes* }, Suspended = { *Yes*, *No* }, Restarting = { *No*, *Yes* }, Terminating = { *No*, *Yes* }, Protected = { *Yes*, *No* }, State = { *Enqueued*, *Ready*, *Blocked* }, Timer = { *Inactive*, *Active* }

5.1.296 spec:/rtems/task/req/exit

spec:/rtems/task/req/exit

When the rtems_task_exit() (see: [spec:/rtems/task/if/exit](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_exit

5.1.296.1 pre-conditions

Restarting

Yes While the calling task is restarting.

No While the calling task is not restarting.

Terminating

Yes While the calling task is terminating.

No While the calling task is not terminating.

Protected

Yes While the thread life of the calling task is protected.

No While the thread life of the calling task is not protected.

ThreadDispatch

Enabled While thread dispatching is enabled for the calling task.

Disabled While thread dispatching is disabled for the calling task.

5.1.296.2 post-conditions

FatalError

Yes The fatal error with a fatal source of INTERNAL_ERROR_CORE and a fatal code of INTERNAL_ERROR_BAD_THREAD_DISPATCH_DISABLE_LEVEL shall occur by the rtems_task_exit call.

Nop No fatal error shall occur by the rtems_task_exit call.

DeleteExtensions

Nop The thread delete user extensions shall not be invoked by the rtems_task_exit call.

RestartExtensions

Nop The thread restart user extensions shall not be invoked by the rtems_task_exit call.

TerminateExtensions

Yes The thread terminate user extensions shall be invoked by the rtems_task_exit call.

Nop The thread terminate user extensions shall not be invoked by the rtems_task_exit call.

Block

Yes The calling task shall be blocked exactly once by the rtems_task_exit call.

Nop No task shall be blocked by the rtems_task_exit call.

ID

Valid The object identifier of the calling task shall be valid.

Invalid The object identifier of the calling task shall be invalid.

Delete

NextAllocate The calling task shall be deleted by the next directive which allocates a task.

Nop The calling task shall not be deleted by the next directive which allocates a task.

5.1.296.3 transition-map

FatalError = *Nop*, DeleteExtensions = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Yes*, Block = *Yes*, ID = *Invalid*, Delete = *NextAllocate*

- Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, ThreadDispatch = *Enabled*

FatalError = *Yes*, DeleteExtensions = *Nop*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*, Block = *Nop*, ID = *Valid*, Delete = *Nop*

- Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, ThreadDispatch = *Disabled*

5.1.297 spec:/rtems/task/req/get-affinity

spec:/rtems/task/req/get-affinity

When the rtems_task_get_affinity() (see: [spec:/rtems/task/if/get-affinity](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_get_affinity

5.1.297.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

CPUSetSize

Valid While the cpusetsize parameter is an integral multiple of the size of long, while the cpusetsize parameter specifies a processor set which is large enough to contain the processor affinity set of the task.

TooSmall While the cpusetsize parameter is an integral multiple of the size of long, while the cpusetsize parameter specifies a processor set which is not large enough to contain the processor affinity set of the task.

Askew While the cpusetsize parameter is not an integral multiple of the size of long.

CPUSet

Valid While the cpuset parameter references an object of type cpu_set_t.

Null While the cpuset parameter is equal to NULL.

5.1.297.2 post-conditions

Status

Ok The return status of rtems_task_get_affinity shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_get_affinity shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_get_affinity shall be RTEMS_INVALID_ID.

InvSize The return status of rtems_task_get_affinity shall be RTEMS_INVALID_SIZE.

CPUSetObj

Set The value of the object referenced by the cpuset parameter shall be set to the processor affinity set of the task specified by the id parameter at some point during the call after the return of the rtems_task_get_affinity call.

Nop Objects referenced by the cpuset parameter in past calls to rtems_task_get_affinity shall not be accessed by the rtems_task_get_affinity call.

5.1.297.3 transition-map

Status = *Ok*, CPUSetObj = *Set*

- Id = *Task*, CPUSetSize = *Valid*, CPUSet = *Valid*

Status = *InvSize*, CPUSetObj = *Nop*

- Id = *Task*, CPUSetSize = { *TooSmall*, *Askew* }, CPUSet = *Valid*

Status = *InvId*, CPUSetObj = *Nop*

- Id = *Invalid*, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Valid*

Status = *InvAddr*, CPUSetObj = *Nop*

- Id = { *Invalid*, *Task* }, CPUSetSize = { *Valid*, *TooSmall*, *Askew* }, CPUSet = *Null*

5.1.298 spec:/rtems/task/req/get-priority

spec:/rtems/task/req/get-priority

When the rtems_task_get_priority() (see: [spec:/rtems/task/if/get-priority](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_get_priority

5.1.298.1 pre-conditions

TaskId

Invalid While the task_id parameter is not associated with a task.

Task While the task_id parameter is associated with a task.

SchedulerId

Invalid While the scheduler_id parameter is not associated with a scheduler.

Scheduler While the scheduler_id parameter is associated with a scheduler.

Scheduler

Eligible While the scheduler_id parameter is associated with an eligible scheduler of the task specified by task_id.

Ineligible While the scheduler_id parameter is associated with an ineligible scheduler of the task specified by task_id.

Priority

Valid While the priority parameter references an object of type rtems_task_priority.

Null While the priority parameter is equal to NULL.

5.1.298.2 post-conditions

Status

Ok The return status of rtems_task_get_priority shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_get_priority shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_get_priority shall be RTEMS_INVALID_ID.

NotDef The return status of rtems_task_get_priority shall be RTEMS_NOT_DEFINED.

PriorityObj

Set The value of the object referenced by the scheduler_id parameter shall be set to the object identifier of the home scheduler of the task specified by the task_id parameter at some point during the call after the return of the rtems_task_get_priority call.

Nop Objects referenced by the scheduler_id parameter in past calls to rtems_task_get_priority shall not be accessed by the rtems_task_get_priority call.

5.1.298.3 skip-reasons

NoOtherScheduler Where the system was built with SMP support disabled, exactly one scheduler is present in an application.

5.1.298.4 transition-map

Status = *Ok*, PriorityObj = *Set*

- TaskId = *Task*, SchedulerId = *Scheduler*, Scheduler = *Eligible*, Priority = *Valid*

Status = *NotDef*, PriorityObj = *Nop*

- TaskId = *Task*, SchedulerId = *Scheduler*, Scheduler = *Ineligible*, Priority = *Valid*

Status = *InvId*, PriorityObj = *Nop*

- TaskId = *Invalid*, SchedulerId = { *Invalid*, *Scheduler* }, Scheduler = NA, Priority = *Valid*
- TaskId = *Task*, SchedulerId = *Invalid*, Scheduler = NA, Priority = *Valid*

Status = *InvAddr*, PriorityObj = *Nop*

- TaskId = *Invalid*, SchedulerId = { *Invalid*, *Scheduler* }, Scheduler = NA, Priority = *Null*
- TaskId = *Task*, SchedulerId = *Invalid*, Scheduler = NA, Priority = *Null*

- TaskId = *Task*, SchedulerId = *Scheduler*, Scheduler = { *Eligible*, *Ineligible* }, Priority = *Null*

5.1.299 spec:/rtems/task/req/get-scheduler

spec:/rtems/task/req/get-scheduler

When the rtems_task_get_scheduler() (see: spec:/rtems/task/if/get-scheduler) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_get_scheduler

5.1.299.1 pre-conditions

Id

Invalid While the task_id parameter is not associated with a task.

Task While the task_id parameter is associated with a task.

SchedulerID

Valid While the scheduler_id parameter references an object of type rtems_id.

Null While the scheduler_id parameter is equal to NULL.

5.1.299.2 post-conditions

Status

Ok The return status of rtems_task_get_scheduler shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_get_scheduler shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_get_scheduler shall be RTEMS_INVALID_ID.

SchedulerIDObj

Set The value of the object referenced by the scheduler_id parameter shall be set to the object identifier of the home scheduler of the task specified by the task_id parameter at some point during the call after the return of the rtems_task_get_scheduler call.

Nop Objects referenced by the scheduler_id parameter in past calls to rtems_task_get_scheduler shall not be accessed by the rtems_task_get_scheduler call.

5.1.299.3 transition-map

Status = *InvId*, SchedulerIDObj = *Nop*

- Id = *Invalid*, SchedulerID = *Valid*

Status = *Ok*, SchedulerIDObj = *Set*

- Id = *Task*, SchedulerID = *Valid*

Status = *InvAddr*, SchedulerIDObj = *Nop*

- Id = { *Invalid*, *Task* }, SchedulerID = *Null*

5.1.300 spec:/rtems/task/req/ident

spec:/rtems/task/req/ident

When the rtems_task_ident() (see: spec:/rtems/task/if/ident) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_ident

Traced design component: RTEMSAPIClassicObject - RTEMS_WHO_AM_I

5.1.300.1 pre-conditions

Name

WhoAmI While the name parameter is equal to RTEMS_WHO_AM_I, while id parameter is not equal to NULL.

NotWhoAmI While the name is not equal to RTEMS_WHO_AM_I or id parameter is equal to NULL, the behaviour of rtems_task_ident shall be specified by /rtems/req/ident.

5.1.300.2 post-conditions

Status

OkAndWhoAmI The return status of rtems_task_ident shall be RTEMS_SUCCESSFUL. The value of the object identifier referenced by the name parameter shall be the identifier of the executing thread.

Skip There is no status to validate.

5.1.300.3 transition-map

Status = *OkAndWhoAmI*

- Name = *WhoAmI*

Status = *Skip*

- Name = *NotWhoAmI*

5.1.301 spec:/rtems/task/req/is-suspended

spec:/rtems/task/req/is-suspended

When the rtems_task_is_suspended() (see: spec:/rtems/task/if/is-suspended) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_is_suspended

5.1.301.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

Suspended

Yes While the task specified by the id parameter is suspended.

No While the task specified by the id parameter is not suspended.

5.1.301.2 post-conditions

Status

Ok The return status of rtems_task_is_suspended shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_task_is_suspended shall be RTEMS_INVALID_ID.

AlrdySus The return status of rtems_task_is_suspended shall be RTEMS_ALREADY_SUSPENDED.

5.1.301.3 transition-map

Status = *AlrdySus*

- Id = *Task*, Suspended = *Yes*

Status = *Ok*

- Id = *Task*, Suspended = *No*

Status = *InvId*

- Id = *Invalid*, Suspended = NA

5.1.302 spec:/rtems/task/req/iterate-done

spec:/rtems/task/req/iterate-done

While an iteration over all tasks was started by calling rtems_task_iterate, while no task exists which was not visited, the iteration shall stop and then the object allocator lock shall be released.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_iterate

5.1.303 spec:/rtems/task/req/iterate-start

spec:/rtems/task/req/iterate-start

When rtems_task_iterate is called, the calling task shall obtain the object allocator lock and then start an iteration over all tasks.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_iterate

5.1.304 spec:/rtems/task/req/iterate-stop

spec:/rtems/task/req/iterate-stop

While an iteration over all tasks was started by calling rtems_task_iterate, when the function specified by the visitor returns true during the iteration, the iteration shall stop and the object allocator lock shall be released.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_iterate

5.1.305 spec:/rtems/task/req/iterate-visit

spec:/rtems/task/req/iterate-visit

While an iteration over all tasks was started by calling rtems_task_iterate, while a task exists which was not visited, the task shall be visited by calling the function specified by the visitor parameter of the rtems_task_iterate with the TCB of the task as the first parameter and the second parameter specified by the visitor parameter of the rtems_task_iterate call.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_iterate

5.1.306 spec:/rtems/task/req/maximum-priority

spec:/rtems/task/req/maximum-priority

The RTEMS_MAXIMUM_PRIORITY runtime constant shall be equal to the lowest priority of the scheduler with index zero.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_MAXIMUM_PRIORITY

5.1.307 spec:/rtems/task/req/minimum-priority

spec:/rtems/task/req/minimum-priority

The RTEMS_MINIMUM_PRIORITY constant shall expand to a constant expression which evaluates to the highest task priority available to application tasks.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_MINIMUM_PRIORITY

5.1.308 spec:/rtems/task/req/minimum-stack-size

spec:/rtems/task/req/minimum-stack-size

The RTEMS_MINIMUM_STACK_SIZE constant shall expand to a constant expression which evaluates to the value of STACK_MINIMUM_SIZE.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_MINIMUM_STACK_SIZE

5.1.309 spec:/rtems/task/req/mode

spec:/rtems/task/req/mode

When the rtems_task_mode() (see: spec:/rtems/task/if/mode) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_mode

Traced design component: RTEMSAPIClassicModes - RTEMS_CURRENT_MODE

5.1.309.1 pre-conditions

PrevMode

Valid While the previous_mode_set parameter references an object of type rtems_mode.

Null While the previous_mode_set parameter is NULL.

PreemptCur

Yes While the calling task has preemption enabled.

No Where the scheduler does not support the no-preempt mode, while the calling task has preemption enabled.

Where the scheduler does support the no-preempt mode, while the calling task has preemption disabled.

TimesliceCur

Yes While the calling task has timeslicing enabled.

No While the calling task has timeslicing disabled.

ASRCur

Yes While the calling task has ASR processing enabled.

No While the calling task has ASR processing disabled.

IntLvlCur

Zero While the calling task executes with an interrupt level of zero.

Positive Where the system needs inter-processor interrupts, while the calling task executes with an interrupt level of zero.

Where the system does not need inter-processor interrupts, while the calling task executes with an interrupt level greater than zero and less than or equal to CPU_MODES_INTERRUPT_MASK.

Preempt

Yes While the mode_set parameter specifies that preemption is enabled.

No While the mode_set parameter specifies that preemption is disabled.

Timeslice

Yes While the mode_set parameter specifies that timeslicing is enabled.

No While the mode_set parameter specifies that timeslicing is disabled.

ASR

Yes While the mode_set parameter specifies that ASR processing is enabled.

No While the mode_set parameter specifies that ASR processing is disabled.

IntLvl

Zero While the mode_set parameter specifies an interrupt level of zero.

Positive While the mode_set parameter specifies an interrupt level greater than zero and less than or equal to CPU_MODES_INTERRUPT_MASK.

PreemptMsk

Yes While the mask parameter specifies that the preemption mode shall be set.

No While the mask parameter specifies that the preemption mode shall not be set.

TimesliceMsk

Yes While the mask parameter specifies that the timeslicing mode shall be set.

No While the mask parameter specifies that the timeslicing mode shall not be set.

ASRMsk

Yes While the mask parameter specifies that the ASR processing mode shall be set.

No While the mask parameter specifies that the ASR processing mode shall not be set.

IntLvlMsk

Yes While the mask parameter specifies that the interrupt level shall be set.

No While the mask parameter specifies that the interrupt level shall not be set.

5.1.309.2 post-conditions

Status

Ok The return status of rtems_task_mode shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_mode shall be RTEMS_INVALID_ADDRESS.

NotImplIntLvl The return status of rtems_task_mode shall be RTEMS_NOT_IMPLEMENTED.

NotImplIntLvlSMP Where the system needs inter-processor interrupts, the return status of rtems_task_mode shall be RTEMS_NOT_IMPLEMENTED.

Where the system does not need inter-processor interrupts, the return status of rtems_task_mode shall be RTEMS_SUCCESSFUL.

NotImplNoPreempt Where the scheduler does not support the no-preempt mode, the return status of rtems_task_mode shall be RTEMS_NOT_IMPLEMENTED.

Where the scheduler does support the no-preempt mode, the return status of rtems_task_mode shall be RTEMS_SUCCESSFUL.

Preempt

Yes The calling task shall be preempted by a higher priority ready task during the rtems_task_mode call.

No The calling task shall not be preempted during the rtems_task_mode call.

Maybe Where the scheduler does not support the no-preempt mode, the calling task shall not be preempted during the rtems_task_mode call.

Where the scheduler does support the no-preempt mode, the calling task shall be preempted by a higher priority ready task during the rtems_task_mode call.

ASR

Yes The calling task shall process pending signals during the rtems_task_mode call.

No The calling task shall not process signals during the rtems_task_mode call.

Maybe Where the scheduler does not support the no-preempt mode, the calling task shall not process signals during the rtems_task_mode call.

Where the scheduler does support the no-preempt mode, the calling task shall process pending signals during the rtems_task_mode call.

PMVar

Set The value of the object referenced by the previous_mode_set parameter shall be set to the task modes of the calling task on entry of the call to rtems_task_mode.

Nop Objects referenced by the stack_size parameter in past calls to rtems_task_mode shall not be accessed by the rtems_task_mode call.

Maybe Where the scheduler does not support the no-preempt mode, objects referenced by the stack_size parameter in past calls to rtems_task_mode shall not be accessed by the rtems_task_mode call.

Where the scheduler does support the no-preempt mode, the value of the object referenced by the previous_mode_set parameter shall be set to the task modes of the calling task on entry of the call to rtems_task_mode.

Mode

Set The task modes of the calling task indicated by the mask parameter shall be set to the corresponding modes specified by the mode_set parameter.

Nop The task modes of the calling task shall not be modified by the rtems_task_mode call.

Maybe Where the scheduler does not support the no-preempt mode, the task modes of the calling task shall not be modified by the rtems_task_mode call.

Where the scheduler does support the no-preempt mode, the task modes of the calling task indicated by the mask parameter shall be set to the corresponding modes specified by the mode_set parameter.

5.1.309.3 skip-reasons

RobustThreadDispatching Where the system enabled robust thread dispatching, the interrupt level mode of a task shall be exactly zero.

5.1.309.4 transition-map

Status = *NotImplIntLvlSMP*, Preempt = *Maybe*, ASR = *Maybe*, PMVar = *Maybe*, Mode = *Maybe*

- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *Yes*

Status = *Ok*, Preempt = *Maybe*, ASR = *Yes*, PMVar = *Set*, Mode = *Set*

- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *No*

Status = *NotImplIntLvlSMP*, Preempt = *No*, ASR = *Maybe*, PMVar = *Maybe*, Mode = *Maybe*

- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *Yes*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *Yes*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *Yes*

Status = *NotImplIntLvlSMP*, Preempt = *Maybe*, ASR = *No*, PMVar = *Maybe*, Mode = *Maybe*

- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *Yes*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*

Status = *NotImplNoPreempt*, Preempt = *No*, ASR = *Maybe*, PMVar = *Maybe*, Mode = *Maybe*

- PrevMode = *Valid*, PreemptCur = { *Yes*, *No* }, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = { *Zero*, *Positive* }, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = { *Yes*, *No* }

Status = *Ok*, Preempt = *No*, ASR = *Yes*, PMVar = *Set*, Mode = *Set*

- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *Yes*, IntLvlMsk = *No*

Status = *Ok*, Preempt = *Maybe*, ASR = *No*, PMVar = *Set*, Mode = *Set*

- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Zero*,

PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }

- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*

Status = *NotImpl*, IntLvlSMP, Preempt = *No*, ASR = *No*, PMVar = *Maybe*, Mode = *Maybe*

- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*
 - PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *Yes*

Status = *NotImplNoPreempt*, Preempt = *No*, ASR = *No*, PMVar = *Maybe*, Mode = *Maybe*

- `PrevMode` = *Valid*, `PreemptCur` = { *Yes*, *No* }, `TimesliceCur` = { *Yes*, *No* }, `ASRCur` = *Yes*, `IntLvlCur` = { *Zero*, *Positive* }, `Preempt` = *No*, `Timeslice` = { *Yes*, *No* }, `ASR` = { *Yes*, *No* }, `IntLvl` = { *Zero*, *Positive* }, `PreemptMsk` = *Yes*, `TimesliceMsk` = { *Yes*, *No* }, `ASRMsk` = { *Yes*, *No* }, `IntLvlMsk` = { *Yes*, *No* }

- PrevMode = *Valid*, PreemptCur = { *Yes*, *No* }, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = { *Zero*, *Positive* }, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = { *Yes*, *No* }, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = { *Zero*, *Positive* }, PreemptMsk = *Yes*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }

Status = *Ok*, Preempt = *No*, ASR = *No*, PMVar = *Set*, Mode = *Set*

- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Zero*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *Yes*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = { *Yes*, *No* }

- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *Yes*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = *No*, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *Yes*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *Yes*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = *No*, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Zero*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }
- PrevMode = *Valid*, PreemptCur = *No*, TimesliceCur = { *Yes*, *No* }, ASRCur = *No*, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = *No*, IntLvl = *Positive*, PreemptMsk = *No*, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = *No*

Status = *InvAddr*, Preempt = *No*, ASR = *No*, PMVar = *Nop*, Mode = *Nop*

- PrevMode = *Null*, PreemptCur = { *Yes*, *No* }, TimesliceCur = { *Yes*, *No* }, ASRCur = { *Yes*, *No* }, IntLvlCur = { *Zero*, *Positive* }, Preempt = { *Yes*, *No* }, Timeslice = { *Yes*, *No* }, ASR = { *Yes*, *No* }, IntLvl = { *Zero*, *Positive* }, PreemptMsk = { *Yes*, *No* }, TimesliceMsk = { *Yes*, *No* }, ASRMsk = { *Yes*, *No* }, IntLvlMsk = { *Yes*, *No* }

5.1.310 spec:/rtems/task/req/no-priority

spec:/rtems/task/req/no-priority

The /rtems/task/if/no-priority constant shall expand to a constant expression which evaluates to the value of RTEMS_CURRENT_PRIORITY.

rationale: Some directives with a task priority parameter ignore the priority value under specific circumstances, for example rtems_task_set_priority may be used to get the current priority. This constant may be used as a dummy task priority value to document that the priority parameter is unused.

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_NO_PRIORITY

5.1.311 spec:/rtems/task/req/restart

spec:/rtems/task/req/restart

When the rtems_task_restart() (see: spec:/rtems/task/if/restart) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_restart

5.1.311.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Executing While the id parameter is associated with the calling task.

Other While the id parameter is associated with a task other than the calling task.

Dormant

Yes While the task specified by the id parameter is dormant.

No While the task specified by the id parameter is not dormant.

Suspended

Yes While the task specified by the id parameter is suspended.

No While the task specified by the id parameter is not suspended.

Restarting

Yes While the task specified by the id parameter is restarting.

No While the task specified by the id parameter is not restarting.

Terminating

Yes While the task specified by the id parameter is terminating.

No While the task specified by the id parameter is not terminating.

Protected

Yes While thread life of the task specified by the id parameter is protected.

No While thread life of the task specified by the id parameter is not protected.

Context

Task While the rtems_task_restart directive is called from within task context.

Interrupt While the rtems_task_restart directive is called from within interrupt context.

NestedRequest While the rtems_task_restart directive is called during another rtems_task_restart call with the same task as a nested request.

State

Ready While the task specified by the id parameter is a ready task or scheduled task.

Blocked While the task specified by the id parameter is blocked.

Enqueued While the task specified by the id parameter is enqueued on a wait queue.

Timer

Inactive While timer of the task specified by the id parameter is inactive.

Active While timer of the task specified by the id parameter is active.

RealPriority

Initial While real priority of the task specified by the id parameter is equal to the initial priority.

Changed While real priority of the task specified by the id parameter is not equal to the initial priority.

ThreadDispatch

Disabled While thread dispatching is disabled for the calling task.

Enabled While thread dispatching is enabled for the calling task.

5.1.311.2 post-conditions

Status

Ok The return status of rtems_task_restart shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_task_restart shall be RTEMS_INVALID_ID.

IncStat The return status of rtems_task_restart shall be RTEMS_INCORRECT_STATE.

NoReturn The rtems_task_restart call shall not return.

FatalError

Yes The fatal error with a fatal source of INTERNAL_ERROR_CORE and a fatal code of INTERNAL_ERROR_BAD_THREAD_DISPATCH_DISABLE_LEVEL shall occur through the rtems_task_restart call.

Nop No fatal error shall occur through the rtems_task_restart call.

Argument

Set The entry point argument of the task specified by the id parameter shall be set to the value specified by the argument parameter before the task is unblocked by the rtems_task_restart call.

Nop No entry point argument of a task shall be modified by the rtems_task_restart call.

State

Dormant The state of the task specified by the id parameter shall be dormant after the rtems_task_restart call.

DormantSuspended The state of the task specified by the id parameter shall be dormant and suspended after the rtems_task_restart call.

Blocked The state of the task specified by the id parameter shall be blocked after the rtems_task_restart call.

Ready The state of the task specified by the id parameter shall be ready after the rtems_task_restart call.

Zombie The state of the task specified by the id parameter shall be the zombie state after the rtems_task_restart call.

Nop The state of the task specified by the id parameter shall not be modified by the rtems_task_restart call.

Enqueued

Yes The task specified by the id parameter shall be enqueued on a wait queue after the rtems_task_restart call.

No The task specified by the id parameter shall not be enqueued on a wait queue after the rtems_task_restart call.

Timer

Active The timer of the task specified by the id parameter shall be active after the rtems_task_restart call.

Inactive The timer of the task specified by the id parameter shall be inactive after the rtems_task_restart call.

Restarting

Yes The task specified by the id parameter shall be restarting after the rtems_task_restart call.

No The task specified by the id parameter shall not be restarting after the rtems_task_restart call.

Terminating

Yes The task specified by the id parameter shall be terminating after the rtems_task_restart call.

No The task specified by the id parameter shall not be terminating after the rtems_task_restart call.

Protected

Yes The thread life of the task specified by the id parameter be protected after the rtems_task_restart call.

No The thread life of the task specified by the id parameter shall not be protected after the rtems_task_restart call.

RestartExtensions

Yes The thread restart user extensions shall be invoked by the rtems_task_restart call.

Nop The thread restart user extensions shall not be invoked by the rtems_task_restart call.

TerminateExtensions

Yes The thread terminate user extensions shall be invoked by the rtems_task_restart call.

Nop The thread terminate user extensions shall not be invoked by the rtems_task_restart call.

5.1.311.3 skip-reasons

Executing IsNotDormant An executing thread was started and thus is never dormant.

Executing IsNotBlocked An executing thread is not blocked.

NotBlocked Has Inactive Timer The timer of a not blocked thread is inactive.

Thread Dispatch Disabled While ISRs or nested requests are processed, the thread dispatching is disabled.

NestedRequestNotDormant A nested request can only happen if the thread is not dormant.

NestedRequestNotProtected A nested request can only happen if the thread life protection of the task is disabled.

NestedRequestNeedsLifeChanging A nested request can only happen if the thread life is changing.

NestedRequestNoNestedExit A nested request cannot happen during a thread exit.

NestedRequestNeedsTask A nested request needs a task to restart.

5.1.311.4 transition-map

Status = *NoReturn*, FatalError = *Yes*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *NoReturn*, FatalError = *Yes*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *NoReturn*, FatalError = *Yes*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *NoReturn*, FatalError = *Yes*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *NoReturn*, FatalError = *Nop*, Argument = *Set*, State = *Zombie*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Yes*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = { *Yes*, *No* }, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Enabled*

Status = *NoReturn*, FatalError = *Nop*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *No*, Terminating = *No*, Protected = *No*, RestartExtensions = *Yes*, TerminateExtensions = *Nop*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = { *Yes*, *No* }, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Enabled*

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Interrupt*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Interrupt*, State = *Blocked*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Task*, State = *Blocked*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *No*, Timer = *Active*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Interrupt*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Task*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *Yes*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Interrupt*, State = *Enqueued*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Task*, State = *Enqueued*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *Yes*, Timer = *Active*, Restarting = *Yes*, Terminating = *Yes*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Interrupt*, State = *Enqueued*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *Yes*, Context = *Task*, State = *Enqueued*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Interrupt*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Task*, State = *Ready*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Interrupt*, State = *Blocked*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Task*, State = *Blocked*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *No*, Timer = *Active*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Interrupt*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Task*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *Yes*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Interrupt*, State = *Enqueued*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Task*, State = *Enqueued*, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Blocked*, Enqueued = *Yes*, Timer = *Active*, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Interrupt*, State = *Enqueued*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Task*, State = *Enqueued*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Nop*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked* }, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *No*, Context = *NestedRequest*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *No*, Context = *NestedRequest*, State = *Enqueued*, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Nop*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked* }, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *NestedRequest*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *NestedRequest*, State = *Enqueued*, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *Yes*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Interrupt*, State = { *Ready*, *Blocked* }, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Interrupt*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Interrupt*, State = *Enqueued*, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Task*, State = { *Ready*, *Blocked* }, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Task*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = *No*, Context = *Task*, State = *Enqueued*, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *Ok*, FatalError = *Nop*, Argument = *Set*, State = *Ready*, Enqueued = *No*, Timer = *Inactive*, Restarting = *Yes*, Terminating = *No*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Interrupt*, State = { *Ready*, *Blocked* }, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Interrupt*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Interrupt*, State = *Enqueued*, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Task*, State = { *Ready*, *Blocked* }, Timer = *Inactive*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Task*, State = *Blocked*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *No*, Context = *Task*, State = *Enqueued*, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

Status = *IncStat*, FatalError = *Nop*, Argument = *Nop*, State = *DormantSuspended*, Enqueued = *No*, Timer = *Inactive*, Restarting = *No*, Terminating = *No*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Other*, Dormant = *Yes*, Suspended = *Yes*, Restarting = NA, Terminating = { *Yes*, *No* }, Protected = NA, Context = *Task*, State = NA, Timer = NA, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

- Id = *Other*, Dormant = *Yes*, Suspended = *Yes*, Restarting = NA, Terminating = { *Yes*, *No* }, Protected = NA, Context = *Interrupt*, State = NA, Timer = NA, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *IncStat*, FatalError = *Nop*, Argument = *Nop*, State = *Dormant*, Enqueued = *No*, Timer = *Inactive*, Restarting = *No*, Terminating = *No*, Protected = *No*, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Other*, Dormant = *Yes*, Suspended = *No*, Restarting = NA, Terminating = { *Yes*, *No* }, Protected = NA, Context = *Task*, State = NA, Timer = NA, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = *Yes*, Suspended = *No*, Restarting = NA, Terminating = { *Yes*, *No* }, Protected = NA, Context = *Interrupt*, State = NA, Timer = NA, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Status = *InvId*, FatalError = *Nop*, Argument = NA, State = NA, Enqueued = NA, Timer = NA, Restarting = NA, Terminating = NA, Protected = NA, RestartExtensions = *Nop*, TerminateExtensions = *Nop*

- Id = *Invalid*, Dormant = NA, Suspended = NA, Restarting = NA, Terminating = NA, Protected = NA, Context = *Task*, State = NA, Timer = NA, RealPriority = NA, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Invalid*, Dormant = NA, Suspended = NA, Restarting = NA, Terminating = NA, Protected = NA, Context = { *Task*, *Interrupt* }, State = NA, Timer = NA, RealPriority = NA, ThreadDispatch = *Disabled*

Executing IsNotDormant

- Id = *Executing*, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, Context = *Task*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Executing*, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, Context = *Interrupt*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

Executing IsNotBlocked

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, Context = *Task*, State = { *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

NotBlocked HasInActiveTimer

- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, Context = *Task*, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = { *Yes*, *No* }, Context = *Interrupt*, State = *Ready*, Timer = *Active*,

RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *No*, Protected = *Yes*, Context = *Interrupt*, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *No*, Context = { *Interrupt*, *NestedRequest* }, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = { *Executing*, *Other* }, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = *No*, Protected = *No*, Context = *Interrupt*, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = *Yes*, Context = *Interrupt*, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *Yes*, *No* }, Protected = *No*, Context = { *Interrupt*, *NestedRequest* }, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = *Yes*, Protected = *No*, Context = { *Interrupt*, *NestedRequest* }, State = *Ready*, Timer = *Active*, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Disabled*

ThreadDispatchDisabled

- Id = { *Invalid*, *Executing*, *Other* }, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, Context = *Interrupt*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Enabled*
- Id = *Executing*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Enabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *Yes*, *No* }, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Enabled*
- Id = *Other*, Dormant = *No*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = *Yes*, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = *Enabled*

NestedRequestNotDormant

- Id = *Executing*, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *Yes*, *No* }, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

- Id = *Other*, Dormant = *Yes*, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = *Yes*, Protected = *No*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

NestedRequestNotProtected

- Id = *Executing*, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = *No*, Protected = *Yes*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = *Yes*, Terminating = { *Yes*, *No* }, Protected = *Yes*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }
- Id = *Other*, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = *Yes*, Protected = *Yes*, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

NestedRequestNeedsLifeChanging

- Id = { *Executing*, *Other* }, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = *No*, Terminating = *No*, Protected = { *Yes*, *No* }, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

NestedRequestNoNestedExit

- Id = *Executing*, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = *Yes*, Protected = { *Yes*, *No* }, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

NestedRequestNeedsTask

- Id = *Invalid*, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }, Restarting = { *Yes*, *No* }, Terminating = { *Yes*, *No* }, Protected = { *Yes*, *No* }, Context = *NestedRequest*, State = { *Ready*, *Blocked*, *Enqueued* }, Timer = { *Inactive*, *Active* }, RealPriority = { *Initial*, *Changed* }, ThreadDispatch = { *Disabled*, *Enabled* }

5.1.312 spec:/rtems/task/req/resume

spec:/rtems/task/req/resume

When the rtems_task_resume() (see: spec:/rtems/task/if/resume) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_resume

5.1.312.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

Suspended

Yes While the task specified by the id parameter is suspended.

No While the task specified by the id parameter is not suspended.

5.1.312.2 post-conditions

Status

Ok The return status of rtems_task_resume shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_task_resume shall be RTEMS_INVALID_ID.

IncStat The return status of rtems_task_resume shall be RTEMS_INCORRECT_STATE.

5.1.312.3 transition-map

Status = *Ok*

- Id = *Task*, Suspended = *Yes*

Status = *IncStat*

- Id = *Task*, Suspended = *No*

Status = *InvId*

- Id = *Invalid*, Suspended = NA

5.1.313 spec:/rtems/task/req/self

spec:/rtems/task/req/self

The rtems_task_self directive call shall return the object identifier of the calling task.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - rtems_task_self

5.1.314 spec:/rtems/task/req/self-define

spec:/rtems/task/req/self-define

While the RTEMS_SELF constant is used as a task identifier parameter in a directive call, it shall be associated with the calling task.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_SELF

Traced design component: RTEMSAPIClassicTasks - rtems_task_delete

Traced design component: RTEMSAPIClassicTasks - rtems_task_get_affinity

Traced design component: RTEMSAPIClassicTasks - rtems_task_get_priority

Traced design component: RTEMSAPIClassicTasks - rtems_task_get_scheduler

Traced design component: RTEMSAPIClassicTasks - rtems_task_is_suspended

Traced design component: RTEMSAPIClassicTasks - rtems_task_restart

Traced design component: RTEMSAPIClassicTasks - rtems_task_resume

Traced design component: RTEMSAPIClassicTasks - rtems_task_set_affinity

Traced design component: RTEMSAPIClassicTasks - rtems_task_set_priority

Traced design component: RTEMSAPIClassicTasks - rtems_task_set_scheduler

Traced design component: RTEMSAPIClassicTasks - rtems_task_suspend

Traced design component: RTEMSAPIClassicEvent - rtems_event_send

Traced design component: event_8h - rtems_event_system_send

Traced design component: RTEMSAPIClassicSignal - rtems_signal_send

5.1.315 spec:/rtems/task/req/set-affinity

spec:/rtems/task/req/set-affinity

When the rtems_task_set_affinity() (see: [spec:/rtems/task/if/set-affinity](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_set_affinity

5.1.315.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

CPUSetSize

Askew While the cpusetsize parameter is not an integral multiple of the size of long.

Normal While the cpusetsize parameter is an integral multiple of the size of long, while the cpusetsize parameter is less than or equal to the maximum processor set size storable in the system.

Huge While the cpusetsize parameter is an integral multiple of the size of long, while the cpusetsize parameter is greater than the maximum processor set size storable in the system.

CPUSetOnline

Supported While the intersection of the processor set specified by the cpuset-size and cpuset parameters and the set of online processors represents an affinity set supported by the home scheduler of the task specified by the id parameter at some point during the rtems_task_set_affinity call.

Unsupported While the intersection of the processor set specified by the cpusetsize and cpuset parameters and the set of online processors represents an affinity set not supported by the home scheduler of the task specified by the id parameter at some point during the rtems_task_set_affinity call.

CPUSetHuge

NotZero While the processor set specified by the cpusetsize and cpuset parameters contains at least one processor which is not storable in a processor set supported by the system.

Zero While the processor set specified by the cpusetsize and cpuset parameters contains no processor which is not storable in a processor set supported by the system.

CPUSet

Valid While the cpuset parameter references an object of type cpu_set_t.

Null While the cpuset parameter is equal to NULL.

5.1.315.2 post-conditions

Status

Ok The return status of rtems_task_set_affinity shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_set_affinity shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_set_affinity shall be RTEMS_INVALID_ID.

InvNum The return status of rtems_task_set_affinity shall be RTEMS_INVALID_NUMBER.

SetAffinity

Set The affinity set of the task specified by the id parameter shall be set with respect to the home scheduler of the task at some point during the rtems_task_set_affinity call.

Nop No task affinity shall be modified by the rtems_task_set_affinity call.

5.1.315.3 transition-map

Status = *Ok*, SetAffinity = *Set*

- Id = *Task*, CPUSetSize = *Normal*, CPUSetOnline = *Supported*, CPUSetHuge = NA, CPUSet = *Valid*
- Id = *Task*, CPUSetSize = *Huge*, CPUSetOnline = *Supported*, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*

Status = *InvNum*, SetAffinity = *Nop*

- Id = *Task*, CPUSetSize = *Askew*, CPUSetOnline = { *Supported*, *Unsupported* }, CPUSetHuge = NA, CPUSet = *Valid*
- Id = *Task*, CPUSetSize = *Normal*, CPUSetOnline = *Unsupported*, CPUSetHuge = NA, CPUSet = *Valid*
- Id = *Task*, CPUSetSize = *Huge*, CPUSetOnline = *Unsupported*, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*

Status = *InvId*, SetAffinity = *Nop*

- Id = *Invalid*, CPUSetSize = { *Askew*, *Normal* }, CPUSetOnline = NA, CPUSetHuge = NA, CPUSet = *Valid*
- Id = *Invalid*, CPUSetSize = *Huge*, CPUSetOnline = NA, CPUSetHuge = { *NotZero*, *Zero* }, CPUSet = *Valid*

Status = *InvAddr*, SetAffinity = *Nop*

- Id = { *Invalid*, *Task* }, CPUSetSize = { *Askew*, *Normal*, *Huge* }, CPUSetOnline = NA, CPUSetHuge = NA, CPUSet = *Null*

5.1.316 spec:/rtems/task/req/set-priority

spec:/rtems/task/req/set-priority

When the rtems_task_set_priority() (see: [spec:/rtems/task/if/set-priority](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_set_priority

Traced design component: RTEMSAPIClassicTasks - RTEMS_CURRENT_PRIORITY

5.1.316.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

State

Dormant While the task specified by the id parameter is dormant.

Ready While the task specified by the id parameter is ready.

Scheduled While the task specified by the id parameter is scheduled.

Blocked While the task specified by the id parameter is blocked.

NewPriority

Current While the value of the new_priority parameter is equal to RTEMS_CURRENT_PRIORITY.

Other While the value of the new_priority parameter is not equal to RTEMS_CURRENT_PRIORITY.

TaskPriority

High While the value of the new_priority parameter is a valid task priority with respect to the home scheduler of the task specified by the id parameter when the new priority is set, while the value of the new_priority parameter is higher than the task priority with respect to the home scheduler of the task specified by the id parameter at time when the scheduler evaluates the new priority.

Equal While the value of the new_priority parameter is a valid task priority with respect to the home scheduler of the task specified by the id parameter when the new priority is set, while the value of the new_priority parameter is equal to the task priority with respect to the home scheduler of the task specified by the id parameter at time when the scheduler evaluates the new priority.

Low While the value of the new_priority parameter is a valid task priority with respect to the home scheduler of the task specified by the id parameter when the new priority is set, while the value of the new_priority parameter is lower than the task priority with respect to the home scheduler of the task specified by the id parameter at time when the scheduler evaluates the new priority.

Invalid While the value of the new_priority parameter is an invalid task priority with respect to the home scheduler of the task specified by the id parameter when the new priority is evaluated.

OldPriority

Valid While the old_priority parameter references an object of type rtems_task_priority.

Null While the old_priority parameter is equal to NULL.

5.1.316.2 post-conditions

Status

Ok The return status of rtems_task_set_priority shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_set_priority shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_set_priority shall be RTEMS_INVALID_ID.

InvPrio The return status of rtems_task_set_priority shall be RTEMS_INVALID_PRIORITY.

Priority

Set The real priority of the task specified by the id parameter shall be set to the value specified by the new_priority parameter at some point during the rtems_task_set_priority call.

Nop No real priority of a task shall be modified by the rtems_task_set_priority call.

OldPriorityObj

Set The value of the object referenced by the old_priority parameter shall be set after the return of the rtems_task_set_priority call to the current priority of the task specified by the id parameter at some point during the call and before the real priority is modified by the call if it is modified by the call.

Nop Objects referenced by the old_priority parameter in past calls to rtems_task_set_priority shall not be accessed by the rtems_task_set_priority call.

5.1.316.3 transition-map

Status = *InvPrio*, Priority = *Nop*, OldPriorityObj = *Set*

- Id = *Task*, State = { *Dormant*, *Ready*, *Scheduled*, *Blocked* }, NewPriority = *Other*, TaskPriority = *Invalid*, OldPriority = *Valid*

Status = *Ok*, Priority = *Set*, OldPriorityObj = *Set*

- Id = *Task*, State = { *Dormant*, *Ready*, *Scheduled*, *Blocked* }, NewPriority = *Other*, TaskPriority = { *High*, *Equal*, *Low* }, OldPriority = *Valid*

Status = *Ok*, Priority = *Nop*, OldPriorityObj = *Set*

- Id = *Task*, State = { *Dormant*, *Ready*, *Scheduled*, *Blocked* }, NewPriority = *Current*, TaskPriority = NA, OldPriority = *Valid*

Status = *InvId*, Priority = *Nop*, OldPriorityObj = *Nop*

- Id = *Invalid*, State = NA, NewPriority = { *Current*, *Other* }, TaskPriority = NA, OldPriority = *Valid*

Status = *InvAddr*, Priority = *Nop*, OldPriorityObj = *Nop*

- Id = *Invalid*, State = NA, NewPriority = { *Current*, *Other* }, TaskPriority = NA, OldPriority = *Null*
- Id = *Task*, State = { *Dormant*, *Ready*, *Scheduled*, *Blocked* }, NewPriority = *Current*, TaskPriority = NA, OldPriority = *Null*
- Id = *Task*, State = { *Dormant*, *Ready*, *Scheduled*, *Blocked* }, NewPriority = *Other*, TaskPriority = { *High*, *Equal*, *Low*, *Invalid* }, OldPriority = *Null*

5.1.317 spec:/rtems/task/req/set-scheduler

spec:/rtems/task/req/set-scheduler

When the rtems_task_set_scheduler() (see: spec:/rtems/task/if/set-scheduler) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_set_scheduler

5.1.317.1 pre-conditions

TaskId

Task While the task_id parameter is associated with a task.

Invalid While the task_id parameter is not associated with a task.

Scheduler

Home While the scheduler specified by the scheduler_id parameter is the home scheduler of the task specified by the task_id parameter.

Other While the scheduler specified by the scheduler_id parameter is not the home scheduler of the task specified by the task_id parameter.

SchedulerHasCPU

Yes While the scheduler specified by the scheduler_id parameter owns at least one processor.

No While the scheduler specified by the scheduler_id parameter owns no processor.

SchedulerId

Scheduler While the scheduler_id parameter is associated with a scheduler.

Invalid While the scheduler_id parameter is not associated with a scheduler.

Priority

Valid While the task priority specified by the priority parameter is valid with respect to the scheduler specified by the scheduler_id parameter.

Invalid While the task priority specified by the priority parameter is invalid with respect to the scheduler specified by the scheduler_id parameter.

HomePriority

Real While the current priority of the task specified by the task_id parameter consists only of the real priority.

More While the current priority of the task specified by the task_id parameter consists of more than the real priority.

EligiblePriorities

OnlyOne While the set of eligible priorities of the task specified by the task_id parameter consists of exactly the current priority.

More While the set of eligible priorities of the task specified by the task_id parameter consists of more than the current priority.

Pinned

Yes While the task specified by the task_id parameter is pinned.

No While the task specified by the task_id parameter is not pinned.

TaskState

Ready While the task specified by the task_id parameter is ready.

Blocked While the task specified by the task_id parameter is blocked, while the task specified by the task_id parameter is not enqueued on a wait queue.

Enqueued While the task specified by the task_id parameter is blocked, while the task specified by the task_id parameter is enqueued on a wait queue.

AffinitySupported

Yes While the affinity set of the task specified by the task_id parameter is supported by the scheduler specified by the scheduler_id parameter.

No While the affinity set of the task specified by the task_id parameter is not supported by the scheduler specified by the scheduler_id parameter.

5.1.317.2 post-conditions

Status

Ok The return status of rtems_task_set_scheduler shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_set_scheduler shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_set_scheduler shall be RTEMS_INVALID_ID.

InvPrio The return status of rtems_task_set_scheduler shall be RTEMS_INVALID_PRIORITY.

InUse The return status of rtems_task_set_scheduler shall be RTEMS_RESOURCE_IN_USE.

Unsat The return status of rtems_task_set_scheduler shall be RTEMS_UNSATISFIED.

Scheduler

Set The home scheduler of the task specified by the task_id parameter shall be set to the scheduler specified by the scheduler_id parameter at some point during the rtems_task_set_scheduler call.

Nop No home scheduler of a task shall be modified by the rtems_task_set_scheduler call.

Priority

Set The real priority of the task specified by the task_id parameter shall be set to the priority specified by the priority parameter at some point during the rtems_task_set_scheduler call.

Nop No task priority shall be modified by the rtems_task_set_scheduler call.

5.1.317.3 skip-reasons

OnlyOneCPU Where the system was built with SMP support disabled, exactly one scheduler is present in an application using exactly one processor.

HomeScheduler The home scheduler of a task owns always at least one processor and the affinity set of the task is always supported its home scheduler.

5.1.317.4 transition-map

Status = *Ok*, Scheduler = *Set*, Priority = *Set*

- TaskId = *Task*, Scheduler = { *Home*, *Other* }, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *No*, TaskState = { *Ready*, *Blocked* }, AffinitySupported = *Yes*

Status = *Unsat*, Scheduler = *Nop*, Priority = *Nop*

- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *No*, TaskState = { *Ready*, *Blocked* }, AffinitySupported = *No*
- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = *No*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *No*, TaskState = { *Ready*, *Blocked* }, AffinitySupported = { *Yes*, *No* }

Status = *InUse*, Scheduler = *Nop*, Priority = *Nop*

- TaskId = *Task*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *Yes*, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = *Yes*
- TaskId = *Task*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *No*, TaskState = *Enqueued*, AffinitySupported = *Yes*
- TaskId = *Task*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *More*, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = *Yes*
- TaskId = *Task*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *More*, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = *Yes*
- TaskId = *Task*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *More*, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = *Yes*
- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = { *Yes*, *No* }, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *Yes*, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = { *Yes*, *No* }
- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = { *Yes*, *No* }, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *OnlyOne*, Pinned = *No*, TaskState = *Enqueued*, AffinitySupported = { *Yes*, *No* }
- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = { *Yes*, *No* }, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *Real*, EligiblePriorities = *More*, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = { *Yes*, *No* }

- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = { *Yes*, *No* }, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = *More*, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = { *Yes*, *No* }

Status = *InvPrio*, Scheduler = *Nop*, Priority = *Nop*

- TaskId = *Task*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Invalid*, HomePriority = { *Real*, *More* }, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = *Yes*
- TaskId = *Task*, Scheduler = *Other*, SchedulerHasCPU = { *Yes*, *No* }, SchedulerId = *Scheduler*, Priority = *Invalid*, HomePriority = { *Real*, *More* }, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = { *Yes*, *No* }
- TaskId = *Invalid*, Scheduler = { *Home*, *Other* }, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Invalid*, HomePriority = NA, EligiblePriorities = NA, Pinned = NA, TaskState = NA, AffinitySupported = *Yes*
- TaskId = *Invalid*, Scheduler = *Other*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Invalid*, HomePriority = NA, EligiblePriorities = NA, Pinned = NA, TaskState = NA, AffinitySupported = { *Yes*, *No* }
- TaskId = *Invalid*, Scheduler = *Other*, SchedulerHasCPU = *No*, SchedulerId = *Scheduler*, Priority = *Invalid*, HomePriority = NA, EligiblePriorities = NA, Pinned = NA, TaskState = NA, AffinitySupported = { *Yes*, *No* }

Status = *InvId*, Scheduler = *Nop*, Priority = *Nop*

- TaskId = *Task*, Scheduler = NA, SchedulerHasCPU = NA, SchedulerId = *Invalid*, Priority = NA, HomePriority = { *Real*, *More* }, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = NA
- TaskId = *Task*, Scheduler = NA, SchedulerHasCPU = NA, SchedulerId = *Invalid*, Priority = NA, HomePriority = { *Real*, *More* }, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = NA
- TaskId = *Invalid*, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = NA, EligiblePriorities = NA, Pinned = NA, TaskState = NA, AffinitySupported = *Yes*
- TaskId = *Invalid*, Scheduler = NA, SchedulerHasCPU = NA, SchedulerId = *Invalid*, Priority = NA, HomePriority = NA, EligiblePriorities = NA, Pinned = NA, TaskState = NA, AffinitySupported = NA
- TaskId = *Invalid*, Scheduler = *Other*, SchedulerHasCPU = { *Yes*, *No* }, SchedulerId = *Scheduler*, Priority = *Valid*, HomePriority = NA, EligiblePriorities = NA, Pinned = NA, TaskState = NA, AffinitySupported = { *Yes*, *No* }

HomeScheduler

- TaskId = { *Task*, *Invalid* }, Scheduler = *Home*, SchedulerHasCPU = *Yes*, SchedulerId = { *Scheduler*, *Invalid* }, Priority = { *Valid*, *Invalid* }, HomePriority = { *Real*, *More* }, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = *No*

- TaskId = { *Task*, *Invalid* }, Scheduler = *Home*, SchedulerHasCPU = *No*, SchedulerId = { *Scheduler*, *Invalid* }, Priority = { *Valid*, *Invalid* }, HomePriority = { *Real*, *More* }, EligiblePriorities = { *OnlyOne*, *More* }, Pinned = { *Yes*, *No* }, TaskState = { *Ready*, *Blocked*, *Enqueued* }, AffinitySupported = { *Yes*, *No* }

5.1.318 spec:/rtems/task/req/start

spec:/rtems/task/req/start

When the rtems_task_start() (see: spec:/rtems/task/if/start) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_start

5.1.318.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

EntryPoint

Valid While the task entry point specified by the entry_point parameter is valid.

Null While the task entry point specified by the entry_point parameter is equal to NULL.

Argument

Pointer While the entry point argument specified by the argument parameter is a pointer.

Number While the entry point argument specified by the argument parameter is a 32-bit number.

Dormant

Yes While the task specified by the id parameter is dormant.

No While the task specified by the id parameter is not dormant.

Suspended

Yes While the task specified by the id parameter is suspended.

No While the task specified by the id parameter is not suspended.

5.1.318.2 post-conditions

Status

Ok The return status of rtems_task_start shall be RTEMS_SUCCESSFUL.

InvAddr The return status of rtems_task_start shall be RTEMS_INVALID_ADDRESS.

InvId The return status of rtems_task_start shall be RTEMS_INVALID_ID.

IncStat The return status of rtems_task_start shall be RTEMS_INCORRECT_STATE.

EntryPoint

Set The entry point of the task specified by the id parameter shall be set to the function specified by the entry_point parameter before the task is unblocked by the rtems_task_start call.

Nop No entry point of a task shall be modified by the rtems_task_start call.

Argument

Set The entry point argument of the task specified by the id parameter shall be set to the value specified by the argument parameter before the task is unblocked by the rtems_task_start call.

Nop No entry point argument of a task shall be modified by the rtems_task_start call.

Unblock

Yes The task specified by the id parameter shall be unblocked by the rtems_task_start call.

Nop No task shall be unblocked by the rtems_task_start call.

StartExtensions

Yes The thread start user extensions shall be invoked by the rtems_task_start call.

Nop The thread start user extensions shall not be invoked by the rtems_task_start call.

5.1.318.3 transition-map

Status = **Ok**, EntryPoint = **Set**, Argument = **Set**, Unblock = **Yes**, StartExtensions = **Yes**

- Id = **Task**, EntryPoint = **Valid**, Argument = { **Pointer**, **Number** }, Dormant = **Yes**, Suspended = { **Yes**, **No** }

Status = **IncStat**, EntryPoint = **Nop**, Argument = **Nop**, Unblock = **Nop**, StartExtensions = **Nop**

- Id = **Task**, EntryPoint = **Valid**, Argument = { **Pointer**, **Number** }, Dormant = **No**, Suspended = { **Yes**, **No** }

Status = *InvId*, EntryPoint = *Nop*, Argument = *Nop*, Unblock = *Nop*, StartExtensions = *Nop*

- Id = *Invalid*, EntryPoint = *Valid*, Argument = { *Pointer*, *Number* }, Dormant = NA, Suspended = NA

Status = *InvAddr*, EntryPoint = *Nop*, Argument = *Nop*, Unblock = *Nop*, StartExtensions = *Nop*

- Id = { *Invalid*, *Task* }, EntryPoint = *Null*, Argument = { *Pointer*, *Number* }, Dormant = { *Yes*, *No* }, Suspended = { *Yes*, *No* }

5.1.319 spec:/rtems/task/req/storage-alignment

spec:/rtems/task/req/storage-alignment

The RTEMS_TASK_STORAGE_ALIGNMENT constant shall expand to a constant expression which evaluates to the value of CPU_STACK_ALIGNMENT.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTasks - RTEMS_TASK_STORAGE_ALIGNMENT

5.1.320 spec:/rtems/task/req/storage-size

spec:/rtems/task/req/storage-size

When the RTEMS_TASK_STORAGE_SIZE() (see: [spec:/rtems/task/if/storage-size](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - RTEMS_TASK_STORAGE_SIZE

5.1.320.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

Suspended

Yes While the task specified by the id parameter is suspended.

No While the task specified by the id parameter is not suspended.

5.1.320.2 post-conditions

Status

Ok The return status of rtems_task_suspend shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_task_suspend shall be RTEMS_INVALID_ID.

AlrdySus The return status of rtems_task_suspend shall be RTEMS_ALREADY_SUSPENDED.

5.1.320.3 transition-map

Status = *AlrdySus*

- Id = *Task*, Suspended = *Yes*

Status = *Ok*

- Id = *Task*, Suspended = *No*

Status = *InvId*

- Id = *Invalid*, Suspended = NA

5.1.321 spec:/rtems/task/req/suspend

spec:/rtems/task/req/suspend

When the rtems_task_suspend() (see: spec:/rtems/task/if/suspend) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_suspend

5.1.321.1 pre-conditions

Id

Invalid While the id parameter is not associated with a task.

Task While the id parameter is associated with a task.

Suspended

Yes While the task specified by the id parameter is suspended.

No While the task specified by the id parameter is not suspended.

5.1.321.2 post-conditions

Status

Ok The return status of rtems_task_suspend shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_task_suspend shall be RTEMS_INVALID_ID.

AlrdySus The return status of rtems_task_suspend shall be RTEMS_ALREADY_SUSPENDED.

5.1.321.3 transition-map

Status = *AlrdySus*

- Id = *Task*, Suspended = *Yes*

Status = *Ok*

- Id = *Task*, Suspended = *No*

Status = *InvId*

- Id = *Invalid*, Suspended = NA

5.1.322 spec:/rtems/task/req/wake-after

spec:/rtems/task/req/wake-after

When the rtems_task_wake_after() (see: [spec:/rtems/task/if/wake-after](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_wake_after

Traced design component: RTEMSAPIClassicTasks - RTEMS_YIELD_PROCESSOR

5.1.322.1 pre-conditions

Ticks

Yield While the ticks parameter is equal to RTEMS_YIELD_PROCESSOR.

Interval While the ticks parameter is not equal to RTEMS_YIELD_PROCESSOR.

Suspended

Yes While the calling task is suspended.

No While the calling task is not suspended.

5.1.322.2 post-conditions

Status

Ok The return status of rtems_task_wake_after shall be RTEMS_SUCCESSFUL.

Timer

Inactive The timer of the calling task shall be inactive.

Ticks The timer of the calling task shall be active using the clock tick.

Expire

Relative The timer of the calling task shall expire at the time point specified by the sum of the current clock tick and the interval specified by the ticks parameter.

Scheduler

Block The calling task shall be blocked by the scheduler exactly once by the rtems_task_wake_after call.

Yield The calling task shall yield by the scheduler exactly once by the rtems_task_wake_after call.

Nop The calling task shall not carry out a scheduler operation through the rtems_task_wake_after call.

5.1.322.3 transition-map

Status = *Ok*, Timer = *Inactive*, Expire = NA, Scheduler = *Nop*

- Ticks = *Yield*, Suspended = *Yes*

Status = *Ok*, Timer = *Inactive*, Expire = NA, Scheduler = *Yield*

- Ticks = *Yield*, Suspended = *No*

Status = *Ok*, Timer = *Ticks*, Expire = *Relative*, Scheduler = *Nop*

- Ticks = *Interval*, Suspended = *Yes*

Status = *Ok*, Timer = *Ticks*, Expire = *Relative*, Scheduler = *Block*

- Ticks = *Interval*, Suspended = *No*

5.1.323 spec:/rtems/task/req/wake-when

spec:/rtems/task/req/wake-when

When the rtems_task_wake_when() (see: spec:/rtems/task/if/wake-when) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTasks - rtems_task_wake_when

5.1.323.1 pre-conditions

TODSet

Yes While the CLOCK_REALTIME was set at least once.

No While the CLOCK_REALTIME was never set.

TOD

Valid While the time_buffer parameter references an object of type rtems_time_of_day.

Null While the time_buffer parameter is equal to NULL.

TODOBJ

Future While the object referenced by the time_buffer parameter specifies a valid time of day in the future.

PastOrNow While the object referenced by the time_buffer parameter specifies a valid time of day in the past or at the time of the rtems_task_wake_when call.

Invalid While the object referenced by the time_buffer parameter specifies an invalid time of day.

5.1.323.2 post-conditions

Status

Ok The return status of rtems_task_wake_when shall be RTEMS_SUCCESSFUL.

NotDef The return status of rtems_task_wake_when shall be RTEMS_NOT_DEFINED.

InvAddr The return status of rtems_task_wake_when shall be RTEMS_INVALID_ADDRESS.

InvClock The return status of rtems_task_wake_when shall be RTEMS_INVALID_CLOCK.

Timer

Inactive The timer of the calling task shall be inactive.

Realtime The timer of the calling task shall be active using the CLOCK_REALTIME.

Expire

Absolute The timer of the calling task shall expire at the time point specified by the time_buffer parameter.

Scheduler

Block The calling task shall be blocked by the scheduler exactly once by the rtems_task_wake_when call.

Nop The calling task shall not be altered by the scheduler by the rtems_task_wake_when call.

5.1.323.3 transition-map

Status = *Ok*, Timer = *Realtime*, Expire = *Absolute*, Scheduler = *Block*

- TODSet = *Yes*, TOD = *Valid*, TODOObj = *Future*

Status = *InvClock*, Timer = *Inactive*, Expire = NA, Scheduler = *Nop*

- TODSet = *Yes*, TOD = *Valid*, TODOObj = { *PastOrNow*, *Invalid* }

Status = *InvAddr*, Timer = *Inactive*, Expire = NA, Scheduler = *Nop*

- TODSet = *Yes*, TOD = *Null*, TODOObj = NA

Status = *NotDef*, Timer = *Inactive*, Expire = NA, Scheduler = *Nop*

- TODSet = *No*, TOD = *Valid*, TODOObj = { *Future*, *PastOrNow*, *Invalid* }
- TODSet = *No*, TOD = *Null*, TODOObj = NA

5.1.324 spec:/rtems/timer/req/cancel

spec:/rtems/timer/req/cancel

When the rtems_timer_cancel() (see: spec:/rtems/timer/if/cancel) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_cancel

5.1.324.1 pre-conditions

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Context

None While the Timer Service Routine has never been scheduled since creation of the timer. See also none.

Interrupt While the timer is in interrupt context.

Server While the timer is in server context.

Clock

None While the timer has never been scheduled since creation of the timer.

Ticks While the clock used to determine when the timer will fire is the ticks based clock.

Realtime While the clock used to determine when the timer will fire is the realtime clock.

State

Inactive While the timer is in inactive state.

Scheduled While the timer is in scheduled state.

Pending While the timer is in pending state.

5.1.324.2 post-conditions

Status

Ok The return status of rtems_timer_cancel shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_timer_cancel shall be RTEMS_INVALID_ID.

Context

None The timer shall have never been scheduled. See also none.

Interrupt The timer shall be in interrupt context.

Server The timer shall be in server context.

Nop Objects referenced by the id parameter in past call to rtems_timer_cancel shall not be accessed by the rtems_timer_cancel call. See also Nop.

Clock

None The timer shall have never been scheduled.

Ticks The timer shall use the ticks based clock.

Realtime The timer shall use the realtime clock.

Nop Objects referenced by the id parameter in past call to rtems_timer_cancel shall not be accessed by the rtems_timer_cancel call.

State

Inactive The timer shall be in inactive state.

Nop Objects referenced by the id parameter in past call to rtems_timer_cancel shall not be accessed by the rtems_timer_cancel call.

5.1.324.3 skip-reasons

NotExist The pre-condition combination of context, clock and state cannot be produced and does therefore not exist.

5.1.324.4 transition-map

Status = *Ok*, Context = *None*, Clock = *None*, State = *Inactive*

- Id = *Valid*, Context = *None*, Clock = *None*, State = *Inactive*

Status = *Ok*, Context = *Interrupt*, Clock = *Ticks*, State = *Inactive*

- Id = *Valid*, Context = *Interrupt*, Clock = *Ticks*, State = { *Inactive*, *Scheduled* }

Status = *Ok*, Context = *Interrupt*, Clock = *Realtime*, State = *Inactive*

- Id = *Valid*, Context = *Interrupt*, Clock = *Realtime*, State = { *Inactive*, *Scheduled* }

Status = *Ok*, Context = *Server*, Clock = *Ticks*, State = *Inactive*

- Id = *Valid*, Context = *Server*, Clock = *Ticks*, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *Ok*, Context = *Server*, Clock = *Realtime*, State = *Inactive*

- Id = *Valid*, Context = *Server*, Clock = *Realtime*, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvId*, Context = *Nop*, Clock = *Nop*, State = *Nop*

- Id = *Invalid*, Context = *None*, Clock = *None*, State = *Inactive*
- Id = *Invalid*, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Id = *Invalid*, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

NotExist

- Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = { *Scheduled*, *Pending* }
- Id = { *Valid*, *Invalid* }, Context = *None*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *None*, State = { *Inactive*, *Scheduled*, *Pending* }
- Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = *Pending*

5.1.325 spec:/rtems/timer/req/create

spec:/rtems/timer/req/create

When the rtems_timer_create() (see: [spec:/rtems/timer/if/create](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_create

5.1.325.1 pre-conditions

Name

Valid While the name parameter is valid.

Invalid While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

Free

Yes While the system has at least one inactive timer object available.

No While the system has no inactive timer object available.

5.1.325.2 post-conditions

Status

Ok The return status of rtems_timer_create shall be RTEMS_SUCCESSFUL.

InvName The return status of rtems_timer_create shall be RTEMS_INVALID_NAME.

InvAddr The return status of rtems_timer_create shall be RTEMS_INVALID_ADDRESS.

TooMany The return status of rtems_timer_create shall be RTEMS_TOO_MANY.

Name

Valid The unique object name shall identify the timer created by the rtems_timer_create call.

Invalid The unique object name shall not identify a timer.

IdVar

Set The value of the object referenced by the id parameter shall be set to the object identifier of the created timer after the return of the rtems_timer_create call.

Nop Objects referenced by the id parameter in past calls to rtems_timer_create shall not be accessed by the rtems_timer_create call.

5.1.325.3 transition-map

Status = *Ok*, Name = *Valid*, IdVar = *Set*

- Name = *Valid*, Id = *Valid*, Free = *Yes*

Status = *TooMany*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Valid*, Free = *No*

Status = *InvAddr*, Name = *Invalid*, IdVar = *Nop*

- Name = *Valid*, Id = *Null*, Free = { *Yes*, *No* }

Status = *InvName*, Name = *Invalid*, IdVar = *Nop*

- Name = *Invalid*, Id = { *Valid*, *Null* }, Free = { *Yes*, *No* }

5.1.326 spec:/rtems/timer/req/delete

spec:/rtems/timer/req/delete

When the rtems_timer_delete() (see: spec:/rtems/timer/if/delete) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_delete

5.1.326.1 pre-conditions

Id

NoObj While the id parameter is not associated with a timer.

Timer While the id parameter is associated with a timer.

5.1.326.2 post-conditions

Status

Ok The return status of rtems_timer_delete shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_timer_delete shall be RTEMS_INVALID_ID.

Name

Valid The unique object name shall identify a timer.

Invalid The unique object name shall not identify a timer.

5.1.326.3 transition-map

Status = *InvId*, Name = *Valid*

- Id = *NoObj*

Status = *Ok*, Name = *Invalid*

- Id = *Timer*

5.1.327 spec:/rtems/timer/req/fire-after

spec:/rtems/timer/req/fire-after

When the rtems_timer_fire_after() (see: [spec:/rtems/timer/if/fire-after](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_fire_after

5.1.327.1 pre-conditions

Ticks

Valid While the ticks parameter is a positive (greater 0) number.

Is0 While the ticks parameter is 0.

Routine

Valid While the routine parameter references an object of type rtems_timer_service_routine_entry.

Null While the routine parameter is NULL..

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Context

None While the Timer Service Routine has never been scheduled since creation of the timer. See also none.

Interrupt While the timer is in interrupt context.

Server While the timer is in server context.

Clock

None While the timer has never been scheduled since creation of the timer.

Ticks While the clock used to determine when the timer will fire is the ticks based clock.

Realtime While the clock used to determine when the timer will fire is the realtime clock.

State

Inactive While the timer is in inactive state.

Scheduled While the timer is in scheduled state.

Pending While the timer is in pending state.

5.1.327.2 post-conditions

Status

Ok The return status of rtems_timer_fire_after shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_timer_fire_after shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_timer_fire_after shall be RTEMS_INVALID_ADDRESS.

InvNum The return status of rtems_timer_fire_after shall be RTEMS_INVALID_NUMBER.

Context

None The timer shall have never been scheduled. See also none.

Interrupt The timer shall be in interrupt context.

Server The timer shall be in server context.

Nop Objects referenced by the parameters in the past call to rtems_timer_fire_after shall not be accessed by the rtems_timer_fire_after call. See also Nop.

Clock

None The timer shall have never been scheduled.

Ticks The timer shall use the ticks based clock.

Realtime The timer shall use the realtime clock.

Nop Objects referenced by the parameters in the past call to rtems_timer_fire_after shall not be accessed by the rtems_timer_fire_after call.

State

Scheduled The timer shall be in scheduled state.

Nop Objects referenced by the parameters in the past call to rtems_timer_fire_after shall not be accessed by the rtems_timer_fire_after call.

Interval

Param The Timer Service Routine shall be invoked the number of ticks (see tick), which are provided by the ticks parameter in the past call to rtems_timer_fire_after, after a point in time during the execution of the rtems_timer_fire_after call.

Nop If and when the Timer Service Routine will be invoked shall not be changed by the past call to rtems_timer_fire_after.

Routine

Param The function reference used to invoke the Timer Service Routine when the timer will fire shall be the one provided by the routine parameter in the past call to rtems_timer_fire_after.

Nop The function reference used for any invocation of the Timer Service Routine shall not be changed by the past call to rtems_timer_fire_after.

UserData

Param The user data argument for invoking the Timer Service Routine when the timer will fire shall be the one provided by the user_data parameter in the past call to rtems_timer_fire_after.

Nop The user data argument used for any invocation of the Timer Service Routine shall not be changed by the past call to rtems_timer_fire_after.

5.1.327.3 skip-reasons

NotExist The pre-condition combination of context, clock and state cannot be produced and does therefore not exist.

5.1.327.4 transition-map

Status = *Ok*, Context = *Interrupt*, Clock = *Ticks*, State = *Scheduled*, Interval = *Param*, Routine = *Param*, UserData = *Param*

- Ticks = *Valid*, Routine = *Valid*, Id = *Valid*, Context = *None*, Clock = *None*, State = *Inactive*
- Ticks = *Valid*, Routine = *Valid*, Id = *Valid*, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Ticks = *Valid*, Routine = *Valid*, Id = *Valid*, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvId*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Ticks = *Valid*, Routine = *Valid*, Id = *Invalid*, Context = *None*, Clock = *None*, State = *Inactive*
- Ticks = *Valid*, Routine = *Valid*, Id = *Invalid*, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Ticks = *Valid*, Routine = *Valid*, Id = *Invalid*, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvAddr*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Ticks = *Valid*, Routine = *Null*, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- Ticks = *Valid*, Routine = *Null*, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Ticks = *Valid*, Routine = *Null*, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvNum*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Ticks = *Is0*, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- Ticks = *Is0*, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Ticks = *Is0*, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

NotExist

- Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = { *Scheduled*, *Pending* }
- Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *None*, State = { *Inactive*, *Scheduled*, *Pending* }

- Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = *Pending*

5.1.328 spec:/rtems/timer/req/fire-order

spec:/rtems/timer/req/fire-order

The timer scheduled at the same processor with the same expiration time point shall fire in FIFO order.

rationale: N/A

functional-type: function

5.1.329 spec:/rtems/timer/req/fire-when

spec:/rtems/timer/req/fire-when

When the rtems_timer_fire_when() (see: [spec:/rtems/timer/if/fire-when](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_fire_when

5.1.329.1 pre-conditions

RtClock

Set While the realtime clock is set to a valid time-of-day.

Unset While the realtime clock has never been set.

Routine

Valid While the routine parameter references an object of type rtems_timer_service_routine_entry.

Null While the routine parameter is NULL..

WallTime

Valid While the wall_time parameter references a time at least one second in the future but not later than the last second of the year 2105. (Times after 2105 are invalid.)

Invalid While the wall_time parameter is invalid.

Past While the wall_time parameter references a time in the current second or in the past but not earlier than 1988. (Times before 1988 are invalid.)

Null While the wall_time parameter is 0.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Context

None While the Timer Service Routine has never been scheduled since creation of the timer. See also none.

Interrupt While the timer is in interrupt context.

Server While the timer is in server context.

Clock

None While the timer has never been scheduled since creation of the timer.

Ticks While the clock used to determine when the timer will fire is the ticks based clock.

Realtime While the clock used to determine when the timer will fire is the realtime clock.

State

Inactive While the timer is in inactive state.

Scheduled While the timer is in scheduled state.

Pending While the timer is in pending state.

5.1.329.2 post-conditions

Status

Ok The return status of rtems_timer_fire_when shall be RTEMS_SUCCESSFUL.

NotDef The return status of rtems_timer_fire_when shall be RTEMS_NOT_DEFINED

InvId The return status of rtems_timer_fire_when shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_timer_fire_when shall be RTEMS_INVALID_ADDRESS.

InvClock The return status of rtems_timer_fire_when shall be RTEMS_INVALID_CLOCK.

Context

None The timer shall have never been scheduled. See also none.

Interrupt The timer shall be in interrupt context.

Server The timer shall be in server context.

Nop Objects referenced by parameters in the past call to `rtems_timer_fire_when` shall not be accessed by the `rtems_timer_fire_when` call. See also Nop.

Clock

None The timer shall have never been scheduled.

Ticks The timer shall use the ticks based clock.

Realtime The timer shall use the realtime clock.

Nop Objects referenced by parameters in the past call to `rtems_timer_fire_when` shall not be accessed by the `rtems_timer_fire_when` call.

State

Scheduled The timer shall be in scheduled state.

Nop Objects referenced by parameters in the past call to `rtems_timer_fire_when` shall not be accessed by the `rtems_timer_fire_when` call.

WallTime

Param The Timer Service Routine shall be invoked at the wall time (see real-time clock) (ignoring ticks), which was provided by the `wall_time` parameter in the past call to `rtems_timer_fire_when`.

Nop If and when the Timer Service Routine will be invoked shall not be changed by the past call to `rtems_timer_fire_when`.

Routine

Param The function reference used to invoke the Timer Service Routine when the timer will fire shall be the one provided by the `routine` parameter in the past call to `rtems_timer_fire_when`.

Nop The function reference used for any invocation of the Timer Service Routine shall not be changed by the past call to `rtems_timer_fire_when`.

UserData

Param The user data argument for invoking the Timer Service Routine when the timer will fire shall be the one provided by the `user_data` parameter in the past call to `rtems_timer_fire_when`.

Nop The user data argument used for any invocation of the Timer Service Routine shall not be changed by the past call to `rtems_timer_fire_when`.

5.1.329.3 skip-reasons

NotExist The pre-condition combination of context, clock and state cannot be produced and does therefore not exist.

ClockNotSet When the realtime clock is not set, the timer cannot be in pre-condition *Clock Realtime* because it is not possible to call the directives `rtems_timer_fire_when` or `rtems_timer_server_fire_when` with status code `RTEMS_SUCCESSFUL`.

5.1.329.4 transition-map

Status = *Ok*, Context = *Interrupt*, Clock = *Realtime*, State = *Scheduled*, WallTime = *Param*, Routine = *Param*, UserData = *Param*

- RtClock = *Set*, Routine = *Valid*, WallTime = *Valid*, Id = *Valid*, Context = *None*, Clock = *None*, State = *Inactive*
- RtClock = *Set*, Routine = *Valid*, WallTime = *Valid*, Id = *Valid*, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- RtClock = *Set*, Routine = *Valid*, WallTime = *Valid*, Id = *Valid*, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvId*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- RtClock = *Set*, Routine = *Valid*, WallTime = *Valid*, Id = *Invalid*, Context = *None*, Clock = *None*, State = *Inactive*
- RtClock = *Set*, Routine = *Valid*, WallTime = *Valid*, Id = *Invalid*, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- RtClock = *Set*, Routine = *Valid*, WallTime = *Valid*, Id = *Invalid*, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvClock*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- RtClock = *Set*, Routine = *Valid*, WallTime = { *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- RtClock = *Set*, Routine = *Valid*, WallTime = { *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- RtClock = *Set*, Routine = *Valid*, WallTime = { *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *NotDef*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = *Ticks*, State = { *Inactive*, *Scheduled* }

- RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = *Ticks*, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvAddr*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = *Null*, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = *Null*, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = *Null*, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- RtClock = *Set*, Routine = *Null*, WallTime = { *Valid*, *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- RtClock = *Set*, Routine = *Null*, WallTime = { *Valid*, *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- RtClock = *Set*, Routine = *Null*, WallTime = { *Valid*, *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

NotExist

- RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = { *Scheduled*, *Pending* }
- RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *None*, State = { *Inactive*, *Scheduled*, *Pending* }
- RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = *Pending*
- RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = *Ticks*, State = *Pending*

ClockNotSet

- RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *Realtime*, State = { *Inactive*, *Scheduled*, *Pending* }

5.1.330 spec:/rtems/timer/req/ident

spec:/rtems/timer/req/ident

The rtems_timer_ident directive shall identify an Classic API timer class object by its name as specified by /rtems/req/ident-local.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTimer - rtems_timer_ident

5.1.331 spec:/rtems/timer/req/initiate-server

spec:/rtems/timer/req/initiate-server

When the rtems_timer_initiate_server() (see: spec:/rtems/timer/if/initiate-server) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_initiate_server

Traced design component: RTEMSAPIClassicTimer - RTEMS_TIMER_SERVER_DEFAULT_PRIORITY

5.1.331.1 pre-conditions

Priority

Valid While the priority parameter is valid.

Default While the priority parameter is equal to RTEMS_TIMER_SERVER_DEFAULT_PRIORITY.

Invalid While the priority parameter is invalid.

Stack

Allocatable While the stack_size parameter specifies a task stack size which is allocatable by the system.

TooLarge While the stack_size parameter specifies a task stack size which is not allocatable by the system.

Started

Yes While the Timer Server task is started.

No While the Timer Server task is not started.

TaskObj

Available While the system has at least one inactive task object available.

Unavailable While the system has no inactive task object available.

5.1.331.2 post-conditions

Status

Ok The return status of rtems_timer_initiate_server shall be RTEMS_SUCCESSFUL.

IncStat The return status of rtems_timer_initiate_server shall be RTEMS_INCORRECT_STATE

InvPrio The return status of rtems_timer_initiate_server shall be RTEMS_INVALID_PRIORITY.

TooMany The return status of rtems_timer_initiate_server shall be RTEMS_TOO_MANY.

Unsat The return status of rtems_timer_initiate_server shall be RTEMS_UNSATISFIED.

Started

Yes The Timer Server task shall be started after the rtems_timer_initiate_server call.

No The Timer Server task shall not be started after the rtems_timer_initiate_server call.

TaskPriorities

Set The priority of the Timer Server task shall be equal to the priority specified by the priority parameter in the rtems_timer_initiate_server call.

Nop The priority of the Timer Server task shall not be modified by the rtems_timer_initiate_server call.

TaskStack

Set The stack size of the Timer Server task shall be greater than or equal to the stack size specified by the stack_size parameter in the rtems_timer_initiate_server call.

Nop The stack size of the Timer Server task shall not be modified by the rtems_timer_initiate_server call.

TaskAttributes

Set The task attributes of the Timer Server task shall be specified by the attribute_set parameter in the rtems_timer_initiate_server call.

Nop The task attributes of the Timer Server task shall not be modified by the rtems_timer_initiate_server call.

5.1.331.3 transition-map

Status = *Ok*, Started = *Yes*, TaskPrio = *Set*, TaskStack = *Set*, TaskAttr = *Set*

- Priority = { *Valid*, *Default* }, Stack = *Allocatable*, Started = *No*, TaskObj = *Available*

Status = *Unsat*, Started = *No*, TaskPrio = NA, TaskStack = NA, TaskAttr = NA

- Priority = { *Valid*, *Default* }, Stack = *TooLarge*, Started = *No*, TaskObj = *Available*

Status = *TooMany*, Started = *No*, TaskPrio = NA, TaskStack = NA, TaskAttr = NA

- Priority = { *Valid*, *Default* }, Stack = { *Allocatable*, *TooLarge* }, Started = *No*, TaskObj = *Unavailable*

Status = *InvPrio*, Started = *No*, TaskPrio = NA, TaskStack = NA, TaskAttr = NA

- Priority = *Invalid*, Stack = { *Allocatable*, *TooLarge* }, Started = *No*, TaskObj = { *Available*, *Unavailable* }

Status = *IncStat*, Started = *Yes*, TaskPrio = *Nop*, TaskStack = *Nop*, TaskAttr = *Nop*

- Priority = { *Valid*, *Default*, *Invalid* }, Stack = { *Allocatable*, *TooLarge* }, Started = *Yes*, TaskObj = { *Available*, *Unavailable* }

5.1.332 spec:/rtems/timer/req/reset

spec:/rtems/timer/req/reset

When the rtems_timer_reset() (see: spec:/rtems/timer/if/reset) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_reset

5.1.332.1 pre-conditions

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Context

None While the Timer Service Routine has never been scheduled since creation of the timer. See also none.

Interrupt While the timer is in interrupt context.

Server While the timer is in server context.

Clock

None While the timer has never been scheduled since creation of the timer.

Ticks While the clock used to determine when the timer will fire is the ticks based clock.

Realtime While the clock used to determine when the timer will fire is the realtime clock.

State

Inactive While the timer is in inactive state.

Scheduled While the timer is in scheduled state.

Pending While the timer is in pending state.

5.1.332.2 post-conditions

Status

Ok The return status of rtems_timer_reset shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_timer_reset shall be RTEMS_INVALID_ID.

NotDef The return status of rtems_timer_reset shall be RTEMS_NOT_DEFINED

Context

None The timer shall have never been scheduled. See also none.

Interrupt The timer shall be in interrupt context.

Server The timer shall be in server context.

Nop Objects referenced by the id parameter in past call to rtems_timer_reset shall not be accessed by the rtems_timer_reset call. See also Nop.

Clock

None The timer shall have never been scheduled.

Ticks The timer shall use the ticks based clock.

Realtime The timer shall use the realtime clock.

Nop Objects referenced by the id parameter in past call to rtems_timer_reset shall not be accessed by the rtems_timer_reset call.

State

Scheduled The timer shall be in scheduled state.

Nop Objects referenced by the id parameter in past call to rtems_timer_reset shall not be accessed by the rtems_timer_reset call.

Interval

Last The Timer Service Routine shall be invoked the same number of ticks (see tick), as defined by the last scheduled interval, after a point in time during the execution of the rtems_timer_reset call.

Nop If and when the Timer Service Routine will be invoked shall not be changed by the past call to `rtems_timer_reset`.

Routine

Last The function reference used to invoke the Timer Service Routine when the timer will fire shall be the same one as the last one scheduled.

Nop The function reference used for any invocation of the Timer Service Routine shall not be changed by the past call to `rtems_timer_reset`.

UserData

Last The user data argument for invoking the Timer Service Routine when the timer will fire shall be the same as the last scheduled user data argument.

Nop The user data argument used for any invocation of the Timer Service Routine shall not be changed by the past call to `rtems_timer_reset`.

5.1.332.3 skip-reasons

NotExist The pre-condition combination of context, clock and state cannot be produced and does therefore not exist.

5.1.332.4 transition-map

Status = *Ok*, Context = *Interrupt*, Clock = *Ticks*, State = *Scheduled*, Interval = *Last*, Routine = *Last*, UserData = *Last*

- Id = *Valid*, Context = *Interrupt*, Clock = *Ticks*, State = { *Inactive*, *Scheduled* }

Status = *Ok*, Context = *Server*, Clock = *Ticks*, State = *Scheduled*, Interval = *Last*, Routine = *Last*, UserData = *Last*

- Id = *Valid*, Context = *Server*, Clock = *Ticks*, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *NotDef*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Id = *Valid*, Context = *None*, Clock = *None*, State = *Inactive*
- Id = *Valid*, Context = *Interrupt*, Clock = *Realtime*, State = { *Inactive*, *Scheduled* }
- Id = *Valid*, Context = *Server*, Clock = *Realtime*, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvId*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Id = *Invalid*, Context = *None*, Clock = *None*, State = *Inactive*
- Id = *Invalid*, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Id = *Invalid*, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

NotExist

- Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = { *Scheduled*, *Pending* }
- Id = { *Valid*, *Invalid* }, Context = *None*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *None*, State = { *Inactive*, *Scheduled*, *Pending* }
- Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = *Pending*

5.1.333 spec:/rtems/timer/req/server-fire-after

spec:/rtems/timer/req/server-fire-after

When the rtems_timer_server_fire_after() (see: spec:/rtems/timer/if/server-fire-after) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_server_fire_after

5.1.333.1 pre-conditions

Server

Init While the Timer Server task has been successfully initialized by a call to rtems_timer_initiate_server.

NotInit While the Timer Server task has not been initialized and does not exist.

Ticks

Valid While the ticks parameter is a positive (greater 0) number.

Is0 While the ticks parameter is 0.

Routine

Valid While the routine parameter references an object of type rtems_timer_service_routine_entry.

Null While the routine parameter is NULL..

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Context

None While the Timer Service Routine has never been scheduled since creation of the timer. See also none.

Interrupt While the timer is in interrupt context.

Server While the timer is in server context.

Clock

None While the timer has never been scheduled since creation of the timer.

Ticks While the clock used to determine when the timer will fire is the ticks based clock.

Realtime While the clock used to determine when the timer will fire is the realtime clock.

State

Inactive While the timer is in inactive state.

Scheduled While the timer is in scheduled state.

Pending While the timer is in pending state.

5.1.333.2 post-conditions

Status

Ok The return status of rtems_timer_server_fire_after shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_timer_server_fire_after shall be RTEMS_INVALID_ID.

InvAddr The return status of rtems_timer_server_fire_after shall be RTEMS_INVALID_ADDRESS.

InvNum The return status of rtems_timer_server_fire_after shall be RTEMS_INVALID_NUMBER.

IncStat The return status of rtems_timer_server_fire_after shall be RTEMS_INCORRECT_STATE.

Context

None The timer shall have never been scheduled. See also none.

Interrupt The timer shall be in interrupt context.

Server The timer shall be in server context.

Nop Objects referenced by the parameters in the past call to rtems_timer_server_fire_after shall not be accessed by the rtems_timer_server_fire_after call. See also Nop.

Clock

None The timer shall have never been scheduled.

Ticks The timer shall use the ticks based clock.

Realtime The timer shall use the realtime clock.

Nop Objects referenced by the parameters in the past call to rtems_timer_server_fire_after shall not be accessed by the rtems_timer_server_fire_after call.

State

Scheduled The timer shall be in scheduled state.

Nop Objects referenced by the parameters in the past call to rtems_timer_server_fire_after shall not be accessed by the rtems_timer_server_fire_after call.

Interval

Param The Timer Service Routine shall be invoked the number of ticks (see tick), which are provided by the ticks parameter in the past call to rtems_timer_server_fire_after, after a point in time during the execution of the rtems_timer_server_fire_after call.

Nop If and when the Timer Service Routine will be invoked shall not be changed by the past call to rtems_timer_server_fire_after.

Routine

Param The function reference used to invoke the Timer Service Routine when the timer will fire shall be the one provided by the routine parameter in the past call to rtems_timer_server_fire_after.

Nop The function reference used for any invocation of the Timer Service Routine shall not be changed by the past call to rtems_timer_server_fire_after.

UserData

Param The user data argument for invoking the Timer Service Routine when the timer will fire shall be the one provided by the user_data parameter in the past call to rtems_timer_server_fire_after.

Nop The user data argument used for any invocation of the Timer Service Routine shall not be changed by the past call to rtems_timer_server_fire_after.

5.1.333.3 skip-reasons

NotExist The pre-condition combination of context, clock and state cannot be produced and does therefore not exist.

NoServer If the server task is not initialized, the timer cannot be in server context.

5.1.333.4 transition-map

Status = *Ok*, Context = *Server*, Clock = *Ticks*, State = *Scheduled*, Interval = *Param*, Routine = *Param*, UserData = *Param*

- Server = *Init*, Ticks = *Valid*, Routine = *Valid*, Id = *Valid*, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, Ticks = *Valid*, Routine = *Valid*, Id = *Valid*, Context = *Interrupt*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled*}
- Server = *Init*, Ticks = *Valid*, Routine = *Valid*, Id = *Valid*, Context = *Server*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled, Pending*}

Status = *InvId*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *Init*, Ticks = *Valid*, Routine = *Valid*, Id = *Invalid*, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, Ticks = *Valid*, Routine = *Valid*, Id = *Invalid*, Context = *Interrupt*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled*}
- Server = *Init*, Ticks = *Valid*, Routine = *Valid*, Id = *Invalid*, Context = *Server*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled, Pending*}

Status = *InvAddr*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *Init*, Ticks = *Valid*, Routine = *Null*, Id = {*Valid, Invalid*}, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, Ticks = *Valid*, Routine = *Null*, Id = {*Valid, Invalid*}, Context = *Interrupt*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled*}
- Server = *Init*, Ticks = *Valid*, Routine = *Null*, Id = {*Valid, Invalid*}, Context = *Server*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled, Pending*}

Status = *IncStat*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *NotInit*, Ticks = {*Valid, Is0*}, Routine = {*Valid, Null*}, Id = {*Valid, Invalid*}, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *NotInit*, Ticks = {*Valid, Is0*}, Routine = {*Valid, Null*}, Id = {*Valid, Invalid*}, Context = *Interrupt*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled*}

Status = *InvNum*, Context = *Nop*, Clock = *Nop*, State = *Nop*, Interval = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *Init*, Ticks = *Is0*, Routine = {*Valid, Null*}, Id = {*Valid, Invalid*}, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, Ticks = *Is0*, Routine = {*Valid, Null*}, Id = {*Valid, Invalid*}, Context = *Interrupt*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled*}
- Server = *Init*, Ticks = *Is0*, Routine = {*Valid, Null*}, Id = {*Valid, Invalid*}, Context = *Server*, Clock = {*Ticks, Realtime*}, State = {*Inactive, Scheduled, Pending*}

NotExist

- Server = { *Init*, *NotInit* }, Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = { *Scheduled*, *Pending* }
- Server = { *Init*, *NotInit* }, Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- Server = { *Init*, *NotInit* }, Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *None*, State = { *Inactive*, *Scheduled*, *Pending* }
- Server = { *Init*, *NotInit* }, Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = *Pending*

NoServer

- Server = *NotInit*, Ticks = { *Valid*, *Is0* }, Routine = { *Valid*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

5.1.334 spec:/rtems/timer/req/server-fire-when

spec:/rtems/timer/req/server-fire-when

When the rtems_timer_server_fire_when() (see: spec:/rtems/timer/if/server-fire-when) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicTimer - rtems_timer_server_fire_when

5.1.334.1 pre-conditions

Server

Init While the Timer Server task has been successfully initialized by a call to rtems_timer_initiate_server.

NotInit While the Timer Server task has not been initialized and does not exist.

RtClock

Set While the realtime clock is set to a valid time-of-day.

Unset While the realtime clock has never been set.

Routine

Valid While the routine parameter references an object of type rtems_timer_service_routine_entry.

Null While the routine parameter is NULL..

WallTime

Valid While the wall_time parameter references a time at least one second in the future but not later than the last second of the year 2105. (Times after 2105 are invalid.)

Invalid While the wall_time parameter is invalid.

Past While the wall_time parameter references a time in the current second or in the past but not earlier than 1988. (Times before 1988 are invalid.)

Null While the wall_time parameter is 0.

Id

Valid While the id parameter is valid.

Invalid While the id parameter is invalid.

Context

None While the Timer Service Routine has never been scheduled since creation of the timer. See also none.

Interrupt While the timer is in interrupt context.

Server While the timer is in server context.

Clock

None While the timer has never been scheduled since creation of the timer.

Ticks While the clock used to determine when the timer will fire is the ticks based clock.

Realtime While the clock used to determine when the timer will fire is the realtime clock.

State

Inactive While the timer is in inactive state.

Scheduled While the timer is in scheduled state.

Pending While the timer is in pending state.

5.1.334.2 post-conditions

Status

Ok The return status of rtems_timer_server_fire_when shall be RTEMS_SUCCESSFUL.

NotDef The return status of rtems_timer_server_fire_when shall be RTEMS_NOT_DEFINED

InvId The return status of rtems_timer_server_fire_when shall be RTEMS_INVALID_ID.

InvAddr The return status of `rtems_timer_server_fire_when` shall be `RTEMS_INVALID_ADDRESS`.

InvClock The return status of `rtems_timer_server_fire_when` shall be `RTEMS_INVALID_CLOCK`.

IncStat The return status of `rtems_timer_server_fire_when` shall be `RTEMS_INCORRECT_STATE`.

Context

None The timer shall have never been scheduled. See also `none`.

Interrupt The timer shall be in interrupt context.

Server The timer shall be in server context.

Nop Objects referenced by parameters in the past call to `rtems_timer_server_fire_when` shall not be accessed by the `rtems_timer_server_fire_when` call. See also `Nop`.

Clock

None The timer shall have never been scheduled.

Ticks The timer shall use the ticks based clock.

Realtime The timer shall use the realtime clock.

Nop Objects referenced by parameters in the past call to `rtems_timer_server_fire_when` shall not be accessed by the `rtems_timer_server_fire_when` call.

State

Scheduled The timer shall be in scheduled state.

Nop Objects referenced by parameters in the past call to `rtems_timer_server_fire_when` shall not be accessed by the `rtems_timer_server_fire_when` call.

WallTime

Param The Timer Service Routine shall be invoked at the wall time (see real-time clock) (ignoring ticks), which was provided by the `wall_time` parameter in the past call to `rtems_timer_server_fire_when`.

Nop If and when the Timer Service Routine will be invoked shall not be changed by the past call to `rtems_timer_server_fire_when`.

Routine

Param The function reference used to invoke the Timer Service Routine when the timer will fire shall be the one provided by the `routine` parameter in the past call to `rtems_timer_server_fire_when`.

Nop The function reference used for any invocation of the Timer Service Routine shall not be changed by the past call to `rtems_timer_server_fire_when`.

UserData

Param The user data argument for invoking the Timer Service Routine when the timer will fire shall be the one provided by the `user_data` parameter in the past call to `rtems_timer_server_fire_when`.

Nop The user data argument used for any invocation of the Timer Service Routine shall not be changed by the past call to `rtems_timer_server_fire_when`.

5.1.334.3 skip-reasons

NotExist The pre-condition combination of context, clock and state cannot be produced and does therefore not exist.

ClockNotSet When the realtime clock is not set, the timer cannot be in pre-condition `Clock Realtime` because it is not possible to call the directives `rtems_timer_fire_when` or `rtems_timer_server_fire_when` with status code `RTEMS_SUCCESSFUL`.

NoServer If the server task is not initialized, the timer cannot be in server context.

5.1.334.4 transition-map

Status = `Ok`, Context = `Server`, Clock = `Realtime`, State = `Scheduled`, WallTime = `Param`, Routine = `Param`, UserData = `Param`

- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = `Valid`, Id = `Valid`, Context = `None`, Clock = `None`, State = `Inactive`
- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = `Valid`, Id = `Valid`, Context = `Interrupt`, Clock = { `Ticks`, `Realtime` }, State = { `Inactive`, `Scheduled` }
- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = `Valid`, Id = `Valid`, Context = `Server`, Clock = { `Ticks`, `Realtime` }, State = { `Inactive`, `Scheduled`, `Pending` }

Status = `InvId`, Context = `Nop`, Clock = `Nop`, State = `Nop`, WallTime = `Nop`, Routine = `Nop`, UserData = `Nop`

- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = `Valid`, Id = `Invalid`, Context = `None`, Clock = `None`, State = `Inactive`
- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = `Valid`, Id = `Invalid`, Context = `Interrupt`, Clock = { `Ticks`, `Realtime` }, State = { `Inactive`, `Scheduled` }
- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = `Valid`, Id = `Invalid`, Context = `Server`, Clock = { `Ticks`, `Realtime` }, State = { `Inactive`, `Scheduled`, `Pending` }

Status = `InvClock`, Context = `Nop`, Clock = `Nop`, State = `Nop`, WallTime = `Nop`, Routine = `Nop`, UserData = `Nop`

- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = { `Invalid`, `Past` }, Id = { `Valid`, `Invalid` }, Context = `None`, Clock = `None`, State = `Inactive`
- Server = `Init`, RtClock = `Set`, Routine = `Valid`, WallTime = { `Invalid`, `Past` }, Id = { `Valid`, `Invalid` }, Context = `Interrupt`, Clock = { `Ticks`, `Realtime` }, State = { `Inactive`, `Scheduled` }

- Server = *Init*, RtClock = *Set*, Routine = *Valid*, WallTime = { *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *NotDef*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *Init*, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = *Ticks*, State = { *Inactive*, *Scheduled* }
- Server = *Init*, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = *Ticks*, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *InvAddr*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *Init*, RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = *Null*, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = *Null*, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Server = *Init*, RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = *Null*, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- Server = *Init*, RtClock = *Set*, Routine = *Null*, WallTime = { *Valid*, *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *Init*, RtClock = *Set*, Routine = *Null*, WallTime = { *Valid*, *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Server = *Init*, RtClock = *Set*, Routine = *Null*, WallTime = { *Valid*, *Invalid*, *Past* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

Status = *IncStat*, Context = *Nop*, Clock = *Nop*, State = *Nop*, WallTime = *Nop*, Routine = *Nop*, UserData = *Nop*

- Server = *NotInit*, RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = *Inactive*
- Server = *NotInit*, RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled* }
- Server = *NotInit*, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = *Ticks*, State = { *Inactive*, *Scheduled* }

NotExist

- Server = { *Init*, *NotInit* }, RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = *None*, State = { *Scheduled*, *Pending* }
- Server = { *Init*, *NotInit* }, RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *None*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }
- Server = { *Init*, *NotInit* }, RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *None*, State = { *Inactive*, *Scheduled*, *Pending* }
- Server = { *Init*, *NotInit* }, RtClock = *Set*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = { *Ticks*, *Realtime* }, State = *Pending*
- Server = { *Init*, *NotInit* }, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = *Ticks*, State = *Pending*

ClockNotSet

- Server = *Init*, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = { *Interrupt*, *Server* }, Clock = *Realtime*, State = { *Inactive*, *Scheduled*, *Pending* }
- Server = *NotInit*, RtClock = *Unset*, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Interrupt*, Clock = *Realtime*, State = { *Inactive*, *Scheduled*, *Pending* }

NoServer

- Server = *NotInit*, RtClock = { *Set*, *Unset* }, Routine = { *Valid*, *Null* }, WallTime = { *Valid*, *Invalid*, *Past*, *Null* }, Id = { *Valid*, *Invalid* }, Context = *Server*, Clock = { *Ticks*, *Realtime* }, State = { *Inactive*, *Scheduled*, *Pending* }

5.1.335 spec:/rtems/type/req/id-none

spec:/rtems/type/req/id-none

The RTEMS_ID_NONE constant shall be a constant expression which evaluates to an invalid object identifier.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicTypes - RTEMS_ID_NONE

5.1.336 spec:/rtems/userext/req/create

spec:/rtems/userext/req/create

When the rtems_extension_create() (see: [spec:/rtems/userext/if/create](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicUserExt - rtems_extension_create

5.1.336.1 pre-conditions

Name

Valid While the name parameter is valid.

Invalid While the name parameter is invalid.

Id

Valid While the id parameter references an object of type rtems_id.

Null While the id parameter is NULL.

Table

TdSw While the extension_table parameter references an object of type rtems_extensions_table, while all extensions except the thread switch extension of the referenced object are set to NULL or the address of a corresponding extension, while the thread switch extension of the referenced object is set to the address of a thread switch extension.

NoTdSw While the extension_table parameter references an object of type rtems_id, while all extensions except the thread switch extension of the referenced object are set to NULL or the address of a corresponding extension, while the thread switch extension of the referenced object is set to NULL.

Null While the extension_table parameter is NULL.

Free

Yes While the system has at least one inactive extension set object available.

No While the system has no inactive extension set object available.

5.1.336.2 post-conditions

Status

Ok The return status of rtems_extension_create shall be RTEMS_SUCCESSFUL.

InvName The return status of rtems_extension_create shall be RTEMS_INVALID_NAME.

InvAddr The return status of rtems_extension_create shall be RTEMS_INVALID_ADDRESS.

TooMany The return status of rtems_extension_create shall be RTEMS_TOO_MANY.

Name

Valid The unique object name shall identify the extension set created by the rtems_extension_create call.

Invalid The unique object name shall not identify an extension set.

IdVar

Set The value of the object referenced by the extension_table parameter shall be set to the object identifier of the created extension set after the return of the rtems_extension_create call.

Nop Objects referenced by the extension_table parameter in past calls to rtems_extension_create shall not be accessed by the rtems_extension_create call.

5.1.336.3 transition-map

Status = **Ok**, Name = **Valid**, IdVar = **Set**

- Name = **Valid**, Id = **Valid**, Table = { **TdSw**, **NoTdSw** }, Free = **Yes**

Status = **TooMany**, Name = **Invalid**, IdVar = **Nop**

- Name = **Valid**, Id = **Valid**, Table = { **TdSw**, **NoTdSw** }, Free = **No**

Status = **InvAddr**, Name = **Invalid**, IdVar = **Nop**

- Name = **Valid**, Id = **Valid**, Table = **Null**, Free = { **Yes**, **No** }
- Name = **Valid**, Id = **Null**, Table = { **TdSw**, **NoTdSw**, **Null** }, Free = { **Yes**, **No** }

Status = **InvName**, Name = **Invalid**, IdVar = **Nop**

- Name = **Invalid**, Id = { **Valid**, **Null** }, Table = { **TdSw**, **NoTdSw**, **Null** }, Free = { **Yes**, **No** }

5.1.337 spec:/rtems/userext/req/delete

spec:/rtems/userext/req/delete

When the rtems_extension_delete() (see: [spec:/rtems/userext/if/delete](#)) directive is called.

rationale: N/A

functional-type: action

Traced design component: RTEMSAPIClassicUserExt - rtems_extension_delete

5.1.337.1 pre-conditions

Id

NoObj While the id parameter is not associated with an extension set.

ExtTdSw While the id parameter is associated with an extension set with a thread switch extension.

ExtNoTdSw While the id parameter is associated with an extension set without a thread switch extension.

5.1.337.2 post-conditions

Status

Ok The return status of rtems_extension_delete shall be RTEMS_SUCCESSFUL.

InvId The return status of rtems_extension_delete shall be RTEMS_INVALID_ID.

Name

Valid The unique object name shall identify an extension set.

Invalid The unique object name shall not identify an extension set.

5.1.337.3 transition-map

Status = *InvId*, Name = *Valid*

- Id = *NoObj*

Status = *Ok*, Name = *Invalid*

- Id = { *ExtTdSw*, *ExtNoTdSw* }

5.1.338 spec:/rtems/userext/req/fatal-iterate-remove

spec:/rtems/userext/req/fatal-iterate-remove

While the user extensions are iterated to invoke the fatal extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.339 spec:/rtems/userext/req/fatal-order

spec:/rtems/userext/req/fatal-order

The fatal user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.340 spec:/rtems/userext/req/ident

spec:/rtems/userext/req/ident

The rtems_extension_ident directive shall identify an Classic API user extension class object by its name as specified by /rtems/req/ident-local.

rationale: N/A

functional-type: function

Traced design component: RTEMSAPIClassicUserExt - rtems_extension_ident

5.1.341 spec:/rtems/userext/req/thread-begin-iterate-remove

spec:/rtems/userext/req/thread-begin-iterate-remove

While the user extensions are iterated to invoke the thread begin extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.342 spec:/rtems/userext/req/thread-begin-order

spec:/rtems/userext/req/thread-begin-order

The thread begin user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.343 spec:/rtems/userext/req/thread-create-allocator-owner

spec:/rtems/userext/req/thread-create-allocator-owner

The thread invoking the thread create user extensions shall be the owner of the allocator mutex.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/object/req/allocator-mutex*.

5.1.344 spec:/rtems/userext/req/thread-create-iterate-remove

spec:/rtems/userext/req/thread-create-iterate-remove

While the user extensions are iterated to invoke the thread create extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.345 spec:/rtems/userext/req/thread-create-order

spec:/rtems/userext/req/thread-create-order

The thread create user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.346 spec:/rtems/userext/req/thread-delete-allocator-owner

spec:/rtems/userext/req/thread-delete-allocator-owner

The thread invoking the thread delete user extensions shall be the owner of the allocator mutex.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/object/req/allocator-mutex*.

5.1.347 spec:/rtems/userext/req/thread-delete-iterate-remove

spec:/rtems/userext/req/thread-delete-iterate-remove

While the user extensions are iterated to invoke the thread delete extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.348 spec:/rtems/userext/req/thread-delete-order

spec:/rtems/userext/req/thread-delete-order

The thread delete user extensions shall be invoked in extension reverse order.

rationale: N/A

functional-type: function

5.1.349 spec:/rtems/userext/req/thread-exitted-iterate-remove

spec:/rtems/userext/req/thread-exitted-iterate-remove

While the user extensions are iterated to invoke the thread exitted extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.350 spec:/rtems/userext/req/thread-exitted-order

spec:/rtems/userext/req/thread-exitted-order

The thread exitted user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.351 spec:/rtems/userext/req/thread-restart-iterate-remove

spec:/rtems/userext/req/thread-restart-iterate-remove

While the user extensions are iterated to invoke the thread restart extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.352 spec:/rtems/userext/req/thread-restart-order

spec:/rtems/userext/req/thread-restart-order

The thread restart user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.353 spec:/rtems/userext/req/thread-start-iterate-remove

spec:/rtems/userext/req/thread-start-iterate-remove

While the user extensions are iterated to invoke the thread start extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.354 spec:/rtems/userext/req/thread-start-order

spec:/rtems/userext/req/thread-start-order

The thread start user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.355 spec:/rtems/userext/req/thread-switch-order

spec:/rtems/userext/req/thread-switch-order

The thread switch user extensions shall be invoked in extension forward order.

rationale: N/A

functional-type: function

5.1.356 spec:/rtems/userext/req/thread-terminate-iterate-remove

spec:/rtems/userext/req/thread-terminate-iterate-remove

While the user extensions are iterated to invoke the thread terminate extensions, when an extension set is removed, the removed extension set shall not be a next extension set for any iteration in progress after the removal.

rationale: N/A

functional-type: function

5.1.357 spec:/rtems/userext/req/thread-terminate-order

spec:/rtems/userext/req/thread-terminate-order

The thread terminate user extensions shall be invoked in extension reverse order.

rationale: N/A

functional-type: function

5.1.358 spec:/score/cpu/req/fatal-halt

spec:/score/cpu/req/fatal-halt

The _CPU_Fatal_halt() function shall halt or reset the target.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/cpu/if/group*.

This requirement is refined by the following requirements:

- *spec:/score/cpu/req/fatal-halt-sparc*

Traced design component: None

5.1.359 spec:/score/cpu/req/fatal-halt-sparc

spec:/score/cpu/req/fatal-halt-sparc

The _CPU_Fatal_halt() directive shall cause a system error halt with the primary exit code set to the fatal source and the secondary exit code set to the fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/cpu/req/fatal-halt*.

5.1.360 spec:/score/interr/req/terminate

spec:/score/interr/req/terminate

The `_Terminate()` function shall terminate the system.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement is refined by the following requirements:

- *spec:/score/interr/req/terminate-shutdown*
- *spec:/score/interr/req/terminate-shutdown-halt*
- *spec:/score/interr/req/terminate-state*
- *spec:/score/interr/req/terminate-userext*

Traced design component: RTEMSScoreIntErr - `_Terminate`

5.1.361 spec:/score/interr/req/terminate-shutdown

spec:/score/interr/req/terminate-shutdown

While `_Terminate()` executes, when the system state was set, a shutdown request to all configured processors shall be issued.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/interr/req/terminate*.

5.1.362 spec:/score/interr/req/terminate-shutdown-halt

spec:/score/interr/req/terminate-shutdown-halt

While `_Terminate()` executes, when the shutdown request was issued, the system shall halt.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/interr/req/terminate*.

5.1.363 spec:/score/interr/req/terminate-state

spec:/score/interr/req/terminate-state

While `_Terminate()` executes, when the fatal extensions were invoked, the system state shall be set to terminated.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/interr/req/terminate*.

5.1.364 spec:/score/interr/req/terminate-userext

spec:/score/interr/req/terminate-userext

When `_Terminate()` is called, the fatal user extensions shall be invoked with the fatal source and fatal code specified by the respective parameter.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/interr/req/terminate*.

5.1.365 spec:/score/mtx/req/seize-try

spec:/score/mtx/req/seize-try

When the calling thread tries to seize the mutex.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/mtx/if/group*.

5.1.365.1 pre-conditions

Protocol

Ceiling Where the mutex uses the priority ceiling locking protocol.

MrsP Where the mutex uses the MrsP locking protocol.

Other Where the mutex does not use the priority ceiling or MrsP locking protocol.

Discipline

FIFO Where the thread queue of the mutex uses the FIFO discipline.

Priority Where the thread queue of the mutex uses the priority discipline.

Recursive

Allowed Where a recursive seize of the mutex is allowed.

Unavailable Where a recursive seize of the mutex results in an unavailable status.

Deadlock Where a recursive seize of the mutex results in a deadlock status.

Owner

None While the mutex has no owner.

Caller While the owner of the mutex is the calling thread.

Other While the owner of the mutex is a thread other than the calling thread.

Priority

High While the calling thread has a current priority higher than the priority ceiling.

Equal While the calling thread has a current priority equal to the priority ceiling.

Low While the calling thread has a current priority lower than the priority ceiling.

5.1.365.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

MutexCeilingViolated The return status of the directive call shall be derived from STATUS_MUTEX_CEILING_VIOLATED.

Deadlock The return status of the directive call shall be derived from STATUS_DEADLOCK.

Unavailable The return status of the directive call shall be derived from `STATUSTUS_UNAVAILABLE`.

Owner

Other The owner of the mutex shall not be modified.

Caller The owner of the mutex shall be the calling thread.

None The mutex shall have no owner.

Priority

Nop The priorities of the calling thread shall not be modified.

Ceiling The calling thread shall use the priority ceiling of the mutex.

5.1.365.3 skip-reasons

CeilingOwner Where the mutex provides a priority ceiling, the owner of the mutex cannot have a current priority lower than the priority ceiling.

PriorityDisciplineByProtocol The priority ceiling and MrsP locking protocol requires a priority discipline.

5.1.365.4 transition-map

Status = `MutexCeilingViolated`, Owner = `None`, Priority = `Nop`

- Protocol = { `Ceiling`, `MrsP` }, Discipline = `Priority`, Recursive = { `Allowed`, `Unavailable`, `Deadlock` }, Owner = `None`, Priority = `High`

Status = `Unavailable`, Owner = `Caller`, Priority = `Nop`

- Protocol = { `Ceiling`, `MrsP` }, Discipline = `Priority`, Recursive = `Unavailable`, Owner = `Caller`, Priority = { `High`, `Equal` }
- Protocol = `Other`, Discipline = { `FIFO`, `Priority` }, Recursive = `Unavailable`, Owner = `Caller`, Priority = NA

Status = `Deadlock`, Owner = `Caller`, Priority = `Nop`

- Protocol = { `Ceiling`, `MrsP` }, Discipline = `Priority`, Recursive = `Deadlock`, Owner = `Caller`, Priority = { `High`, `Equal` }
- Protocol = `Other`, Discipline = { `FIFO`, `Priority` }, Recursive = `Deadlock`, Owner = `Caller`, Priority = NA

Status = `Ok`, Owner = `Caller`, Priority = `Ceiling`

- Protocol = { `Ceiling`, `MrsP` }, Discipline = `Priority`, Recursive = { `Allowed`, `Unavailable`, `Deadlock` }, Owner = `None`, Priority = { `Equal`, `Low` }

Status = `Ok`, Owner = `Caller`, Priority = `Nop`

- Protocol = { `Ceiling`, `MrsP` }, Discipline = `Priority`, Recursive = `Allowed`, Owner = `Caller`, Priority = { `High`, `Equal` }

- Protocol = *Other*, Discipline = { *FIFO*, *Priority* }, Recursive = *Allowed*, Owner = { *None*, *Caller* }, Priority = NA
- Protocol = *Other*, Discipline = { *FIFO*, *Priority* }, Recursive = { *Unavailable*, *Deadlock* }, Owner = *None*, Priority = NA

Status = *Unavailable*, Owner = *Other*, Priority = *Nop*

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = { *Allowed*, *Unavailable*, *Deadlock* }, Owner = *Other*, Priority = { *High*, *Equal*, *Low* }
- Protocol = *Other*, Discipline = { *FIFO*, *Priority* }, Recursive = { *Allowed*, *Unavailable*, *Deadlock* }, Owner = *Other*, Priority = NA

CeilingOwner

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = { *Allowed*, *Unavailable*, *Deadlock* }, Owner = *Caller*, Priority = *Low*

PriorityDisciplineByProtocol

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *FIFO*, Recursive = { *Allowed*, *Unavailable*, *Deadlock* }, Owner = { *None*, *Caller*, *Other* }, Priority = { *High*, *Equal*, *Low* }

5.1.366 spec:/score mtx/req/seize-wait

spec:/score mtx/req/seize-wait

When the calling thread tries to seize the mutex.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- spec:/score/tq/req/enqueue-ceiling*
- spec:/score/tq/req/enqueue-deadlock*
- spec:/score/tq/req/enqueue-fifo*
- spec:/score/tq/req/enqueue-mrsp*
- spec:/score/tq/req/enqueue-priority*
- spec:/score/tq/req/enqueue-priority-inherit*

This requirement refines *spec:/score/mtx/if/group*.

5.1.366.1 pre-conditions

Protocol

None Where the mutex does not use a locking protocol.

Inherit Where the mutex uses the priority inheritance locking protocol.

Ceiling Where the mutex uses the priority ceiling locking protocol.

MrsP Where the mutex uses the MrsP locking protocol.

Discipline

FIFO Where the thread queue of the mutex uses the FIFO discipline.

Priority Where the thread queue of the mutex uses the priority discipline.

DeadlockResult

Status Where a detected deadlock results in a return with a status code.

Fatal Where a detected deadlock results in a fatal error.

Recursive

Allowed Where a recursive seize of the mutex is allowed.

Deadlock Where a recursive seize of the mutex results in a deadlock.

Owner

None While the mutex has no owner.

Caller While the owner of the mutex is the calling thread.

Other While the owner of the mutex is a thread other than the calling thread.

Deadlock While the attempt to seize the mutex results in a deadlock.

Priority

High While the calling thread has a current priority higher than the priority ceiling.

Equal While the calling thread has a current priority equal to the priority ceiling.

Low While the calling thread has a current priority lower than the priority ceiling.

5.1.366.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

MutexCeilingViolated The return status of the directive call shall be derived from STATUS_MUTEX_CEILING_VIOLATED.

DeadlockStatus The return status of the directive call shall be derived from STATUS_DEADLOCK.

DeadlockFatal The system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_THREAD_QUEUE_DEADLOCK fatal code.

Enqueued

No The calling thread shall not be enqueued on the thread queue of the mutex.

FIFO The calling thread shall be enqueued in FIFO order.

Priority The calling thread shall be enqueued in priority order.

PriorityInherit The calling thread shall be enqueued in priority order with priorit inheritance.

PriorityCeiling The calling thread shall be enqueued in priority order according to the priority ceiling locking protocol.

PriorityMrsP The calling thread shall be enqueued in priority order according to the MrsP locking protocol.

Owner

Other The owner of the mutex shall not be modified.

Caller The owner of the mutex shall be the calling thread.

None The mutex shall have no owner.

Priority

Nop The priorities of the calling thread shall not be modified.

Ceiling The calling thread shall use the priority ceiling of the mutex.

5.1.366.3 skip-reasons

CeilingOwner Where the mutex provides a priority ceiling, the owner of the mutex cannot have a current priority lower than the priority ceiling.

PriorityDisciplineByProtocol The priority ceiling and MrsP locking protocol requires a priority discipline.

5.1.366.4 transition-map

Status = *MutexCeilingViolated*, Enqueued = *No*, Owner = *None*, Priority = *Nop*

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *None*, Priority = *High*

Status = *MutexCeilingViolated*, Enqueued = *No*, Owner = *Other*, Priority = *Nop*

- Protocol = *MrsP*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = { *Other*, *Deadlock* }, Priority = *High*

Status = NA, Enqueued = *PriorityMrsP*, Owner = *Other*, Priority = *Ceiling*

- Protocol = *MrsP*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *Other*, Priority = { *Equal*, *Low* }

Status = NA, Enqueued = *FIFO*, Owner = *Other*, Priority = *Nop*

- Protocol = *None*, Discipline = *FIFO*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *Other*, Priority = NA

Status = NA, Enqueued = *Priority*, Owner = *Other*, Priority = *Nop*

- Protocol = *None*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *Other*, Priority = NA

Status = NA, Enqueued = *PriorityInherit*, Owner = *Other*, Priority = *Nop*

- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *Other*, Priority = NA

Status = NA, Enqueued = *PriorityCeiling*, Owner = *Other*, Priority = *Nop*

- Protocol = *Ceiling*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *Other*, Priority = { *High*, *Equal*, *Low* }

Status = *DeadlockStatus*, Enqueued = *No*, Owner = *Caller*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, DeadlockResult = *Status*, Recursive = *Deadlock*, Owner = *Caller*, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = *Status*, Recursive = *Deadlock*, Owner = *Caller*, Priority = NA
- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, DeadlockResult = *Status*, Recursive = *Deadlock*, Owner = *Caller*, Priority = { *High*, *Equal* }

Status = *DeadlockFatal*, Enqueued = *No*, Owner = *Caller*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, DeadlockResult = *Fatal*, Recursive = *Deadlock*, Owner = *Caller*, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = *Fatal*, Recursive = *Deadlock*, Owner = *Caller*, Priority = NA
- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, DeadlockResult = *Fatal*, Recursive = *Deadlock*, Owner = *Caller*, Priority = { *High*, *Equal* }

Status = *Ok*, Enqueued = *No*, Owner = *Caller*, Priority = *Ceiling*

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *None*, Priority = { *Equal*, *Low* }

Status = *DeadlockStatus*, Enqueued = *No*, Owner = *Other*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, DeadlockResult = *Status*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = *Status*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = NA
- Protocol = *Ceiling*, Discipline = *Priority*, DeadlockResult = *Status*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = { *High*, *Equal*, *Low* }
- Protocol = *MrsP*, Discipline = *Priority*, DeadlockResult = *Status*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = { *Equal*, *Low* }

Status = *DeadlockFatal*, Enqueued = *No*, Owner = *Other*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, DeadlockResult = *Fatal*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = *Fatal*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = NA
- Protocol = *Ceiling*, Discipline = *Priority*, DeadlockResult = *Fatal*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = { *High*, *Equal*, *Low* }
- Protocol = *MrsP*, Discipline = *Priority*, DeadlockResult = *Fatal*, Recursive = { *Allowed*, *Deadlock* }, Owner = *Deadlock*, Priority = { *Equal*, *Low* }

Status = *Ok*, Enqueued = *No*, Owner = *Caller*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, DeadlockResult = { *Status*, *Fatal* }, Recursive = *Allowed*, Owner = { *None*, *Caller* }, Priority = NA
- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, DeadlockResult = { *Status*, *Fatal* }, Recursive = *Deadlock*, Owner = *None*, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = *Allowed*, Owner = { *None*, *Caller* }, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = *Deadlock*, Owner = *None*, Priority = NA
- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = *Allowed*, Owner = *Caller*, Priority = { *High*, *Equal* }

CeilingOwner

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = *Caller*, Priority = *Low*

PriorityDisciplineByProtocol

- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *FIFO*, DeadlockResult = { *Status*, *Fatal* }, Recursive = { *Allowed*, *Deadlock* }, Owner = { *None*, *Caller*, *Other*, *Deadlock* }, Priority = { *High*, *Equal*, *Low* }

5.1.367 spec:/score mtx/req/surrender

spec:/score mtx/req/surrender

When the calling thread surrenders the mutex.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- *spec:/score/tq/req/surrender*
- *spec:/score/tq/req/surrender-mrsp*
- *spec:/score/tq/req/surrender-priority-inherit*

This requirement refines *spec:/score mtx/if/group*.

5.1.367.1 pre-conditions

Protocol

None Where the mutex does not use a locking protocol.

Inherit Where the mutex uses the priority inheritance locking protocol.

Ceiling Where the mutex uses the priority ceiling locking protocol.

MrsP Where the mutex uses the MrsP locking protocol.

Discipline

FIFO Where the thread queue of the mutex uses the FIFO discipline.

Priority Where the thread queue of the mutex uses the priority discipline.

Recursive

Allowed Where a recursive seize of the mutex is allowed.

NotAllowed Where a recursive seize of the mutex is not allowed.

OwnerCheck

Yes Where the surrender checks that the mutex owner is the calling thread.

No Where the surrender does not check that the mutex owner is the calling thread.

Owner

None While the mutex has no owner.

Caller While the owner of the mutex is the calling thread.

Other While the owner of the mutex is a thread other than the calling thread.

Nested

Yes While calling thread seized the mutex recursively.

No While calling thread seized the mutex not recursively.

Blocked

Yes While the mutex has threads blocked on the mutex.

No While no threads are blocked on the mutex.

Priority

High While the current priority of the calling thread without the priorities available through the mutex would be higher than the highest priority of the priorities available through the mutex.

Equal While the current priority of the calling thread without the priorities available through the mutex would be equal to the highest priority of the priorities available through the mutex.

Low While the current priority of the calling thread without the priorities available through the mutex would be lower than the highest priority of the priorities available through the mutex.

5.1.367.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

NotOwner The return status of the directive call shall be derived from STATUS_NOT_OWNER.

Owner

None The mutex shall have no owner.

Caller The owner of the mutex shall be the calling thread.

Other The owner of the mutex shall not be modified.

First The owner of the mutex shall be dequeued thread.

Surrender

Nop The thread queue of the mutex shall not be surrendered to a thread.

FIFO The thread queue of the mutex shall be surrendered in FIFO order.

Priority The thread queue of the mutex shall be surrendered in priority order.

PriorityInherit The thread queue of the mutex shall be surrendered in priority order with priority inheritance.

MrsP The thread queue of the mutex shall be surrendered in priority order with MrsP.

Priority

Nop The current priority of the calling thread shall be not be modified.

Low The current priority of the calling thread shall be lowered to reflect the removal of the priorities available through the mutex.

5.1.367.3 skip-reasons

BlockedNeedsOwner Blocked threads only exist, if the mutex has an owner.

NestedNeedsRecursive Where the mutex does not allow a recursive seize, the mutex cannot be recursively seized.

MustBeOwner Where the mutex does not check that the mutex owner is the calling thread, the mutex owner must be the calling thread, otherwise the system behaviour is undefined.

PriorityDisciplineByProtocol The locking protocol require a priority discipline.

5.1.367.4 transition-map

Status = *Ok*, Owner = *First*, Surrender = *PriorityInherit*, Priority = *Low*

- Protocol = *Inherit*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = *Low*

Status = *Ok*, Owner = *First*, Surrender = *Priority*, Priority = *Low*

- Protocol = *Ceiling*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = *Low*

Status = *Ok*, Owner = *First*, Surrender = *MrsP*, Priority = *Low*

- Protocol = *MrsP*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = *Low*

Status = *Ok*, Owner = *First*, Surrender = *PriorityInherit*, Priority = *Nop*

- Protocol = *Inherit*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = { *High*, *Equal* }

Status = *Ok*, Owner = *None*, Surrender = *Nop*, Priority = *Low*

- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *No*, Priority = *Low*

Status = *Ok*, Owner = *First*, Surrender = *MrsP*, Priority = *Nop*

- Protocol = *MrsP*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = { *High*, *Equal* }

Status = *Ok*, Owner = *First*, Surrender = *FIFO*, Priority = *Nop*

- Protocol = *None*, Discipline = *FIFO*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = NA

Status = *Ok*, Owner = *First*, Surrender = *Priority*, Priority = *Nop*

- Protocol = *None*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = NA
- Protocol = *Ceiling*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *Yes*, Priority = { *High*, *Equal* }

Status = *NotOwner*, Owner = *None*, Surrender = *Nop*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = *Yes*, Owner = *None*, Nested = NA, Blocked = *No*, Priority = NA
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = *Yes*, Owner = *None*, Nested = NA, Blocked = *No*, Priority = NA

Status = *Ok*, Owner = *None*, Surrender = *Nop*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *No*, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *No*, Priority = NA
- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *No*, Blocked = *No*, Priority = { *High*, *Equal* }

Status = *Ok*, Owner = *Caller*, Surrender = *Nop*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = *Allowed*, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *Yes*, Blocked = { *Yes*, *No* }, Priority = NA
- Protocol = *Inherit*, Discipline = *Priority*, Recursive = *Allowed*, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *Yes*, Blocked = *Yes*, Priority = { *High*, *Equal*, *Low* }
- Protocol = *Inherit*, Discipline = *Priority*, Recursive = *Allowed*, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *Yes*, Blocked = *No*, Priority = NA
- Protocol = { *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = *Allowed*, OwnerCheck = { *Yes*, *No* }, Owner = *Caller*, Nested = *Yes*, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }

Status = *NotOwner*, Owner = *Other*, Surrender = *Nop*, Priority = *Nop*

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = *Allowed*, OwnerCheck = *Yes*, Owner = *Other*, Nested = NA, Blocked = { *Yes*, *No* }, Priority = NA
- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = *NotAllowed*, OwnerCheck = *Yes*, Owner = *Other*, Nested = NA, Blocked = { *Yes*, *No* }, Priority = NA
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = *Allowed*, OwnerCheck = *Yes*, Owner = *Other*, Nested = NA, Blocked = { *Yes*, *No* }, Priority = NA
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = *NotAllowed*, OwnerCheck = *Yes*, Owner = *Other*, Nested = NA, Blocked = { *Yes*, *No* }, Priority = NA

BlockedNeedsOwner

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = *Yes*, Owner = *None*, Nested = NA, Blocked = *Yes*, Priority = NA
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = *Yes*, Owner = *None*, Nested = NA, Blocked = *Yes*, Priority = NA

NestedNeedsRecursive

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = *NotAllowed*, OwnerCheck = { *Yes*, *No* }, Owner = { *None*, *Caller*, *Other* }, Nested = *Yes*, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = *NotAllowed*, OwnerCheck = { *Yes*, *No* }, Owner = { *None*, *Caller*, *Other* }, Nested = *Yes*, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }

MustBeOwner

- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = *Allowed*, OwnerCheck = *No*, Owner = { *None*, *Other* }, Nested = { *Yes*, *No* }, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }
- Protocol = *None*, Discipline = { *FIFO*, *Priority* }, Recursive = *NotAllowed*, OwnerCheck = *No*, Owner = { *None*, *Other* }, Nested = *No*, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = *Allowed*, OwnerCheck = *No*, Owner = { *None*, *Other* }, Nested = { *Yes*, *No* }, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }
- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *Priority*, Recursive = *NotAllowed*, OwnerCheck = *No*, Owner = { *None*, *Other* }, Nested = *No*, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }

PriorityDisciplineByProtocol

- Protocol = { *Inherit*, *Ceiling*, *MrsP* }, Discipline = *FIFO*, Recursive = { *Allowed*, *NotAllowed* }, OwnerCheck = { *Yes*, *No* }, Owner = { *None*, *Caller*, *Other* }, Nested = { *Yes*, *No* }, Blocked = { *Yes*, *No* }, Priority = { *High*, *Equal*, *Low* }

5.1.368 spec:/score/object/req/allocator-mutex

spec:/score/object/req/allocator-mutex

The object services shall use a mutex to protect the object allocation and deallocation.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/object/if/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/userext/req/thread-create-allocator-owner*
- *spec:/rtems/userext/req/thread-delete-allocator-owner*
- *spec:/score/object/req/fatal-allocator-mutex-deadlock*

5.1.369 spec:/score/object/req/fatal-allocator-mutex-deadlock

spec:/score/object/req/fatal-allocator-mutex-deadlock

If obtaining the object allocator mutex would result in a deadlock, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_THREAD_QUEUE_DEADLOCK fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/object/req/allocator-mutex*.

5.1.370 spec:/score/sched/req/yield

spec:/score/sched/req/yield

When the thread yields.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/sched/req/group*.

5.1.370.1 pre-conditions

EligibleScheduler

Home While the only eligible scheduler of the thread is the home scheduler.

Helping While the thread has at least one helping scheduler.

UsedScheduler

Home While the thread is scheduled on the home scheduler.

Helping While the thread is scheduled on a helping scheduler.

HomeSchedulerState

Blocked The thread shall be blocked in its home scheduler.

Scheduled The thread shall be scheduled in its home scheduler.

Ready The thread shall be ready in its home scheduler.

Sticky

Yes While the thread is sticky.

No While the thread not sticky.

Other

Yes While at least one ready thread with a priority equal to the priority of the thread exists in the home scheduler of the thread.

No While no ready thread with a priority equal to the priority of the thread exists in the home scheduler of the thread.

5.1.370.2 post-conditions

HomeSchedulerState

Blocked The thread shall be blocked in its home scheduler.

Scheduled The thread shall be scheduled in its home scheduler.

Ready The thread shall be ready in its home scheduler.

Idle An idle thread shall execute on behalf of the thread in its home scheduler.

AskForHelp

Yes The thread shall ask all its eligible scheduler for help.

No The thread shall not ask for help.

5.1.370.3 skip-reasons

HelpingNeedsHelping In order to use a helping scheduler a thread needs a helping scheduler.

HelpingNeedsSMP A helping scheduler is only available where the system was built with SMP support enabled.

StickyNeedsSMP A thread may be sticky only where the system was built with SMP support enabled.

Sticky Sticky threads are never blocked in their home scheduler. Only sticky threads may be scheduled on two schedulers at the same time.

HomeNeedsScheduled While a thread uses its home scheduler, it is scheduled in the home scheduler.

5.1.370.4 transition-map

HomeSchedulerState = *Idle*, AskForHelp = *No*

- EligibleScheduler = *Helping*, UsedScheduler = *Helping*, HomeSchedulerState = *Scheduled*, Sticky = *Yes*, Other = *No*

HomeSchedulerState = *Ready*, AskForHelp = *Yes*

- EligibleScheduler = *Helping*, UsedScheduler = *Home*, HomeSchedulerState = *Scheduled*, Sticky = { *Yes*, *No* }, Other = *Yes*

HomeSchedulerState = *Blocked*, AskForHelp = *No*

- EligibleScheduler = *Helping*, UsedScheduler = *Helping*, HomeSchedulerState = *Blocked*, Sticky = *No*, Other = { *Yes*, *No* }

HomeSchedulerState = *Scheduled*, AskForHelp = *No*

- EligibleScheduler = { *Home*, *Helping* }, UsedScheduler = *Home*, HomeSchedulerState = *Scheduled*, Sticky = { *Yes*, *No* }, Other = *No*

HomeSchedulerState = *Ready*, AskForHelp = *No*

- EligibleScheduler = *Home*, UsedScheduler = *Home*, HomeSchedulerState = *Scheduled*, Sticky = { *Yes*, *No* }, Other = *Yes*
- EligibleScheduler = *Helping*, UsedScheduler = *Helping*, HomeSchedulerState = { *Scheduled*, *Ready* }, Sticky = *Yes*, Other = *Yes*
- EligibleScheduler = *Helping*, UsedScheduler = *Helping*, HomeSchedulerState = *Ready*, Sticky = *Yes*, Other = *No*

HelpingNeedsHelping

- EligibleScheduler = *Home*, UsedScheduler = *Helping*, HomeSchedulerState = *Blocked*, Sticky = *No*, Other = { *Yes*, *No* }
- EligibleScheduler = *Home*, UsedScheduler = *Helping*, HomeSchedulerState = { *Scheduled*, *Ready* }, Sticky = { *Yes*, *No* }, Other = { *Yes*, *No* }

Sticky

- EligibleScheduler = { *Home*, *Helping* }, UsedScheduler = *Helping*, HomeSchedulerState = *Blocked*, Sticky = *Yes*, Other = { *Yes*, *No* }
- EligibleScheduler = *Helping*, UsedScheduler = *Helping*, HomeSchedulerState = { *Scheduled*, *Ready* }, Sticky = *No*, Other = { *Yes*, *No* }

HomeNeedsScheduled

- EligibleScheduler = { *Home*, *Helping* }, UsedScheduler = *Home*, HomeSchedulerState = { *Blocked*, *Ready* }, Sticky = { *Yes*, *No* }, Other = { *Yes*, *No* }

5.1.371 spec:/score/sched/smp/edf/req/set-affinity

spec:/score/sched/smp/edf/req/set-affinity

When the thread to processor affinity is set for task T.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/sched/smp/edf/req/group*.

5.1.371.1 pre-conditions

Before

- All** While task T is affine to all processors of its home scheduler before the new thread to processor affinity is set.
- X** While task T is affine to processor X before the new thread to processor affinity is set.

After

- All** While task T is set to be affine to all processors of its home scheduler.
- X** While task T is set to be affine to processor X.
- Y** While task T is set to be affine to processor Y.

Priority

High While task T has a high priority.

Low While task T has a low priority.

State

Ready While task T is ready.

Blocked While task T is blocked.

Sticky

Yes While task T is sticky.

No While task T is not sticky.

Pinned

Yes While task T is pinned to a processor.

No While task T is not pinned to a processor.

AlphaPriority

High While task A has a high priority.

Low While task A has a low priority.

AlphaAffinity

All While task A is affine to all processors of its home scheduler.

X While task A is affine to processor X.

AlphaIdle

Yes While task A is sticky, while task A is blocked.

No While task A is not sticky.

BetaPriority

High While task B has a high priority.

Low While task B has a low priority.

BetaAffinity

All While task B is affine to all processors of its home scheduler.

Y While task B is affine to processor Y.

BetaIdle

Yes While task B is sticky, while task B is blocked.

No While task B is not sticky,

5.1.371.2 post-conditions

X

Task The task T shall be scheduled on processor X.

TaskIdle An idle task on behalf of task T shall be scheduled on processor X.

Alpha The task A shall be scheduled on processor X.

AlphaIdle An idle task on behalf of task A shall be scheduled on processor X.

Beta The task B shall be scheduled on processor X.

BetaIdle An idle task on behalf of task B shall be scheduled on processor X.

Y

Task The task T shall be scheduled on processor Y.

TaskIdle An idle task on behalf of task T shall be scheduled on processor Y.

Alpha The task A shall be scheduled on processor Y.

AlphaIdle An idle task on behalf of task A shall be scheduled on processor Y.

Beta The task B shall be scheduled on processor Y.

BetaIdle An idle task on behalf of task B shall be scheduled on processor Y.

5.1.371.3 skip-reasons

NoStickyAndPinned Thread pinning while the thread owns a sticky mutex is undefined behaviour.

5.1.371.4 transition-map

X = *BetaIdle*, Y = *TaskIdle*

- Before = { *All*, X }, After = Y, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, X }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*

X = *Beta*, Y = *TaskIdle*

- Before = { *All*, X }, After = Y, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, X }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*

X = *BetaIdle*, Y = *Task*

- Before = { *All*, X }, After = Y, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, X }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*

X = *Beta*, Y = *Task*

- Before = { *All*, X }, After = Y, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, X }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*

X = *TaskIdle*, Y = *AlphaIdle*

- Before = { *All*, X }, After = All, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, X }, After = X, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, Y }, BetaIdle = { *Yes*, *No* }

X = *TaskIdle*, Y = *Alpha*

- Before = { *All*, X }, After = All, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, X }, After = X, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, Y }, BetaIdle = { *Yes*, *No* }

X = *TaskIdle*, Y = *BetaIdle*

- Before = { *All*, X }, After = All, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, X }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, Y }, BetaIdle = *Yes*

- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*

X = *TaskIdle*, Y = *Beta*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

X = *AlphaIdle*, Y = *TaskIdle*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

X = *Alpha*, Y = *TaskIdle*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

X = *BetaIdle*, Y = *AlphaIdle*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *Yes*

X = *Beta*, Y = *AlphaIdle*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *No*

X = *BetaIdle*, Y = *Alpha*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *Yes*

X = *Beta*, Y = *Alpha*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *All*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = *No*

X = *AlphaIdle*, Y = *Task*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

X = *Alpha*, Y = *Task*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

X = *Task*, Y = *AlphaIdle*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

X = *Task*, Y = *Alpha*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *All*, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

X = *Task*, Y = *BetaIdle*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*

X = *Task*, *Y* = *Beta*

- Before = { *All*, *X* }, After = *All*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = { *Yes*, *No* }, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

X = *AlphaIdle*, *Y* = *BetaIdle*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*

- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *All*, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *Yes*

- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *X*, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *Y*, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*

X = *AlphaIdle*, *Y* = *Beta*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *All*, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *Yes*,

BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *Yes*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *Yes*, BetaPriority = *High*,

BetaAffinity = *Y*, BetaIdle = *No*

- Before = { *All*, *X* }, After = *Y*, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *Yes*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

X = *Alpha*, *Y* = *BetaIdle*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *Yes*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *X*, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *Yes*
- Before = { *All*, *X* }, After = *Y*, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *Yes*

X = *Alpha*, Y = *Beta*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *Y* }, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*

High, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *All*, Priority = *Low*, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *Yes*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = { *Yes*, *No* }, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *No*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *All*, AlphaIdle = *No*, BetaPriority = *Low*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *X*, Priority = { *High*, *Low* }, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Ready*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }

{ *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

- Before = { *All*, *X* }, After = { *X*, *Y* }, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Ready*, Sticky = { *Yes*, *No* }, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *Y*, Priority = *High*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *Low*, AlphaAffinity = { *All*, *X* }, AlphaIdle = *No*, BetaPriority = *High*, BetaAffinity = *Y*, BetaIdle = *No*
- Before = { *All*, *X* }, After = *Y*, Priority = *Low*, State = *Blocked*, Sticky = *Yes*, Pinned = *No*, AlphaPriority = *High*, AlphaAffinity = *X*, AlphaIdle = *No*, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = *No*

NoStickyAndPinned

- Before = { *All*, *X* }, After = { *All*, *X*, *Y* }, Priority = { *High*, *Low* }, State = { *Ready*, *Blocked* }, Sticky = *Yes*, Pinned = *Yes*, AlphaPriority = { *High*, *Low* }, AlphaAffinity = { *All*, *X* }, AlphaIdle = { *Yes*, *No* }, BetaPriority = { *High*, *Low* }, BetaAffinity = { *All*, *Y* }, BetaIdle = { *Yes*, *No* }

5.1.372 spec:/score/sched/smp/edf/req/set-affinity-unsupported-subset

spec:/score/sched/smp/edf/req/set-affinity-unsupported-subset

While a thread to processor affinity set is a proper subset of the set of the online processors, while the affinity set contains at least two processors, while at least one processor of the affinity set is owned by a scheduler, while a thread uses the scheduler as its home scheduler, when an attempt is made to set the thread to processor affinity set of the thread to the affinity set, the operation shall abort with an error status derived from STATUS_INVALID_NUMBER.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/edf/req/group*.

5.1.373 spec:/score/sched/smp/req/ask-for-help-helping

spec:/score/sched/smp/req/ask-for-help-helping

While a thread is registered for help in a helping scheduler, while the thread is scheduled by another scheduler or blocked, when the helping scheduler tries to schedule the thread, the thread shall be blocked with respect to the helping scheduler.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/req/group*.

5.1.374 spec:/score/sched/smp/req/ask-for-help-home

spec:/score/sched/smp/req/ask-for-help-home

While a thread is registered for help in its home scheduler, while the thread is scheduled by another scheduler, while the thread is not sticky, when the home scheduler tries to schedule the thread, the thread shall be blocked with respect to the home scheduler.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/req/group*.

5.1.375 spec:/score/sched/smp/req/ask-for-help-request

spec:/score/sched/smp/req/ask-for-help-request

When a need for help is detected for a thread, the ask for help request shall be registered on the current processor.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/req/group*.

5.1.376 spec:/score/sched/smp/req/block-reuse-idle

spec:/score/sched/smp/req/block-reuse-idle

While a thread is scheduled by the home scheduler, while an idle thread executes on behalf of the thread in the home scheduler, when the thread is blocked, the idle thread executing on behalf of the thread shall continue to execute on behalf of the thread.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/req/group*.

5.1.377 spec:/score/sched/smp/req/idle-to-scheduled

spec:/score/sched/smp/req/idle-to-scheduled

While an idle thread is scheduled on behalf of a thread by the home scheduler, while the thread is not scheduled by another scheduler, when the thread is rescheduled by the home scheduler, the thread shall be scheduled.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/req/group*.

5.1.378 spec:/score/sched/smp/req/preempt-blocked

spec:/score/sched/smp/req/preempt-blocked

While a thread is scheduled by a helping scheduler, while the thread is blocked, when the thread is preempted by the helping scheduler, the thread shall be blocked by the helping scheduler.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/sched/smp/req/group*.

5.1.379 spec:/score/sem/req/seize-try

spec:/score/sem/req/seize-try

When the calling task tries to seize the semaphore.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/sem/if/group*.

5.1.379.1 pre-conditions

Count

Zero While the count of the semaphore is zero.

Positive While the count of the semaphore is greater than zero.

5.1.379.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

Unsat The return status of the directive call shall be derived from STATUS_UNSATISFIED.

Count

Nop The count of the semaphore shall not be modified.

MinusOne The count of the semaphore shall be decremented by one.

5.1.379.3 transition-map

Status = *Unsat*, Count = *Nop*

- Count = *Zero*

Status = *Ok*, Count = *MinusOne*

- Count = *Positive*

5.1.380 spec:/score/sem/req/seize-wait

spec:/score/sem/req/seize-wait

When the calling task tries to seize the semaphore.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- *spec:/score/tq/req/enqueue-fifo*
- *spec:/score/tq/req/enqueue-priority*

This requirement refines *spec:/score/sem/if/group*.

5.1.380.1 pre-conditions

Count

Zero While the count of the semaphore is zero.

Positive While the count of the semaphore is greater than zero.

5.1.380.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

Enqueued Where the thread queue uses the FIFO discipline, the calling thread shall be enqueued in FIFO order.

Where the thread queue uses the priority discipline, the calling thread shall be enqueued in priority order.

Count

Nop The count of the semaphore shall not be modified.

MinusOne The count of the semaphore shall be decremented by one.

Timer

Optional Where the directive was called with a timeout in clock ticks, the thread timer of the calling task shall fire after the specified clock ticks.

Where the directive was called without a timeout, the thread timer of the calling task shall be inactive.

No The thread timer of the calling task shall be inactive.

5.1.380.3 transition-map

Status = *Enqueued*, Count = *Nop*, Timer = *Optional*

- Count = *Zero*

Status = *Ok*, Count = *MinusOne*, Timer = *No*

- Count = *Positive*

5.1.381 spec:/score/sem/req/surrender

spec:/score/sem/req/surrender

When the calling surrenders the semaphore.

rationale: N/A

functional-type: action

Functions of this item are implemented by:

- *spec:/score/tq/req/surrender*

This requirement refines *spec:/score/sem/if/group*.

5.1.381.1 pre-conditions

Variant

Binary Where the semaphore is a binary semaphore.

Counting Where the semaphore is a counting semaphore.

Discipline

FIFO Where the thread queue of the semaphore uses the FIFO discipline.

Priority Where the thread queue of the semaphore uses the priority discipline.

Count

LessMax While the count of the semaphore is less than the maximum count.

Max While the count of the semaphore is equal to the maximum count.

Blocked While the semaphore has threads blocked on the semaphore.

5.1.381.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

MaxCountExceeded The return status of the directive call shall be derived from STATUS_MAXIMUM_COUNT_EXCEEDED.

Surrender

FIFO The thread queue of the semaphore shall be surrendered in FIFO order.

Priority The thread queue of the semaphore shall be surrendered in priority order.

Count

Zero The count of the semaphore shall be zero.

One The count of the semaphore shall be one.

PlusOne The count of the semaphore shall be incremented by one.

Nop The count of the semaphore shall not be modified.

5.1.381.3 transition-map

Status = *Ok*, Surrender = *FIFO*, Count = *Zero*

- Variant = { *Binary*, *Counting* }, Discipline = *FIFO*, Count = *Blocked*

Status = *Ok*, Surrender = *Priority*, Count = *Zero*

- Variant = { *Binary*, *Counting* }, Discipline = *Priority*, Count = *Blocked*

Status = *Ok*, Surrender = NA, Count = *PlusOne*

- Variant = *Counting*, Discipline = { *FIFO*, *Priority* }, Count = *LessMax*

Status = *MaxCountExceeded*, Surrender = NA, Count = *Nop*

- Variant = *Counting*, Discipline = { *FIFO*, *Priority* }, Count = *Max*

Status = *Ok*, Surrender = NA, Count = *One*

- Variant = *Binary*, Discipline = { *FIFO*, *Priority* }, Count = { *LessMax*, *Max* }

5.1.382 spec:/score/smp/req/fatal-boot-processor-not-assigned-to-scheduler

spec:/score/smp/req/fatal-boot-processor-not-assigned-to-scheduler

While the system is initialized, if the boot processor is not assigned to a scheduler, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_BOOT_PROCESSOR_NOT_ASSIGNED_TO_SCHEDULER fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/smp/if/group*.

5.1.383 spec:/score/smp/req/fatal-mandatory-processor-not-present

spec:/score/smp/req/fatal-mandatory-processor-not-present

While the system is initialized, if a mandatory processor defined by the application configuration is not present, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_MANDATORY_PROCESSOR_NOT_PRESENT fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/smp/if/group*.

5.1.384 spec:/score/smp/req/fatal-multitasking-start-on-invalid-processor

spec:/score/smp/req/fatal-multitasking-start-on-invalid-processor

While the system is initialized, if multitasking is started on a processor those index is greater than or equal to the processor maximum defined by the application configuration, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_MULTITASKING_START_ON_INVALID_PROCESSOR fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/smp/if/group*.

5.1.385 spec:/score/smp/req/fatal-multitasking-start-on-unassigned-processor

spec:/score/smp/req/fatal-multitasking-start-on-unassigned-processor

While the system is initialized, if multitasking is started on a processor which has no scheduler assigned by the application configuration, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_MULTITASKING_START_ON_UNASSIGNED_PROCESSOR fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines [spec:/score/smp/if/group](#).

5.1.386 spec:/score/smp/req/fatal-scheduler-requires-exactly-one-processor

spec:/score/smp/req/fatal-scheduler-requires-exactly-one-processor

Where the configured scheduler does support exactly one processor, where the system has not exactly one processor, if the system is initialized, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_SCHEDULERQUIRES_EXACTLY_ONE_PROCESSOR fatal code.

rationale: N/A

functional-type: function

This requirement refines [spec:/req/fatal-error](#).

This requirement refines [spec:/score/smp/if/group](#).

5.1.387 spec:/score/smp/req/fatal-shutdown-response

spec:/score/smp/req/fatal-shutdown-response

If the processor receives an SMP shutdown request message, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_SHUTDOWN_RESPONSE fatal code.

rationale: N/A

functional-type: function

This requirement refines [spec:/req/fatal-error](#).

This requirement refines [spec:/score/smp/if/group](#).

5.1.388 spec:/score/smp/req/fatal-start-of-mandatory-processor-failed

spec:/score/smp/req/fatal-start-of-mandatory-processor-failed

While the system is initialized, if the start of a mandatory processor defined by the application configuration fails, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_START_OF_MANDATORY_PROCESSOR_FAILED fatal code.

rationale: N/A

functional-type: function

This requirement refines [spec:/req/fatal-error](#).

This requirement refines [spec:/score/smp/if/group](#).

5.1.389 spec:/score/smp/req/fatal-start-on-not-online-processor

spec:/score/smp/req/fatal-start-on-not-online-processor

If multitasking is started on a not online processor, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_START_ON_NOT_ONLINE_PROCESSOR fatal code.

rationale: N/A

functional-type: function

This requirement refines [spec:/req/fatal-error](#).

This requirement refines [spec:/score/smp/if/group](#).

5.1.390 spec:/score/smp/req/fatal-wrong-cpu-state-to-perform-jobs

spec:/score/smp/req/fatal-wrong-cpu-state-to-perform-jobs

If the target processor of a job is not in the initial, ready to start multiprocessing, or up state, then the system shall terminate with the RTEMS_FATAL_SOURCE_SMP fatal source and the SMP_FATAL_START_OF_MANDATORY_PROCESSOR_FAILED fatal code.

rationale: N/A

functional-type: function

This requirement refines [spec:/req/fatal-error](#).

This requirement refines [spec:/score/smp/if/group](#).

5.1.391 spec:/score/smp/req/per-cpu-jobs-order

spec:/score/smp/req/per-cpu-jobs-order

Per-processor jobs shall be processed in FIFO order.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/smp/if/group](#).

5.1.392 spec:/score/smp/req/processor-online

spec:/score/smp/req/processor-online

Processors which were successfully started shall be marked as online.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/smp/if/group*.

5.1.393 spec:/score/smp/req/start-of-optional-processor-failed-continue

spec:/score/smp/req/start-of-optional-processor-failed-continue

While the system is initialized, if the start of an optional processor defined by the application configuration fails, then the system initialization shall continue.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/smp/if/group*.

5.1.394 spec:/score/smp/req/start-of-processor-failed-not-online

spec:/score/smp/req/start-of-processor-failed-not-online

Processors which failed to start shall not be marked as online.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/smp/if/group*.

5.1.395 spec:/score/thread/req/cancel-killer

spec:/score/thread/req/cancel-killer

While a terminating thread has exactly one joining thread, while the joining thread can be cancelled, if the terminating thread cancels the joining thread, then the terminating thread shall wait for threads to join.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/if/group*.

5.1.396 spec:/score/thread/req/fatal-bad-thread-dispatch-disable-level

spec:/score/thread/req/fatal-bad-thread-dispatch-disable-level

While the thread dispatch disable level is not equal to one, if a direct thread dispatch is done, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_BAD_THREAD_DISPATCH_DISABLE_LEVEL fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/thread/if/group*.

5.1.397 spec:/score/thread/req/fatal-bad-thread-dispatch-environment-ipi

spec:/score/thread/req/fatal-bad-thread-dispatch-environment-ipi

Where the system needs inter-processor interrupts, if a thread dispatch is done with maskable interrupts disabled, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_BAD_THREAD_DISPATCH_ENVIRONMENT fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/thread/if/group*.

5.1.398 spec:/score/thread/req/fatal-thread-exitted

spec:/score/thread/req/fatal-thread-exitted

If a thread body function returned and the thread exitted extension invocation for the thread returned, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_THREAD_EXITTED fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/thread/if/group*.

5.1.399 spec:/score/thread/req/pinning-helping

spec:/score/thread/req/pinning-helping

While a thread is pinned, while the thread executes on a processor which is not owned by its home scheduler, the thread shall only use the scheduler of the processor to which it is pinned.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/req/pinning*.

5.1.400 spec:/score/thread/req/pinning-nested

spec:/score/thread/req/pinning-nested

A thread may be pinned more than once at a time.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/req/pinning*.

5.1.401 spec:/score/thread/req/pinning-preemptible

spec:/score/thread/req/pinning-preemptible

While a thread is pinned, the thread may be preempted by a scheduler.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/req/pinning*.

5.1.402 spec:/score/thread/req/pinning-unpin-suspended

spec:/score/thread/req/pinning-unpin-suspended

While a thread is suspended, the thread may be unpinned.

rationale: Unpinning the thread is done with thread dispatching disabled. Other processors or interrupts may suspend the thread while it runs with thread dispatching disabled.

functional-type: function

This requirement refines *spec:/score/thread/req/pinning*.

5.1.403 spec:/score/thread/req/suspended-helping

spec:/score/thread/req/suspended-helping

While a thread is suspended, the thread shall not reconsider help requests.

rationale: This is an performance optimization.

functional-type: function

This requirement refines *spec:/score/thread/if/group*.

5.1.404 spec:/score/thread/req/thread-dispatch-isr-disabled

spec:/score/thread/req/thread-dispatch-isr-disabled

Where the system does not need inter-processor interrupts, when a thread dispatch is done with maskable interrupts disabled, the thread dispatch shall be carried out.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/if/group*.

5.1.405 spec:/score/thread/req/tls-big-alignment

spec:/score/thread/req/tls-big-alignment

The alignment of thread-local storage objects may be greater than CPU_STACK_ALIGNMENT.

rationale: N/A

functional-type: capability

5.1.406 spec:/score/thread/req/tls-initialized-data

spec:/score/thread/req/tls-initialized-data

While the thread entry execution begins, the value of non-zero initialized thread-local storage objects shall be the initialization value.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/if/group*.

5.1.407 spec:/score/thread/req/tls-max-zero

spec:/score/thread/req/tls-max-zero

Where the configured maximum thread-local storage size is equal to zero, the thread-local storage allocation size shall be defined by the thread-local storage size of the application executable.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/if/group*.

5.1.408 spec:/score/thread/req/tls-zero-initialized-data

spec:/score/thread/req/tls-zero-initialized-data

While the thread entry execution begins, the storage space of zero initialized thread-local storage objects shall be cleared to zero.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/thread/if/group*.

5.1.409 spec:/score/timecounter/req/clock-monotonic-initial

spec:/score/timecounter/req/clock-monotonic-initial

The initial value of the CLOCK_MONOTONIC shall be 0000-00-00T00:00:01.0000000000.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/clock-monotonic*.

5.1.410 spec:/score/timecounter/req/clock-monotonic-initial-frozen

spec:/score/timecounter/req/clock-monotonic-initial-frozen

While no timecounter is installed, the CLOCK_MONOTONIC shall remain at the initial value.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/clock-monotonic*.

5.1.411 spec:/score/timecounter/req/clock-realtime-initial

spec:/score/timecounter/req/clock-realtime-initial

The initial value of the CLOCK_REALTIME shall be 1988-01-01T00:00:00.0000000000.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/clock-realtime*.

5.1.412 spec:/score/timecounter/req/clock-realtime-initial-frozen

spec:/score/timecounter/req/clock-realtime-initial-frozen

While no timecounter is installed, the CLOCK_REALTIME shall remain at the initial value.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/clock-realtime*.

5.1.413 spec:/score/timecounter/req/get-coarse-no-device

spec:/score/timecounter/req/get-coarse-no-device

The directives to get time values in a coarse resolution shall not use the currently installed timecounter.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/get-coarse*.

5.1.414 spec:/score/timecounter/req/get-device

spec:/score/timecounter/req/get-device

The directives to get time values in the highest resolution shall use the currently installed timecounter to get a clock snapshot.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/get*.

5.1.415 spec:/score/timecounter/req/get-large-delta

spec:/score/timecounter/req/get-large-delta

While the timehand used to get the time was not updated for one second or more, while the timecounter of the timehand did not overflow, the directives to get time values in the highest resolution shall return the correct time.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/get*.

5.1.416 spec:/score/timecounter/req/install-frequency

spec:/score/timecounter/req/install-frequency

While the timecounter to install has a quality level equal to the quality level of the currently installed timecounter, while the timecounter to install has a higher frequency greater than the frequency of the currently installed timecounter, the timecounter to install shall be installed.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/install*.

5.1.417 spec:/score/timecounter/req/install-quality

spec:/score/timecounter/req/install-quality

While the timecounter to install has a quality level greater than the quality level of the currently installed timecounter, the timecounter to install shall be installed.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/timecounter/req/install*.

5.1.418 spec:/score/timecounter/req/ntp-adjustment

spec:/score/timecounter/req/ntp-adjustment

While the timehand is updated, when the NTP handler is called, the value returned by the adjustment parameter shall be used to adjust the timecounter frequency of the timehand.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/ntp](#).

5.1.419 spec:/score/timecounter/req/ntp-seconds

spec:/score/timecounter/req/ntp-seconds

While the timehand is updated, when the NTP handler is called, the value returned by the newsec parameter shall define the seconds value of the CLOCK_REALTIME time point of the timehand.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/ntp](#).

5.1.420 spec:/score/timecounter/req/ntp-step

spec:/score/timecounter/req/ntp-step

Let T be the seconds value of the CLOCK_REALTIME time point of the timehand which is currently updated, let S be the seconds value of the CLOCK_REALTIME time point of the previous timehand, let D be T minus S, while the NTP handler is not NULL, while D is greater than or equal to one, while D is less than or equal to 200, while the timehand is updated, the NTP handler shall be called exactly D times.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/ntp](#).

5.1.421 spec:/score/timecounter/req/ntp-step-large

spec:/score/timecounter/req/ntp-step-large

Let T be the seconds value of the CLOCK_REALTIME time point of the timehand which is currently updated, let S be the seconds value of the CLOCK_REALTIME time point of the previous timehand, let D be T minus S, while the NTP handler is not NULL, while D is greater than 200, while the timehand is updated, the NTP handler shall be called exactly two times.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/ntp](#).

5.1.422 spec:/score/timecounter/req/tick-large-delta

spec:/score/timecounter/req/tick-large-delta

While the oldest timehand was not updated for one second or more, while the time interval from the time point of the timehand to the time point of the update is measureable with the timecounter of the timehand, the time point of the timehand shall be incremented by the time interval.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/group](#).

5.1.423 spec:/score/timecounter/req/tick-simple-delta

spec:/score/timecounter/req/tick-simple-delta

When the simple timecounter tick service is called, the time points of the current timehand shall be advanced by the time interval specified by the delta parameter.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/tick-simple](#).

5.1.424 spec:/score/timecounter/req/tick-simple-offset

spec:/score/timecounter/req/tick-simple-offset

When the simple timecounter tick service is called, the timecounter offset of the current timehand shall be set to the offset specified by the offset parameter.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/timecounter/req/tick-simple](#).

5.1.425 spec:/score/tq/req/deadlock-concurrent

spec:/score/tq/req/deadlock-concurrent

While a thread A tries to enqueue on a thread queue X, while a thread B tries to enqueue on a thread queue Y, while the thread queue dependency graph built up by threads A and B contains a cycle, the deadlock shall be detected.

rationale: N/A

functional-type: function

This requirement refines [spec:/score/tq/req/deadlock](#).

5.1.426 spec:/score/tq/req/enqueue-ceiling

spec:/score/tq/req/enqueue-ceiling

When the calling thread is enqueued on the thread queue.

rationale: N/A

functional-type: action

This requirement refines [spec:/score/tq/if/group](#).

5.1.426.1 pre-conditions

EligibleScheduler

Home While the enqueueing thread has no helping scheduler.

Helping While the enqueueing thread has at least one helping scheduler.

QueueEligible

None While all priority queues of the thread queue associated with eligible schedulers of the enqueueing thread are empty.

Equal While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is equal to the priority of the enqueueing thread with respect to the eligible scheduler.

Low While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is lower than the priority of the enqueueing thread with respect to the eligible scheduler.

QueueIneligible

None While no priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread.

Before While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while this priority queue is positioned before all priority queues which are associated with eligible schedulers of the enqueueing thread.

After While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while this priority queue is positioned after all priority queues which are associated with eligible schedulers of the enqueueing thread.

5.1.426.2 post-conditions

Position

InitialFirst A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the first priority queue of the thread queue.

First The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

Second The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

FirstFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondQueue The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

5.1.426.3 skip-reasons

Invalid These variants are invalid due to two independent reasons. Firstly, where the system was built with SMP support disabled, no other scheduler can exist. Secondly, a priority queue must be present to have another priority queue positioned before or after the priority queue.

5.1.426.4 transition-map

Position = *InitialFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = *None*

Position = *Second*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Equal*, QueueIneligible = *None*

Position = *SecondFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Equal*, QueueIneligible = *After*

Position = *First*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Low*, QueueIneligible = *None*

Position = *FirstFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Low*, QueueIneligible = *After*

Position = *SecondQueue*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *Equal*, *Low* }, QueueIneligible = *Before*

Invalid

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = { *Before*, *After* }

5.1.427 spec:/score/tq/req/enqueue-deadlock

spec:/score/tq/req/enqueue-deadlock

When the calling thread attempts to be enqueued on the thread queue.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/req/deadlock*.

5.1.427.1 pre-conditions

Notification

Status Where a detected deadlock results in a return with a status code.

Fatal Where a detected deadlock results in a fatal error.

Deadlock

One While the owner of the thread queue is enqueued on another thread queue owned by the calling thread.

More While the owner of the thread queue is enqueued on another thread queue owned by a thread other than the calling thread, and so on, while the owner of the last thread queue of this dependency chain is enqueued on a thread queue owned by the calling thread.

5.1.427.2 post-conditions

Result

Status The return status of the directive call shall be derived from STATUS_DEADLOCK.

Fatal The system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_THREAD_QUEUE_DEADLOCK fatal code.

5.1.427.3 transition-map

Result = *Status*

- Notification = *Status*, Deadlock = { *One*, *More* }

Result = *Fatal*

- Notification = *Fatal*, Deadlock = { *One*, *More* }

5.1.428 spec:/score/tq/req/enqueue-fifo

spec:/score/tq/req/enqueue-fifo

When the calling thread is enqueued on the thread queue.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.428.1 pre-conditions

Queue

Empty While the queue is empty.

NonEmpty While the queue is non-empty.

5.1.428.2 post-conditions

Position

First The thread shall be the first thread in the queue.

Last The thread shall be the last thread in the queue.

5.1.428.3 transition-map

Position = *First*

- Queue = *Empty*

Position = *Last*

- Queue = *NonEmpty*

5.1.429 spec:/score/tq/req/enqueue-mrsp

spec:/score/tq/req/enqueue-mrsp

When the calling thread is enqueued on the thread queue.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.429.1 pre-conditions

EligibleScheduler

Home While the enqueueing thread has no helping scheduler.

Helping While the enqueueing thread has at least one helping scheduler.

QueueEligible

None While all priority queues of the thread queue associated with eligible schedulers of the enqueueing thread are empty.

Equal While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is equal to the priority of the enqueueing thread with respect to the eligible scheduler.

Low While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is lower than the priority of the enqueueing thread with respect to the eligible scheduler.

QueueIneligible

None While no priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread.

Only While exactly one priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread.

Before While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while this priority queue is positioned before all priority queues which are associated with eligible schedulers of the enqueueing thread.

After While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while this priority queue is positioned after all priority queues which are associated with eligible schedulers of the enqueueing thread.

5.1.429.2 post-conditions

Position

InitialFirst A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the first priority queue of the thread queue.

InitialLast A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the last priority queue of the thread queue.

Second The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

SecondFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondLast The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

5.1.429.3 skip-reasons

Invalid These variants are invalid due to two independent reasons. Firstly, a priority queue must be present to have another priority queue positioned before or after the priority queue. Secondly, if only one priority queue shall be present, then no other priority queue can exist.

5.1.429.4 transition-map

Position = *InitialFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = *None*

Position = *InitialLast*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = *Only*

Position = *Second*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *Equal*, *Low* }, QueueIneligible = *None*

Position = *SecondLast*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *Equal*, *Low* }, QueueIneligible = *Before*

Position = *SecondFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *Equal*, *Low* }, QueueIneligible = *After*

Invalid

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = { *Before*, *After* }
- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *Equal*, *Low* }, QueueIneligible = *Only*

5.1.430 spec:/score/tq/req/enqueue-priority

spec:/score/tq/req/enqueue-priority

When the calling task is enqueued on the thread queue.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.430.1 pre-conditions

EligibleScheduler

Home While the enqueueing thread has no helping scheduler.

Helping While the enqueueing thread has at least one helping scheduler.

QueueEligible

None While all priority queues of the thread queue associated with eligible schedulers of the enqueueing thread are empty.

High While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is higher than the priority of the enqueueing thread with respect to the eligible scheduler.

Equal While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is equal to the priority of the enqueueing thread with respect to the eligible scheduler.

Low While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is lower than the priority of the enqueueing thread with respect to the eligible scheduler.

QueueIneligible

None While no priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread.

Only While exactly one priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread.

Before While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while this priority queue is positioned before all priority queues which are associated with eligible schedulers of the enqueueing thread.

After While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while this priority queue is positioned after all priority queues which are associated with eligible schedulers of the enqueueing thread.

5.1.430.2 post-conditions

Position

InitialFirst A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the first priority queue of the thread queue.

InitialLast A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the last priority queue of the thread queue.

First The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

Second The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

FirstFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

FirstLast The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondLast The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

5.1.430.3 skip-reasons

Invalid These variants are invalid due to three independent reasons. Firstly, where the system was built with SMP support disabled, no other scheduler can exist. Secondly, a priority queue must be present to have another priority queue positioned before or after the priority queue. Thirdly, if only one priority queue shall be present, then on other priority queue can exist.

5.1.430.4 transition-map

Position = *InitialFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = *None*

Position = *InitialLast*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = *Only*

Position = *First*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Low*, QueueIneligible = *None*

Position = *FirstLast*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Low*, QueueIneligible = *Before*

Position = *FirstFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *Low*, QueueIneligible = *After*

Position = *Second*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*

Position = *SecondLast*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*

Position = *SecondFirst*

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*

Invalid

- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = *None*, QueueIneligible = { *Before*, *After* }
- EligibleScheduler = { *Home*, *Helping* }, QueueEligible = { *High*, *Equal*, *Low* }, QueueIneligible = *Only*

5.1.431 spec:/score/tq/req/enqueue-priority-inherit

spec:/score/tq/req/enqueue-priority-inherit

When the calling task is enqueued on the thread queue.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.431.1 pre-conditions

Scheduler

- One** Where the system has exactly one schedulers.
- Two** Where the system has exactly two schedulers.
- Three** Where the system has exactly three schedulers.
- More** Where the system has at least three schedulers.

QueueEligible

- None** While all priority queues of the thread queue associated with eligible schedulers of the enqueueing thread are empty.
- High** While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is higher than the priority of the enqueueing thread with respect to the eligible scheduler.
- Equal** While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is equal to the priority of the enqueueing thread with respect to the eligible scheduler.
- Low** While a priority queue of the thread queue associated with an eligible scheduler of the enqueueing thread is non-empty, while the highest priority of the priority queue is lower than the priority of the enqueueing thread with respect to the eligible scheduler.

QueueIneligible

- None** While each priority queue of the thread queue is associated with an eligible scheduler of the enqueueing thread.
- Only** While exactly one priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread.
- Before** While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while the priority queue is positioned before all priority queues which are associated with eligible schedulers of the enqueueing thread.
- After** While a priority queue of the thread queue exists which is not associated with an eligible scheduler of the enqueueing thread, while the priority queue is positioned after all priority queues which are associated with eligible schedulers of the enqueueing thread.

PriorityForOwner

- Vital** While at least one priority of the enqueueing thread is higher than the highest priority of the owner of the thread queue.
- Disposable** While all priorities of the enqueueing thread are lower than or equal to the highest priority of the owner of the thread queue.

SchedulerForOwner

Vital While at least one eligible scheduler of the enqueueing thread is not an eligible scheduler of the owner of the thread queue.

Dispensable While all eligible schedulers of the enqueueing thread are an eligible scheduler of the owner of the thread queue.

OwnerState

NotEnqueued While the owner of the thread queue is not enqueued on a thread queue.

FIFO While the owner of the thread queue is enqueued on a thread queue in FIFO order.

Priority While the owner of the thread queue is enqueued on a thread queue in priority order.

PriorityInherit While the owner of the thread queue is enqueued on a thread queue in priority order with priority inheritance.

5.1.431.2 post-conditions

Position

InitialFirst A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the first priority queue of the thread queue.

InitialLast A priority queue associated with the scheduler which contains exactly the enqueueing thread shall be created as the last priority queue of the thread queue.

First The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

Second The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

FirstFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondFirst The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

FirstLast The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

SecondLast The enqueueing thread shall be enqueued in the priority queue associated with the scheduler.

The position of the priority queue in the thread queue shall not change.

OwnerPriority

Raise Each priority of the enqueueing thread which is higher than the highest priority of the owner of the thread queue shall be made the highest priority of the owner.

Nop The priorities of the owner of the thread queue shall not change.

OwnerScheduler

NewHelper Each eligible scheduler of the enqueueing thread which is not an eligible scheduler of the owner of the thread queue shall be made a helping scheduler of the owner with the priority of the enqueueing thread.

Nop The set of eligible schedulers of the owner of the thread queue shall not change.

OwnerOwnerPriority

Raise Each priority of the enqueueing thread which is higher than the highest priority of the owner of the thread queue on which the owner of the thread queue is enqueued shall be made the highest priority of the owner.

Nop The priorities of the owner of the thread queue on which the owner of the thread queue is enqueued shall not change.

OwnerOwnerScheduler

NewHelper Each eligible scheduler of the enqueueing thread which is not an eligible scheduler of the owner of the thread queue on which the owner of the thread queue is enqueued shall be made a helping scheduler of the owner with the priority of the enqueueing thread.

Nop The set of eligible schedulers of the owner of the thread queue on which the owner of the thread queue is enqueued shall not change.

5.1.431.3 skip-reasons

NoOtherScheduler Where the system has exactly one scheduler, no other scheduler can exist.

NeedsAnotherQueue A priority queue must be present to have another priority queue positioned before or after the priority queue.

NoOtherQueue If only one priority queue shall be present, then no other priority queue can exist.

NeedsSchedulerForVitalPriority A dedicated scheduler is necessary to provide a vital priority for the owner.

NeedsSchedulerForVitalScheduler A dedicated scheduler is necessary to provide a vital eligible scheduler for the owner.

5.1.431.4 transition-map

Position = *InitialLast*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *InitialLast*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *FirstLast*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *FirstLast*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *FirstFirst*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *FirstFirst*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *SecondLast*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = *More*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *SecondFirst*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = *More*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *InitialFirst*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *InitialFirst*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *InitialLast*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *First*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *First*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *FirstLast*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *FirstFirst*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *InitialFirst*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *First*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *Second*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *SecondLast*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three, More* }, QueueEligible = { *High, Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *SecondLast*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three, More* }, QueueEligible = { *High, Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *SecondFirst*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three, More* }, QueueEligible = { *High, Equal* }, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *SecondFirst*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Three, More* }, QueueEligible = { *High, Equal* }, QueueIneligible = *After*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *Second*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Raise*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two, Three, More* }, QueueEligible = { *High, Equal* }, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = *PriorityInherit*

Position = *Second*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *NewHelper*

- Scheduler = { *Two, Three, More* }, QueueEligible = { *High, Equal* }, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = *PriorityInherit*

Position = *InitialLast*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three, More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialLast*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three, More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *FirstLast*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *FirstLast*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *FirstFirst*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *FirstFirst*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondLast*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = *More*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondFirst*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = *More*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialFirst*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialFirst*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialLast*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *First*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *First*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *FirstLast*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *FirstFirst*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialFirst*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *First*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialLast*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *FirstLast*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *FirstFirst*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *After*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *Second*, OwnerPriority = *Raise*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondLast*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondLast*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondFirst*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondFirst*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *InitialFirst*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *First*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = *Low*, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *Second*, OwnerPriority = *Raise*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *Second*, OwnerPriority = *Nop*, OwnerScheduler = *NewHelper*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority* }

Position = *SecondLast*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *SecondFirst*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

Position = *Second*, OwnerPriority = *Nop*, OwnerScheduler = *Nop*, OwnerOwnerPriority = *Nop*, OwnerOwnerScheduler = *Nop*

- Scheduler = { *One*, *Two*, *Three*, *More* }, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

NoOtherScheduler

- Scheduler = *One*, QueueEligible = { *None*, *High*, *Equal*, *Low* }, QueueIneligible = *None*, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *One*, QueueEligible = { *None*, *High*, *Equal*, *Low* }, QueueIneligible = { *Only*, *Before*, *After* }, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = { *Vital*, *Dispensable* }, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

NeedsAnotherQueue

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = *None*, QueueIneligible = { *Before*, *After* }, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = { *Vital*, *Dispensable* }, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

NoOtherQueue

- Scheduler = { *Two*, *Three*, *More* }, QueueEligible = { *High*, *Equal*, *Low* }, QueueIneligible = *Only*, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = { *Vital*, *Dispensable* }, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

NeedsSchedulerForVitalPriority

- Scheduler = *One*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *None*, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *Two*, QueueEligible = { *High*, *Equal* }, QueueIneligible = { *Before*, *After* }, PriorityForOwner = *Vital*, SchedulerForOwner = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

NeedsSchedulerForVitalScheduler

- Scheduler = *Two*, QueueEligible = *None*, QueueIneligible = *Only*, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *Two*, QueueEligible = { *High*, *Equal* }, QueueIneligible = { *None*, *Before* }, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *Two*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *Before*, PriorityForOwner = *Dispensable*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *Two*, QueueEligible = { *High*, *Equal* }, QueueIneligible = *After*, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *Two*, QueueEligible = *Low*, QueueIneligible = { *Before*, *After* }, PriorityForOwner = { *Vital*, *Dispensable* }, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }
- Scheduler = *Three*, QueueEligible = { *High*, *Equal* }, QueueIneligible = { *Before*, *After* }, PriorityForOwner = *Vital*, SchedulerForOwner = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }

5.1.432 spec:/score/tq/req/fatal-enqueue-sticky-from-bad-state

spec:/score/tq/req/fatal-enqueue-sticky-from-bad-state

While no deadlock was detected in a sticky thread queue enqueue, while the thread dispatch disable level is not equal to one, if the sticky thread queue enqueue is performed, then the system shall terminate with the INTERNAL_ERROR_CORE fatal source and the INTERNAL_ERROR_THREAD_QUEUE_ENQUEUE_STICKY_FROM_BAD_STATE fatal code.

rationale: N/A

functional-type: function

This requirement refines *spec:/req/fatal-error*.

This requirement refines *spec:/score/tq/if/group*.

5.1.433 spec:/score/tq/req/flush-fifo

spec:/score/tq/req/flush-fifo

When the FIFO thread queue is flushed.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.433.1 pre-conditions

Queue

Empty While the thread queue is empty.

NonEmpty While the thread queue has at least one enqueued thread.

5.1.433.2 post-conditions

Operation

Nop No operation shall be performed.

TryExtract The enqueued threads of the thread queue may be extracted in FIFO order.

5.1.433.3 transition-map

Operation = *Nop*

- Queue = *Empty*

Operation = *TryExtract*

- Queue = *NonEmpty*

5.1.434 spec:/score/tq/req/flush-filter

spec:/score/tq/req/flush-filter

The thread queue flush operation shall invoke the caller provided filter handler for each thread to extract.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/tq/if/group*.

5.1.435 spec:/score/tq/req/flush-filter-stop

spec:/score/tq/req/flush-filter-stop

When the caller provided filter handler returns a value equal to NULL, the thread queue flush operation shall stop extracting threads from the thread queue.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/tq/if/group*.

5.1.436 spec:/score/tq/req/flush-priority

spec:/score/tq/req/flush-priority

When the priority thread queue is flushed.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.436.1 pre-conditions

Queue

Empty While the thread queue is empty.

NonEmpty While the thread queue has at least one enqueued thread.

5.1.436.2 post-conditions

Operation

Nop No operation shall be performed.

TryExtract The enqueued threads of the thread queue may be extracted in priority order for each priority queue associated with a scheduler. The priority queues of the thread queue shall be accessed in FIFO order.

5.1.436.3 transition-map

Operation = *Nop*

- Queue = *Empty*

Operation = *TryExtract*

- Queue = *NonEmpty*

5.1.437 spec:/score/tq/req/flush-priority-inherit

spec:/score/tq/req/flush-priority-inherit

When the priority thread queue with support for priority inheritance is flushed.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.437.1 pre-conditions

Priority

Minimum While a minimum priority of the owner of the thread queue is inherited from a thread enqueued on the thread queue.

NotMinimum While no minimum priority of the owner of the thread queue is inherited from a thread enqueued on the thread queue.

Queue

Empty While the thread queue is empty.

NonEmpty While the thread queue has at least one enqueued thread.

5.1.437.2 post-conditions

Extract

Nop No operation shall be performed.

All The enqueued threads of the thread queue shall be extracted in priority order for each priority queue associated with a scheduler. The priority queues of the thread queue shall be accessed in FIFO order.

PriorityUpdate

No The current priority of the owner of the thread queue shall not be updated by the thread queue flush operation.

Yes The current priority of the owner of the thread queue shall be updated by the thread queue flush operation to reflect the loss of inherited priorities of the flushed threads.

5.1.437.3 transition-map

Extract = *All*, PriorityUpdate = *Yes*

- Priority = *Minimum*, Queue = *NonEmpty*

Extract = *All*, PriorityUpdate = *No*

- Priority = *NotMinimum*, Queue = *NonEmpty*

Extract = *Nop*, PriorityUpdate = *No*

- Priority = NA, Queue = *Empty*

5.1.438 spec:/score/tq/req/flush-remove-timer

spec:/score/tq/req/flush-remove-timer

While the extracted thread is in the blocked wait state, the thread queue flush operation shall remove the thread timer.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/tq/if/group*.

5.1.439 spec:/score/tq/req/flush-unblock

spec:/score/tq/req/flush-unblock

While the extracted thread is in the blocked wait state, the thread queue flush operation shall unblock the thread.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/tq/if/group*.

5.1.440 spec:/score/tq/req/priority-change

spec:/score/tq/req/priority-change

Where the thread queue uses a priority discipline, when the priority of an enqueued thread changes, the position of the enqueued thread shall be changed to reflect its new priority.

rationale: N/A

functional-type: function

This requirement refines *spec:/score/tq/if/group*.

5.1.441 spec:/score/tq/req/surrender

spec:/score/tq/req/surrender

When the thread queue enqueue operation timed out.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.441.1 pre-conditions

HasOwner

Yes Where the thread queue has a previous owner thread.

No Where the thread queue has no owner threads.

Discipline

FIFO Where the thread queue uses the FIFO discipline.

Priority Where the thread queue uses the priority discipline.

WaitState

Blocked While the dequeued thread is in the blocked wait state.

IntendToBlock While the dequeued thread is in the intend to block wait state.

5.1.441.2 post-conditions

Dequeue

FIFO The first thread in FIFO order shall be dequeued from the thread queue.

Priority The first thread in priority order shall be dequeued from the thread queue.

Unblock

Yes The dequeued thread shall be unblocked by surrender operation.

No The dequeued thread shall not be unblocked by surrender operation.

5.1.441.3 skip-reasons

OnlyOneExecutingThread Where the system was built with SMP support disabled, there is at most one executing thread. Thread queues with an owner can only be surrendered by the previous owner thread. Thus, the dequeued thread cannot be in the intend to block wait state.

5.1.441.4 transition-map

Dequeue = **FIFO**, Unblock = **Yes**

- HasOwner = { **Yes**, **No** }, Discipline = **FIFO**, WaitState = **Blocked**

Dequeue = **FIFO**, Unblock = **No**

- HasOwner = { **Yes**, **No** }, Discipline = **FIFO**, WaitState = **IntendToBlock**

Dequeue = **Priority**, Unblock = **Yes**

- HasOwner = { **Yes**, **No** }, Discipline = **Priority**, WaitState = **Blocked**

Dequeue = **Priority**, Unblock = **No**

- HasOwner = { Yes, No }, Discipline = *Priority*, WaitState = *IntendToBlock*

5.1.442 spec:/score/tq/req/surrender-mrsp

spec:/score/tq/req/surrender-mrsp

When the thread queue is surrendered.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.442.1 pre-conditions

InheritedPriority

Vital While at least one priority inherited through the thread queue for the previous owner is the highest priority of the previous owner.

Disposable While all priorities inherited through the thread queue for the previous owner are not the highest priority of the previous owner.

PreviousHelping

Vital While at least one helping scheduler of the previous owner is only available due to a priority inherited through the thread queue.

Disposable While all helping scheduler of the previous owner are not only available due to a priority inherited through the thread queue.

Scheduler

Home While the previous owner executes in its home scheduler.

Helping While the previous owner executes in a helping scheduler which is available due to a priority inherited through the thread queue.

NewHelping

Vital While at least one helping scheduler of the new owner is only available due to a priority inherited through the thread queue.

Disposable While all helping scheduler of the new owner are not only available due to a priority inherited through the thread queue.

Suspended

Yes Whiel the new owner is suspended.

No Whiel the new owner is not suspended.

WaitState

IntendToBlock While the new owner is in the intend to block wait state.

5.1.442.2 post-conditions

Dequeue

Priority The first thread in priority order shall be dequeued from the thread queue.

Unblock

No The dequeued thread shall not be unblocked by the thread queue surrender operation.

PreviousOwnerPriority

Drop Each eligible priority of the previous owner which had the highest priority inherited through the thread queue shall be updated.

Nop No eligible priority of the previous owner shall be updated.

RemoveHelper

Yes Each helping scheduler of the previous owner which was only available due to a priority inherited through the thread queue shall be removed from the previous owner.

No No helping scheduler shall be removed from the previous owner.

AddHelper

Yes Each helping scheduler of the new owner which is only available due to a priority inherited through the thread queue shall be added to the new owner.

No No helping scheduler shall be added to the new owner.

Suspended

Yes The new owner shall be suspended.

No The new owner shall not be suspended.

5.1.442.3 transition-map

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *Yes*

- InheritedPriority = *Vital*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- InheritedPriority = *Vital*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *Yes*

- InheritedPriority = *Vital*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- InheritedPriority = *Vital*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *Yes*

- InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *Yes*

- InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *Yes*

- InheritedPriority = *Dispensable*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- InheritedPriority = *Dispensable*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *Yes*

- InheritedPriority = *Dispensable*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- InheritedPriority = *Dispensable*, PreviousHelping = *Vital*, Scheduler = { *Home*, *Helping* }, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *Yes*

- InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *Yes*

- InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, Scheduler = { *Home*, *Helping* }, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*

5.1.443 spec:/score/tq/req/surrender-priority-inherit

spec:/score/tq/req/surrender-priority-inherit

When the thread queue enqueue operation timed out.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.443.1 pre-conditions

SchedulerCount

One Where the system has exactly one schedulers.

Two Where the system has exactly two schedulers.

More Where the system has at least three schedulers.

InheritedPriority

Vital While at least one priority inherited through the thread queue for the previous owner is the highest priority of the previous owner.

Disposable While all priorities inherited through the thread queue for the previous owner are not the highest priority of the previous owner.

PreviousHelping

Vital While at least one helping scheduler of the previous owner is only available due to a priority inherited through the thread queue.

Disposable While all helping scheduler of the previous owner are not only available due to a priority inherited through the thread queue.

UsedScheduler

Home While the previous owner executes in its home scheduler.

Helping While the previous owner executes in a helping scheduler which is available due to a priority inherited through the thread queue.

NewPriority

Vital While at least one highest priority of the new owner is only available due to a priority inherited through the thread queue.

Disposable While all highest priorities of the new owner are not only available due to a priority inherited through the thread queue.

NewHelping

Vital While at least one helping scheduler of the new owner is only available due to a priority inherited through the thread queue.

Disposable While all helping scheduler of the new owner are not only available due to a priority inherited through the thread queue.

Suspended

Yes Whiel the new owner is suspended.

No Whiel the new owner is not suspended.

WaitState

Blocked While the new owner is in the blocked wait state.

IntendToBlock While the new owner is in the intend to block wait state.

5.1.443.2 post-conditions

Dequeue

Priority The first thread in priority order shall be dequeued from the thread queue.

Unblock

Yes The dequeued thread shall be unblocked by the thread queue surrender operation.

No The dequeued thread shall not be unblocked by the thread queue surrender operation.

PreviousOwnerPriority

Drop Each eligible priority of the previous owner which had the highest priority inherited through the thread queue shall be updated.

Nop No eligible priority of the previous owner shall be updated.

NewPriority

Raise Each eligible priority of the new owner which inherited the highest priority through the thread queue shall be updated.

Nop No highest priority of the new owner shall be changed.

RemoveHelper

Yes Each helping scheduler of the previous owner which was only available due to a priority inherited through the thread queue shall be removed from the previous owner.

No No helping scheduler shall be removed from the previous owner.

AddHelper

Yes Each helping scheduler of the new owner which is only available due to a priority inherited through the thread queue shall be added to the new owner.

No No helping scheduler shall be added to the new owner.

Suspended

Yes The new owner shall be suspended.

No The new owner shall not be suspended.

5.1.443.3 skip-reasons

NeedsSecondCPUForIntentToBlock Where the system was built with SMP support disabled, exactly one scheduler is present in an application using exactly one processor. There is at most one executing thread. Thread queues with an owner can only be surrendered by the previous owner. Thus, the new owner of the thread queue cannot be in the intend to block wait state.

NeedsSecondScheduler A second scheduler is required to get a helping scheduler.

NeedsThirdScheduler A third scheduler is required to get a vital scheduler and a vital priority for the new owner at the same time.

5.1.443.4 transition-map

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Dispensable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *No*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *One*, *Two* }, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = *Home*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*
- SchedulerCount = *Two*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = *Helping*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *One*, *Two* }, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = *Home*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*
- SchedulerCount = *Two*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = *Helping*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*
- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *Yes*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *One*, *Two* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = *Home*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*
- SchedulerCount = *Two*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = *Helping*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*
- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *Blocked*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *No*

- SchedulerCount = { *One*, *Two* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = *Home*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*
- SchedulerCount = *Two*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = *Helping*, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*
- SchedulerCount = *More*, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *No*, WaitState = *IntendToBlock*

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Vital*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *Yes*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Disposable*, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Raise*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Disposable*, PreviousHelping = *Disposable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Disposable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *Yes*, Suspended = *Yes*

- SchedulerCount = { *Two*, *More* }, InheritedPriority = *Dispensable*, PreviousHelping = *Dispensable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Vital*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Drop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *One*, *Two* }, InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, UsedScheduler = *Home*, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *Two*, InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, UsedScheduler = *Helping*, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *More*, InheritedPriority = *Vital*, PreviousHelping = *Dispensable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

Dequeue = *Priority*, Unblock = *No*, PreviousOwnerPriority = *Nop*, NewPriority = *Nop*, RemoveHelper = *No*, AddHelper = *No*, Suspended = *Yes*

- SchedulerCount = { *One*, *Two* }, InheritedPriority = *Dispensable*, PreviousHelping = *Dispensable*, UsedScheduler = *Home*, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *Two*, InheritedPriority = *Dispensable*, PreviousHelping = *Dispensable*, UsedScheduler = *Helping*, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *More*, InheritedPriority = *Dispensable*, PreviousHelping = *Dispensable*, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Dispensable*, NewHelping = *Dispensable*, Suspended = *Yes*, WaitState = { *Blocked*, *IntendToBlock* }

NeedsSecondScheduler

- SchedulerCount = *One*, InheritedPriority = { *Vital*, *Dispensable* }, PreviousHelping = *Vital*, UsedScheduler = { *Home*, *Helping* }, NewPriority = { *Vital*, *Dispensable* }, NewHelping = { *Vital*, *Dispensable* }, Suspended = { *Yes*, *No* }, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *One*, InheritedPriority = { *Vital*, *Dispensable* }, PreviousHelping = *Dispensable*, UsedScheduler = *Home*, NewPriority = *Vital*, NewHelping = { *Vital*, *Dispensable* }, Suspended = { *Yes*, *No* }, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *One*, InheritedPriority = { *Vital*, *Dispensable* }, PreviousHelping = *Dispensable*, UsedScheduler = *Home*, NewPriority = *Dispensable*, NewHelping = *Vital*, Suspended = { *Yes*, *No* }, WaitState = { *Blocked*, *IntendToBlock* }
- SchedulerCount = *One*, InheritedPriority = { *Vital*, *Dispensable* }, PreviousHelping = *Dispensable*, UsedScheduler = *Helping*, NewPriority = { *Vital*, *Dispensable* }, NewHelping = { *Vital*, *Dispensable* }, Suspended = { *Yes*, *No* }, WaitState = { *Blocked*, *IntendToBlock* }

NeedsThirdScheduler

- SchedulerCount = *Two*, InheritedPriority = { *Vital*, *Disposable* }, PreviousHelping = { *Vital*, *Disposable* }, UsedScheduler = { *Home*, *Helping* }, NewPriority = *Vital*, NewHelping = *Vital*, Suspended = { *Yes*, *No* }, WaitState = { *Blocked*, *IntendToBlock* }

5.1.444 spec:/score/tq/req/timeout

spec:/score/tq/req/timeout

When the thread queue enqueue operation timed out.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.444.1 pre-conditions

WaitState

Blocked While the thread of the timeout operation is in the blocked wait state.

IntendToBlock While the thread of the timeout operation is in the intend to block wait state.

ReadyAgain While the thread of the timeout operation is in the ready again wait state.

5.1.444.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

Timeout The return status of the directive call shall be derived from STATUS_TIMEOUT.

Unblock

Yes The thread of the timeout operation shall be unblocked by the timeout operation.

No The thread of the timeout operation shall not be unblocked by the timeout operation.

5.1.444.3 transition-map

Status = *Timeout*, Unblock = *Yes*

- WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*

- WaitState = *IntendToBlock*

Status = *Ok*, Unblock = *No*

- WaitState = *ReadyAgain*

5.1.445 spec:/score/tq/req/timeout-mrsp

spec:/score/tq/req/timeout-mrsp

When the thread queue enqueue operation timed out.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.445.1 pre-conditions

Scheduler

Same While the home scheduler of the thread is equal to the home scheduler of the thread queue owner.

Other While the home scheduler of the thread is not equal to the home scheduler of the thread queue owner.

WaitState

IntendToBlock While the thread of the timeout operation is in the intend to block wait state.

ReadyAgain While the thread of the timeout operation is in the ready again wait state.

5.1.445.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

Timeout The return status of the directive call shall be derived from STATUS_TIMEOUT.

Unblock

No The thread of the timeout operation shall not be unblocked by the timeout operation.

5.1.445.3 transition-map

Status = *Timeout*, Unblock = *No*

- Scheduler = { *Same*, *Other* }, WaitState = *IntendToBlock*

Status = *Ok*, Unblock = *No*

- Scheduler = { *Same*, *Other* }, WaitState = *ReadyAgain*

5.1.446 spec:/score/tq/req/timeout-priority-inherit

spec:/score/tq/req/timeout-priority-inherit

When the thread queue enqueue operation timed out.

rationale: N/A

functional-type: action

This requirement refines *spec:/score/tq/if/group*.

5.1.446.1 pre-conditions

HomeScheduler

Home While the home scheduler of the thread is the home scheduler of the thread queue owner.

Helping While the home scheduler of the thread is a helping scheduler of the thread queue owner.

EligibleScheduler

One While the thread has exactly one eligible scheduler.

More While the thread has at least two eligible scheduler.

Queue

Only While the priority node of the thread is the only priority node in the priority queue associated with the scheduler of the thread queue.

Vital While the priority node of the thread is not the only priority node in the priority queue associated with the scheduler of the thread queue, while the priority node of the thread is the highest priority node in the priority queue.

Disposable While the priority node of the thread is not the only priority node in the priority queue associated with the scheduler of the thread queue, while the priority node of the thread is not the highest priority node in the priority queue.

OwnerPriority

Only While the priority node of the thread queue is the only priority node associated with the scheduler available to the owner.

Vital While the priority node of the thread queue is not the only priority node associated with the scheduler available to the owner, while the priority node of the thread queue is the highest priority node available to the owner.

Disposable While the priority node of the thread queue is not the only priority node associated with the scheduler available to the owner, while the priority node of the thread queue is not the highest priority node available to the owner.

OwnerState

NotEnqueued While the owner of the thread queue is not enqueued on a thread queue.

FIFO While the owner of the thread queue is enqueued on a thread queue in FIFO order.

Priority While the owner of the thread queue is enqueued on a thread queue in priority order.

PriorityInherit While the owner of the thread queue is enqueued on a thread queue in priority order with priority inheritance.

OwnerQueue

Only While the priority node of the owner is the only priority node in the priority queue associated with the scheduler of the thread queue on which the owner is enqueued.

Vital While the priority node of the owner is not the only priority node in the priority queue associated with the scheduler of the thread queue on which the owner is enqueued, while the priority node of the owner is the highest priority node in the priority queue.

Disposable While the priority node of the owner is not the only priority node in the priority queue associated with the scheduler of the thread queue on

which the owner is enqueued, while the priority node of the owner is not the highest priority node in the priority queue.

OwnerOwnerPriority

Only While the priority node of the thread queue on which the owner is enqueued is the only priority node associated with the scheduler available to the owner of the thread queue on which the owner is enqueued.

Vital While the priority node of the thread queue on which the owner is enqueued is not the only priority node associated with the scheduler available to the owner of the thread queue on which the owner is enqueued, while the priority node of the thread queue on which the owner is enqueued is the highest priority node available to the owner of the thread queue on which the owner is enqueued.

Disposable While the priority node of the thread queue on which the owner is enqueued is not the only priority node associated with the scheduler available to the owner of the thread queue on which the owner is enqueued, while the priority node of the thread queue is on which the owner is enqueued not the highest priority node available to the owner of the thread queue on which the owner is enqueued.

WaitState

Blocked While the thread is in the blocked wait state.

IntendToBlock While the thread is in the intend to block wait state.

ReadyAgain While the thread is in the ready again wait state.

5.1.446.2 post-conditions

Status

Ok The return status of the directive call shall be derived from STATUS_SUCCESSFUL.

Timeout The return status of the directive call shall be derived from STATUS_TIMEOUT.

Unblock

Yes The thread shall be unblocked by the timeout operation.

No The thread shall not be unblocked by the timeout operation.

OwnerPriority

Nop The priority of the owner with respect to the scheduler shall not change by the timeout operation.

Lower The priority of the owner with respect to the scheduler shall be lowered to the next highest priority.

Drop The owner shall not have a priority with respect to the scheduler.

OwnerOwnerPriority

- Nop** The priority of the owner of the thread queue on which the owner is enqueued with respect to the scheduler shall not change by the timeout operation.
- Lower** The priority of the owner of the thread queue on which the owner is enqueued with respect to the scheduler shall be lowered to the next highest priority.
- Drop** The owner of the thread queue on which the owner is enqueued shall not have a priority with respect to the scheduler.

5.1.446.3 skip-reasons

- ReadyAgainNeedsSurrender** For the ready again wait state, the owner must surrender the thread queue to the thread.
- OnlyOneCPU** Where the system was built with SMP support disabled, exactly one scheduler is present in an application using exactly one processor.
- HomeHasRealPriority** There is always at least the real priority available for the home scheduler.
- DisposableStopsVital** Vital priority nodes cannot be after a dispensable priority node.

5.1.446.4 transition-map

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Drop*, OwnerOwnerPriority = *Drop*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Only*, OwnerOwnerPriority = *Only*, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Drop*, OwnerOwnerPriority = *Drop*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Only*, OwnerOwnerPriority = *Only*, WaitState = *IntendToBlock*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Drop*, OwnerOwnerPriority = *Lower*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Only*, OwnerOwnerPriority = *Vital*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Vital*, OwnerOwnerPriority = { *Only*, *Vital* }, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Drop*, OwnerOwnerPriority = *Lower*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Only*, OwnerOwnerPriority = *Vital*, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Vital*, OwnerOwnerPriority = { *Only*, *Vital* }, WaitState = *IntendToBlock*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Drop*, OwnerOwnerPriority = *Nop*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Dispensable*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Dispensable*, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Drop*, OwnerOwnerPriority = *Nop*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Dispensable*, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Dispensable*, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *IntendToBlock*

Status = *Ok*, Unblock = *No*, OwnerPriority = *Nop*, OwnerOwnerPriority = *NA*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Dispensable*, OwnerState = *NotEnqueued*, OwnerQueue = *NA*, OwnerOwnerPriority = *NA*, WaitState = *ReadyAgain*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Dispensable*, OwnerState = *NotEnqueued*, OwnerQueue = *NA*, OwnerOwnerPriority = *NA*, WaitState = *ReadyAgain*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Lower*, OwnerOwnerPriority = *Lower*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Vital*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Only*, *Vital* }, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = { *Only*, *Vital* }, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Only*, *Vital* }, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Lower*, OwnerOwnerPriority = *Lower*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Vital*, WaitState = *IntendToBlock*

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Only*, *Vital* }, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = { *Only*, *Vital* }, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Only*, *Vital* }, WaitState = *IntendToBlock*

Status = *Ok*, Unblock = *No*, OwnerPriority = *Drop*, OwnerOwnerPriority = NA

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Only*, OwnerState = *NotEnqueued*, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *ReadyAgain*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Lower*, OwnerOwnerPriority = *Nop*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = *Disposable*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Disposable*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = *Disposable*, OwnerOwnerPriority = { *Only*, *Disposable* }, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Disposable*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Disposable*, OwnerOwnerPriority = { *Only*, *Disposable* }, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Lower*, OwnerOwnerPriority = *Nop*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = *Disposable*, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Disposable*, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *PriorityInherit*, OwnerQueue = *Disposable*, OwnerOwnerPriority = { *Only*, *Disposable* }, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = *Disposable*, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = *PriorityInherit*, OwnerQueue = *Disposable*, OwnerOwnerPriority = { *Only*, *Disposable* }, WaitState = *IntendToBlock*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Nop*, OwnerOwnerPriority = *Nop*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = *Dispensable*, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Dispensable*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = { *Only*, *Dispensable* }, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Nop*, OwnerOwnerPriority = *Nop*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = *Dispensable*, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Dispensable*, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = { *Only*, *Dispensable* }, OwnerState = *PriorityInherit*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *IntendToBlock*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Drop*, OwnerOwnerPriority = NA

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Drop*, OwnerOwnerPriority = NA

- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Only*, OwnerPriority = *Only*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*

Status = *Ok*, Unblock = *No*, OwnerPriority = *Lower*, OwnerOwnerPriority = NA

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *NotEnqueued*, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *ReadyAgain*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* },

OwnerPriority = *Vital*, OwnerState = *NotEnqueued*, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *ReadyAgain*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Nop*, OwnerOwnerPriority = NA

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = *Only*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Nop*, OwnerOwnerPriority = NA

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = *Only*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*

Status = *Timeout*, Unblock = *Yes*, OwnerPriority = *Lower*, OwnerOwnerPriority = NA

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *Blocked*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = NA, WaitState = *Blocked*

Status = *Timeout*, Unblock = *No*, OwnerPriority = *Lower*, OwnerOwnerPriority = NA

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO* }, OwnerQueue = NA, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Vital*, OwnerPriority = *Only*, OwnerState = *Priority*, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = NA, WaitState = *IntendToBlock*

ReadyAgainNeedsSurrender

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Vital*, *Disposable* }, WaitState = *ReadyAgain*

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = *Dispensable*, OwnerOwnerPriority = *Dispensable*, WaitState = *ReadyAgain*
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Dispensable*, OwnerState = { *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = *Dispensable*, WaitState = *ReadyAgain*
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = *Dispensable*, WaitState = *ReadyAgain*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = { *Only*, *Vital* }, OwnerState = { *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Only*, *Vital*, *Dispensable* }, WaitState = *ReadyAgain*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = { *Only*, *Vital* }, OwnerState = { *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = *Dispensable*, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *ReadyAgain*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Dispensable*, OwnerState = { *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *ReadyAgain*
- HomeScheduler = *Helping*, EligibleScheduler = { *One*, *More* }, Queue = *Dispensable*, OwnerPriority = { *Only*, *Dispensable* }, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = *ReadyAgain*

HomeHasRealPriority

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Vital* }, OwnerOwnerPriority = { *Only*, *Vital*, *Dispensable* }, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = *Dispensable*, OwnerOwnerPriority = { *Only*, *Dispensable* }, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Vital*, *Dispensable* }, OwnerOwnerPriority = *Only*, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Dispensable* }, OwnerPriority = *Dispensable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Dispensable* }, OwnerOwnerPriority = *Only*, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }

- HomeScheduler = *Home*, EligibleScheduler = { *One*, *More* }, Queue = *Disposable*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Disposable* }, OwnerOwnerPriority = { *Only*, *Disposable* }, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }

DisposableStopsVital

- HomeScheduler = { *Home*, *Helping* }, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital* }, OwnerPriority = { *Only*, *Vital* }, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = *Disposable*, OwnerOwnerPriority = *Vital*, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = { *Home*, *Helping* }, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Disposable* }, OwnerPriority = *Disposable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Disposable* }, OwnerOwnerPriority = *Vital*, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = { *Home*, *Helping* }, EligibleScheduler = { *One*, *More* }, Queue = { *Only*, *Vital*, *Disposable* }, OwnerPriority = *Disposable*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = *Vital*, OwnerOwnerPriority = { *Only*, *Vital*, *Disposable* }, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = { *Home*, *Helping* }, EligibleScheduler = { *One*, *More* }, Queue = *Disposable*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Disposable* }, OwnerOwnerPriority = *Vital*, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = { *Home*, *Helping* }, EligibleScheduler = { *One*, *More* }, Queue = *Disposable*, OwnerPriority = *Only*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = *Vital*, OwnerOwnerPriority = { *Only*, *Vital*, *Disposable* }, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }
- HomeScheduler = { *Home*, *Helping* }, EligibleScheduler = { *One*, *More* }, Queue = *Disposable*, OwnerPriority = *Vital*, OwnerState = { *NotEnqueued*, *FIFO*, *Priority*, *PriorityInherit* }, OwnerQueue = { *Only*, *Vital*, *Disposable* }, OwnerOwnerPriority = { *Only*, *Vital*, *Disposable* }, WaitState = { *Blocked*, *IntendToBlock*, *ReadyAgain* }

5.2 Performance requirements

5.2.1 spec:/req/perf-runtime

spec:/req/perf-runtime

The runtime of interface functions shall be measured.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/rtems/req/perf-runtime*

5.2.2 spec:/rtems/barrier/req/perf-release-auto

While the execution environment is below environments, while the barrier is an automatic release barrier, while the measurement sample is the runtime of exactly one successful call to rtems_barrier_wait which automatically releases the barrier, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8e-06
median-lower-bound: 2e-06
median-upper-bound: 8e-06
min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 8e-06
median-lower-bound: 2e-06
median-upper-bound: 8e-06
min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06
median-lower-bound: 2e-06
median-upper-bound: 8e-06
min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 2e-05
median-lower-bound: 5e-06
median-upper-bound: 2e-05
min-lower-bound: 4.5e-06

Load/2

max-upper-bound: 2e-05
median-lower-bound: 5e-06
median-upper-bound: 2e-05

min-lower-bound: 4.5e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574IY=

DirtyCache

max-upper-bound: 3.6e-05

median-lower-bound: 9e-06

median-upper-bound: 3.6e-05

min-lower-bound: 8.5e-06

FullCache

max-upper-bound: 2e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 6.2e-05

median-lower-bound: 1.5e-05

median-upper-bound: 6e-05

min-lower-bound: 1.5e-05

Load/2

max-upper-bound: 6.4e-05

median-lower-bound: 1.55e-05

median-upper-bound: 6.2e-05

min-lower-bound: 1.55e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.2e-05
median-lower-bound: 5.5e-06
median-upper-bound: 2.2e-05
min-lower-bound: 5e-06

FullCache

max-upper-bound: 1.8e-05
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

Load/1

max-upper-bound: 3.6e-05

median-lower-bound: 8.5e-06

median-upper-bound: 3.4e-05

min-lower-bound: 8.5e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcl0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.56e-06

median-lower-bound: 1.89e-06

median-upper-bound: 7.56e-06

min-lower-bound: 1.85e-06

FullCache

max-upper-bound: 7.08e-06

median-lower-bound: 1.77e-06

median-upper-bound: 7.08e-06

min-lower-bound: 1.74e-06

HotCache

max-upper-bound: 7.2e-06

median-lower-bound: 1.8e-06

median-upper-bound: 7.2e-06

min-lower-bound: 1.8e-06

Load/1

max-upper-bound: 1.72e-05

median-lower-bound: 4.28e-06

median-upper-bound: 1.712e-05

min-lower-bound: 4.18e-06

Load/2

max-upper-bound: 1.72e-05

median-lower-bound: 4.28e-06

median-upper-bound: 1.712e-05

min-lower-bound: 4.18e-06

Load/3

max-upper-bound: 1.72e-05

median-lower-bound: 4.28e-06

median-upper-bound: 1.712e-05

min-lower-bound: 4.18e-06

Load/4

max-upper-bound: 1.72e-05

median-lower-bound: 4.28e-06

median-upper-bound: 1.712e-05

min-lower-bound: 4.18e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 2.1384e-05

median-lower-bound: 4.64e-06

median-upper-bound: 1.856e-05

min-lower-bound: 4.33e-06

FullCache

max-upper-bound: 1.212e-05

median-lower-bound: 6.5e-07

median-upper-bound: 2.6e-06

min-lower-bound: 6.46e-07

HotCache

max-upper-bound: 2.072e-06

median-lower-bound: 3.94e-07

median-upper-bound: 1.576e-06

min-lower-bound: 3.94e-07

Load/1

max-upper-bound: 4.1864e-05

median-lower-bound: 5.888e-06

median-upper-bound: 2.3552e-05

min-lower-bound: 5.01e-06

Load/2

max-upper-bound: 8.5344e-05

median-lower-bound: 2.0074e-05

median-upper-bound: 8.0296e-05

min-lower-bound: 1.783e-05

Load/3

max-upper-bound: 0.000106392
median-lower-bound: 2.4808e-05
median-upper-bound: 9.9232e-05
min-lower-bound: 2.3666e-05

Load/4

max-upper-bound: 0.000117912
median-lower-bound: 2.827e-05
median-upper-bound: 0.00011308
min-lower-bound: 2.6474e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.36e-06
median-lower-bound: 1.57e-06
median-upper-bound: 6.28e-06
min-lower-bound: 1.57e-06

FullCache

max-upper-bound: 5.96e-06
median-lower-bound: 1.48e-06
median-upper-bound: 5.92e-06
min-lower-bound: 1.48e-06

HotCache

max-upper-bound: 5.88e-06
median-lower-bound: 1.47e-06
median-upper-bound: 5.88e-06
min-lower-bound: 1.47e-06

Load/1

max-upper-bound: 1.596e-05
median-lower-bound: 3.99e-06
median-upper-bound: 1.596e-05
min-lower-bound: 3.98e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 1.2728e-05

median-lower-bound: 2.578e-06

median-upper-bound: 1.0312e-05

min-lower-bound: 2.546e-06

FullCache

max-upper-bound: 1.164e-05

median-lower-bound: 6.42e-07

median-upper-bound: 2.568e-06

min-lower-bound: 6.4e-07

HotCache

max-upper-bound: 1.192e-06

median-lower-bound: 2.98e-07

median-upper-bound: 1.192e-06

min-lower-bound: 2.98e-07

Load/1

max-upper-bound: 1.6424e-05

median-lower-bound: 3.77e-06

median-upper-bound: 1.508e-05

min-lower-bound: 3.716e-06

This requirement refines [*spec:/rtems/barrier/req/perf-runtime*](#).

5.2.3 [*spec:/rtems/barrier/req/perf-release-auto-other-cpu*](#)

Let U and V be two tasks with distinct home schedulers. Let B be a time point measured by U right before a call to `rtems_barrier_wait` which automaticall releases the barrier. Let E be a time point measured by V right after a call to `rtems_barrier_wait` returns successfully due to the `rtems_barrier_wait` call associated with B . While the execution environment is below environments, while the barrier is an automatic release barrier, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

`sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=`

DirtyCache

max-upper-bound: 9.4e-05

median-lower-bound: 2.3e-05

median-upper-bound: 9.2e-05

min-lower-bound: 2.25e-05

FullCache

max-upper-bound: 9.2e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.2e-05

HotCache

max-upper-bound: 9e-05

median-lower-bound: 2.2e-05

median-upper-bound: 8.8e-05

min-lower-bound: 2.2e-05

Load/1

max-upper-bound: 0.000102

median-lower-bound: 2.55e-05

median-upper-bound: 0.000102

min-lower-bound: 2.5e-05

Load/2

max-upper-bound: 0.000102

median-lower-bound: 2.5e-05

median-upper-bound: 0.0001

min-lower-bound: 2.45e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000242

median-lower-bound: 6e-05

median-upper-bound: 0.00024

min-lower-bound: 5.95e-05

FullCache

max-upper-bound: 0.000172

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 2e-05

HotCache

max-upper-bound: 8e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 1.95e-05

Load/1

max-upper-bound: 0.000238

median-lower-bound: 5.9e-05

median-upper-bound: 0.000236

min-lower-bound: 5.9e-05

Load/2

max-upper-bound: 0.000272

median-lower-bound: 6.75e-05

median-upper-bound: 0.00027

min-lower-bound: 6.7e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.752e-05

median-lower-bound: 2.147e-05

median-upper-bound: 8.588e-05

min-lower-bound: 2.133e-05

FullCache

max-upper-bound: 8.468e-05

median-lower-bound: 2.108e-05

median-upper-bound: 8.432e-05

min-lower-bound: 2.097e-05

HotCache

max-upper-bound: 8.604e-05

median-lower-bound: 2.1e-05

median-upper-bound: 8.4e-05

min-lower-bound: 2.091e-05

Load/1

max-upper-bound: 9.58e-05

median-lower-bound: 2.376e-05

median-upper-bound: 9.504e-05

min-lower-bound: 2.342e-05

Load/2

max-upper-bound: 9.568e-05

median-lower-bound: 2.343e-05

median-upper-bound: 9.372e-05

min-lower-bound: 2.323e-05

Load/3

max-upper-bound: 9.568e-05

median-lower-bound: 2.345e-05

median-upper-bound: 9.38e-05

min-lower-bound: 2.323e-05

Load/4

max-upper-bound: 9.488e-05

median-lower-bound: 2.35e-05

median-upper-bound: 9.4e-05

min-lower-bound: 2.319e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 9.9392e-05

median-lower-bound: 2.0706e-05

median-upper-bound: 8.2824e-05

min-lower-bound: 2.0528e-05

FullCache

max-upper-bound: 7.3736e-05

median-lower-bound: 8.634e-06

median-upper-bound: 3.4536e-05

min-lower-bound: 8.504e-06

HotCache

max-upper-bound: 1.9168e-05

median-lower-bound: 4.614e-06

median-upper-bound: 1.8456e-05

min-lower-bound: 4.614e-06

Load/1

max-upper-bound: 0.000125128

median-lower-bound: 1.9982e-05

median-upper-bound: 7.9928e-05

min-lower-bound: 1.9758e-05

Load/2

max-upper-bound: 0.000440768

median-lower-bound: 0.00011001

median-upper-bound: 0.00044004

min-lower-bound: 0.000107924

Load/3

max-upper-bound: 0.000752952

median-lower-bound: 0.00018604

median-upper-bound: 0.00074416

min-lower-bound: 0.0001854

Load/4

max-upper-bound: 0.000955088

median-lower-bound: 0.00023813

median-upper-bound: 0.00095252

min-lower-bound: 0.000235894

This requirement refines [spec:/rtems/barrier/req/perf-runtime](#).

5.2.4 spec:/rtems/barrier/req/perf-release-manual

While the execution environment is below environments, while the barrier is a manual release barrier, while the measurement sample is the runtime of exactly one successful call to rtems_barrier_release which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 5.6e-05

median-lower-bound: 1.35e-05

median-upper-bound: 5.4e-05

min-lower-bound: 1.35e-05

FullCache

max-upper-bound: 5e-05

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.2e-05

HotCache

max-upper-bound: 5.2e-05

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.25e-05

Load/1

max-upper-bound: 6.8e-05

median-lower-bound: 1.7e-05

median-upper-bound: 6.8e-05

min-lower-bound: 1.65e-05

Load/2

max-upper-bound: 6.8e-05

median-lower-bound: 1.65e-05

median-upper-bound: 6.6e-05

min-lower-bound: 1.65e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000144

median-lower-bound: 3.55e-05

median-upper-bound: 0.000142

min-lower-bound: 3.5e-05

FullCache

max-upper-bound: 9.8e-05

median-lower-bound: 1.3e-05

median-upper-bound: 5.2e-05

min-lower-bound: 1.25e-05

HotCache

max-upper-bound: 5e-05
median-lower-bound: 1.2e-05
median-upper-bound: 4.8e-05
min-lower-bound: 1.2e-05

Load/1

max-upper-bound: 0.000156
median-lower-bound: 3.85e-05
median-upper-bound: 0.000154
min-lower-bound: 3.8e-05

Load/2

max-upper-bound: 0.000174
median-lower-bound: 4.35e-05
median-upper-bound: 0.000174
min-lower-bound: 4.3e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 7.8e-05

median-lower-bound: 1.9e-05

median-upper-bound: 7.6e-05

min-lower-bound: 1.9e-05

FullCache

max-upper-bound: 5.6e-05

median-lower-bound: 5e-06

median-upper-bound: 2e-05

min-lower-bound: 5e-06

HotCache

max-upper-bound: 2.2e-05

median-lower-bound: 5e-06

median-upper-bound: 2e-05

min-lower-bound: 4.5e-06

Load/1

max-upper-bound: 9e-05

median-lower-bound: 2.2e-05

median-upper-bound: 8.8e-05

min-lower-bound: 2.2e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.24e-05

median-lower-bound: 1.31e-05

median-upper-bound: 5.24e-05

min-lower-bound: 1.31e-05

FullCache

max-upper-bound: 4.844e-05

median-lower-bound: 1.205e-05

median-upper-bound: 4.82e-05

min-lower-bound: 1.201e-05

HotCache

max-upper-bound: 4.824e-05

median-lower-bound: 1.206e-05

median-upper-bound: 4.824e-05

min-lower-bound: 1.198e-05

Load/1

max-upper-bound: 6.376e-05

median-lower-bound: 1.594e-05

median-upper-bound: 6.376e-05

min-lower-bound: 1.578e-05

Load/2

max-upper-bound: 6.376e-05

median-lower-bound: 1.594e-05

median-upper-bound: 6.376e-05

min-lower-bound: 1.578e-05

Load/3

max-upper-bound: 6.384e-05

median-lower-bound: 1.596e-05

median-upper-bound: 6.384e-05

min-lower-bound: 1.592e-05

Load/4

max-upper-bound: 6.384e-05

median-lower-bound: 1.596e-05

median-upper-bound: 6.384e-05

min-lower-bound: 1.592e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOfIz2eLDfIYc=

DirtyCache

max-upper-bound: 5.6328e-05

median-lower-bound: 1.3736e-05

median-upper-bound: 5.4944e-05

min-lower-bound: 1.328e-05

FullCache

max-upper-bound: 3.7944e-05
median-lower-bound: 4.84e-06
median-upper-bound: 1.936e-05
min-lower-bound: 4.836e-06

HotCache

max-upper-bound: 1.0888e-05
median-lower-bound: 2.594e-06
median-upper-bound: 1.0376e-05
min-lower-bound: 2.594e-06

Load/1

max-upper-bound: 9.796e-05
median-lower-bound: 1.4458e-05
median-upper-bound: 5.7832e-05
min-lower-bound: 1.348e-05

Load/2

max-upper-bound: 0.000327536
median-lower-bound: 7.9166e-05
median-upper-bound: 0.000316664
min-lower-bound: 7.9102e-05

Load/3

max-upper-bound: 0.000500904
median-lower-bound: 0.000122864
median-upper-bound: 0.000491456
min-lower-bound: 0.000121272

Load/4

max-upper-bound: 0.000578688
median-lower-bound: 0.000143352
median-upper-bound: 0.000573408
min-lower-bound: 0.000140178

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 2.916e-05
median-lower-bound: 7.27e-06

median-upper-bound: 2.908e-05

min-lower-bound: 7.13e-06

FullCache

max-upper-bound: 2.472e-05

median-lower-bound: 6.18e-06

median-upper-bound: 2.472e-05

min-lower-bound: 6.18e-06

HotCache

max-upper-bound: 2.524e-05

median-lower-bound: 6.31e-06

median-upper-bound: 2.524e-05

min-lower-bound: 6.31e-06

Load/1

max-upper-bound: 4.068e-05

median-lower-bound: 1.007e-05

median-upper-bound: 4.028e-05

min-lower-bound: 9.97e-06

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 3.5664e-05

median-lower-bound: 8.54e-06

median-upper-bound: 3.416e-05

min-lower-bound: 8.486e-06

FullCache

max-upper-bound: 3.0464e-05

median-lower-bound: 2.886e-06

median-upper-bound: 1.1544e-05

min-lower-bound: 2.876e-06

HotCache

max-upper-bound: 4.6e-06

median-lower-bound: 1.15e-06

median-upper-bound: 4.6e-06

min-lower-bound: 1.15e-06

Load/1

max-upper-bound: 4.0496e-05
median-lower-bound: 9.764e-06
median-upper-bound: 3.9056e-05
min-lower-bound: 9.682e-06

This requirement refines [spec:/rtems/barrier/req/perf-runtime](#).

5.2.5 spec:/rtems/barrier/req/perf-release-manual-preempt

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to rtems_barrier_release which preempts the caller. Let E be a time point measured by V right after a call to rtems_barrier_wait returns successfully due to the rtems_barrier_release call associated with B .

While the execution environment is below environments, while the barrier is a manual release barrier, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 8.6e-05
median-lower-bound: 2.15e-05
median-upper-bound: 8.6e-05
min-lower-bound: 2.1e-05

FullCache

max-upper-bound: 8.4e-05
median-lower-bound: 2.05e-05
median-upper-bound: 8.2e-05
min-lower-bound: 2.05e-05

HotCache

max-upper-bound: 8.4e-05
median-lower-bound: 2.05e-05
median-upper-bound: 8.2e-05
min-lower-bound: 2.05e-05

Load/1

max-upper-bound: 9.4e-05

median-lower-bound: 2.35e-05

median-upper-bound: 9.4e-05

min-lower-bound: 2.35e-05

Load/2

max-upper-bound: 9.4e-05

median-lower-bound: 2.35e-05

median-upper-bound: 9.4e-05

min-lower-bound: 2.35e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574IY=

DirtyCache

max-upper-bound: 0.000212

median-lower-bound: 5.25e-05

median-upper-bound: 0.00021

min-lower-bound: 5.2e-05

FullCache

max-upper-bound: 0.00017

median-lower-bound: 1.85e-05

median-upper-bound: 7.4e-05

min-lower-bound: 1.8e-05

HotCache

max-upper-bound: 6.6e-05

median-lower-bound: 1.65e-05

median-upper-bound: 6.6e-05

min-lower-bound: 1.6e-05

Load/1

max-upper-bound: 0.000208

median-lower-bound: 5.05e-05

median-upper-bound: 0.000202

min-lower-bound: 5.05e-05

Load/2

max-upper-bound: 0.000248

median-lower-bound: 5.9e-05

median-upper-bound: 0.000236

min-lower-bound: 5.9e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000118

median-lower-bound: 2.9e-05

median-upper-bound: 0.000116

min-lower-bound: 2.9e-05

FullCache

max-upper-bound: 0.00011

median-lower-bound: 1.15e-05

median-upper-bound: 4.6e-05

min-lower-bound: 1.15e-05

HotCache

max-upper-bound: 4.4e-05
median-lower-bound: 1.05e-05
median-upper-bound: 4.2e-05
min-lower-bound: 1.05e-05

Load/1

max-upper-bound: 0.000126
median-lower-bound: 3.1e-05
median-upper-bound: 0.000124
min-lower-bound: 3.05e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.1e-05
median-lower-bound: 2.011e-05
median-upper-bound: 8.044e-05
min-lower-bound: 2.001e-05

FullCache

max-upper-bound: 7.836e-05
median-lower-bound: 1.957e-05
median-upper-bound: 7.828e-05
min-lower-bound: 1.957e-05

HotCache

max-upper-bound: 7.88e-05
median-lower-bound: 1.97e-05
median-upper-bound: 7.88e-05
min-lower-bound: 1.97e-05

Load/1

max-upper-bound: 8.896e-05
median-lower-bound: 2.224e-05
median-upper-bound: 8.896e-05
min-lower-bound: 2.196e-05

Load/2

max-upper-bound: 8.896e-05

median-lower-bound: 2.224e-05

median-upper-bound: 8.896e-05

min-lower-bound: 2.196e-05

Load/3

max-upper-bound: 8.896e-05

median-lower-bound: 2.224e-05

median-upper-bound: 8.896e-05

min-lower-bound: 2.196e-05

Load/4

max-upper-bound: 8.896e-05

median-lower-bound: 2.224e-05

median-upper-bound: 8.896e-05

min-lower-bound: 2.196e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFlZ2eLDfIYc=

DirtyCache

max-upper-bound: 8.1512e-05

median-lower-bound: 1.9456e-05

median-upper-bound: 7.7824e-05

min-lower-bound: 1.863e-05

FullCache

max-upper-bound: 7.4152e-05

median-lower-bound: 9.198e-06

median-upper-bound: 3.6792e-05

min-lower-bound: 9.158e-06

HotCache

max-upper-bound: 1.6408e-05

median-lower-bound: 3.988e-06

median-upper-bound: 1.5952e-05

min-lower-bound: 3.988e-06

Load/1

max-upper-bound: 0.00011332

median-lower-bound: 1.7558e-05

median-upper-bound: 7.0232e-05

min-lower-bound: 1.7358e-05

Load/2

max-upper-bound: 0.000499472

median-lower-bound: 0.000120076

median-upper-bound: 0.000480304

min-lower-bound: 0.00011984

Load/3

max-upper-bound: 0.000795552

median-lower-bound: 0.00019066

median-upper-bound: 0.00076264

min-lower-bound: 0.000189612

Load/4

max-upper-bound: 0.00097216

median-lower-bound: 0.00022585

median-upper-bound: 0.0009034

min-lower-bound: 0.000225152

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.324e-05

median-lower-bound: 1.319e-05

median-upper-bound: 5.276e-05

min-lower-bound: 1.311e-05

FullCache

max-upper-bound: 5.056e-05

median-lower-bound: 1.26e-05

median-upper-bound: 5.04e-05

min-lower-bound: 1.26e-05

HotCache

max-upper-bound: 5.012e-05

median-lower-bound: 1.253e-05

median-upper-bound: 5.012e-05

min-lower-bound: 1.253e-05

Load/1

max-upper-bound: 6.052e-05

median-lower-bound: 1.509e-05

median-upper-bound: 6.036e-05

min-lower-bound: 1.509e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOfI2eLDfIYc=

DirtyCache

max-upper-bound: 5.4416e-05

median-lower-bound: 1.2088e-05

median-upper-bound: 4.8352e-05

min-lower-bound: 1.1988e-05

FullCache

max-upper-bound: 2.9648e-05

median-lower-bound: 5.328e-06

median-upper-bound: 2.1312e-05

min-lower-bound: 5.298e-06

HotCache

max-upper-bound: 8.408e-06

median-lower-bound: 2.102e-06

median-upper-bound: 8.408e-06

min-lower-bound: 2.102e-06

Load/1

max-upper-bound: 5.7376e-05

median-lower-bound: 1.303e-05

median-upper-bound: 5.212e-05

min-lower-bound: 1.2932e-05

This requirement refines [spec:/rtems/barrier/req/perf-runtime](#).

5.2.6 spec:/rtems/barrier/req/perf-runtime

spec:/rtems/barrier/req/perf-runtime

The runtime of @ref RTEMSAPIClassicBarrier directives shall be measured.

rationale: N/A

This requirement refines *spec:/rtems/req/perf-runtime*.

This requirement refines *spec:/rtems/barrier/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/barrier/req/perf-release-auto*
- *spec:/rtems/barrier/req/perf-release-auto-other-cpu*
- *spec:/rtems/barrier/req/perf-release-manual*
- *spec:/rtems/barrier/req/perf-release-manual-preempt*

5.2.7 spec:/rtems/event/req/perf-isr-preempt

Let U be an interrupt service and V be a task which both execute on the same processor. Let B be a time point measured by U right before a call to rtems_event_send which unblocks task V which preempts the executing task. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while task V waits for events, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000102
median-lower-bound: 2.55e-05
median-upper-bound: 0.000102
min-lower-bound: 2.55e-05

FullCache

max-upper-bound: 0.000106
median-lower-bound: 2.6e-05
median-upper-bound: 0.000104
min-lower-bound: 2.55e-05

HotCache

max-upper-bound: 0.000106
median-lower-bound: 2.6e-05
median-upper-bound: 0.000104
min-lower-bound: 2.55e-05

Load/1

max-upper-bound: 0.000104
median-lower-bound: 2.6e-05
median-upper-bound: 0.000104
min-lower-bound: 2.6e-05

Load/2

max-upper-bound: 0.000104
median-lower-bound: 2.6e-05
median-upper-bound: 0.000104
min-lower-bound: 2.6e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000202
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 5e-05

FullCache

max-upper-bound: 0.0002
median-lower-bound: 2.35e-05
median-upper-bound: 9.4e-05
min-lower-bound: 2.35e-05

HotCache

max-upper-bound: 8.6e-05
median-lower-bound: 2.15e-05
median-upper-bound: 8.6e-05
min-lower-bound: 2.1e-05

Load/1

max-upper-bound: 0.0002

median-lower-bound: 4.95e-05

median-upper-bound: 0.000198

min-lower-bound: 4.95e-05

Load/2

max-upper-bound: 0.000244

median-lower-bound: 5.9e-05

median-upper-bound: 0.000236

min-lower-bound: 5.9e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000116

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.85e-05

FullCache

max-upper-bound: 0.000122

median-lower-bound: 1.4e-05

median-upper-bound: 5.6e-05

min-lower-bound: 1.35e-05

HotCache

max-upper-bound: 5e-05

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.25e-05

Load/1

max-upper-bound: 0.000116

median-lower-bound: 2.9e-05

median-upper-bound: 0.000116

min-lower-bound: 2.85e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 9.896e-05

median-lower-bound: 2.474e-05

median-upper-bound: 9.896e-05

min-lower-bound: 2.452e-05

FullCache

max-upper-bound: 9.904e-05

median-lower-bound: 2.476e-05

median-upper-bound: 9.904e-05

min-lower-bound: 2.472e-05

HotCache

max-upper-bound: 9.904e-05

median-lower-bound: 2.476e-05

median-upper-bound: 9.904e-05

min-lower-bound: 2.476e-05

Load/1

max-upper-bound: 9.92e-05
median-lower-bound: 2.48e-05
median-upper-bound: 9.92e-05
min-lower-bound: 2.454e-05

Load/2

max-upper-bound: 9.92e-05
median-lower-bound: 2.48e-05
median-upper-bound: 9.92e-05
min-lower-bound: 2.454e-05

Load/3

max-upper-bound: 9.92e-05
median-lower-bound: 2.48e-05
median-upper-bound: 9.92e-05
min-lower-bound: 2.454e-05

Load/4

max-upper-bound: 9.92e-05
median-lower-bound: 2.48e-05
median-upper-bound: 9.92e-05
min-lower-bound: 2.454e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 7.6408e-05
median-lower-bound: 1.7576e-05
median-upper-bound: 7.0304e-05
min-lower-bound: 1.749e-05

FullCache

max-upper-bound: 5.58e-05
median-lower-bound: 8.964e-06
median-upper-bound: 3.5856e-05
min-lower-bound: 8.904e-06

HotCache

max-upper-bound: 2.0048e-05

median-lower-bound: 4.846e-06

median-upper-bound: 1.9384e-05

min-lower-bound: 4.846e-06

Load/1

max-upper-bound: 0.000101528

median-lower-bound: 1.716e-05

median-upper-bound: 6.864e-05

min-lower-bound: 1.709e-05

Load/2

max-upper-bound: 0.00050236

median-lower-bound: 0.00012228

median-upper-bound: 0.00048912

min-lower-bound: 0.000122274

Load/3

max-upper-bound: 0.000795032

median-lower-bound: 0.000191836

median-upper-bound: 0.000767344

min-lower-bound: 0.00019179

Load/4

max-upper-bound: 0.0009774

median-lower-bound: 0.000232526

median-upper-bound: 0.000930104

min-lower-bound: 0.00023248

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.264e-05

median-lower-bound: 1.566e-05

median-upper-bound: 6.264e-05

min-lower-bound: 1.548e-05

FullCache

max-upper-bound: 6.224e-05

median-lower-bound: 1.556e-05

median-upper-bound: 6.224e-05

min-lower-bound: 1.544e-05

HotCache

max-upper-bound: 6.248e-05

median-lower-bound: 1.562e-05

median-upper-bound: 6.248e-05

min-lower-bound: 1.562e-05

Load/1

max-upper-bound: 6.264e-05

median-lower-bound: 1.566e-05

median-upper-bound: 6.264e-05

min-lower-bound: 1.548e-05

sparc/gr740/uni/qual-only/c1ZkB0sUlJ-siPi7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.008e-05

median-lower-bound: 1.1004e-05

median-upper-bound: 4.4016e-05

min-lower-bound: 1.0942e-05

FullCache

max-upper-bound: 3.4472e-05

median-lower-bound: 5.838e-06

median-upper-bound: 2.3352e-05

min-lower-bound: 5.806e-06

HotCache

max-upper-bound: 1.0104e-05

median-lower-bound: 2.526e-06

median-upper-bound: 1.0104e-05

min-lower-bound: 2.526e-06

Load/1

max-upper-bound: 4.832e-05

median-lower-bound: 1.1052e-05

median-upper-bound: 4.4208e-05

min-lower-bound: 1.0958e-05

This requirement refines [spec:/rtems/event/req/perf-runtime](#).

5.2.8 spec:/rtems/event/req/perf-other

While the execution environment is below environments, while a task waits for events, while the measurement sample is the runtime of exactly one successful call to rtems_event_send which does satisfy the event condition of the waiting task which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 4.8e-05
median-lower-bound: 1.2e-05
median-upper-bound: 4.8e-05
min-lower-bound: 1.15e-05

FullCache

max-upper-bound: 4.4e-05
median-lower-bound: 1.05e-05
median-upper-bound: 4.2e-05
min-lower-bound: 1.05e-05

HotCache

max-upper-bound: 4.4e-05
median-lower-bound: 1.05e-05
median-upper-bound: 4.2e-05
min-lower-bound: 1.05e-05

Load/1

max-upper-bound: 6e-05
median-lower-bound: 1.45e-05
median-upper-bound: 5.8e-05
min-lower-bound: 1.45e-05

Load/2

max-upper-bound: 6e-05
median-lower-bound: 1.5e-05
median-upper-bound: 6e-05
min-lower-bound: 1.45e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000124
median-lower-bound: 3.05e-05
median-upper-bound: 0.000122
min-lower-bound: 3e-05

FullCache

max-upper-bound: 8.4e-05
median-lower-bound: 9e-06
median-upper-bound: 3.6e-05
min-lower-bound: 9e-06

HotCache

max-upper-bound: 3.4e-05
median-lower-bound: 8.5e-06
median-upper-bound: 3.4e-05
min-lower-bound: 8e-06

Load/1

max-upper-bound: 0.000136
median-lower-bound: 3.35e-05
median-upper-bound: 0.000134
min-lower-bound: 3.35e-05

Load/2

max-upper-bound: 0.000154
median-lower-bound: 3.8e-05
median-upper-bound: 0.000152
min-lower-bound: 3.75e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 6.8e-05

median-lower-bound: 1.7e-05

median-upper-bound: 6.8e-05

min-lower-bound: 1.65e-05

FullCache

max-upper-bound: 4.6e-05

median-lower-bound: 4.5e-06

median-upper-bound: 1.8e-05

min-lower-bound: 4e-06

HotCache

max-upper-bound: 1.6e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 3.5e-06

Load/1

max-upper-bound: 8e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 1.95e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcl0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 4.46e-05

median-lower-bound: 1.109e-05

median-upper-bound: 4.436e-05

min-lower-bound: 1.101e-05

FullCache

max-upper-bound: 4.052e-05

median-lower-bound: 1.013e-05

median-upper-bound: 4.052e-05

min-lower-bound: 1.013e-05

HotCache

max-upper-bound: 4.056e-05

median-lower-bound: 1.014e-05

median-upper-bound: 4.056e-05

min-lower-bound: 1.014e-05

Load/1

max-upper-bound: 5.612e-05

median-lower-bound: 1.399e-05

median-upper-bound: 5.596e-05

min-lower-bound: 1.381e-05

Load/2

max-upper-bound: 5.612e-05

median-lower-bound: 1.399e-05

median-upper-bound: 5.596e-05

min-lower-bound: 1.381e-05

Load/3

max-upper-bound: 5.612e-05

median-lower-bound: 1.399e-05

median-upper-bound: 5.596e-05

min-lower-bound: 1.381e-05

Load/4

max-upper-bound: 5.612e-05

median-lower-bound: 1.399e-05

median-upper-bound: 5.596e-05

min-lower-bound: 1.381e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.0584e-05

median-lower-bound: 1.2128e-05

median-upper-bound: 4.8512e-05

min-lower-bound: 1.1526e-05

FullCache

max-upper-bound: 4.4608e-05

median-lower-bound: 3.976e-06

median-upper-bound: 1.5904e-05

min-lower-bound: 3.962e-06

HotCache

max-upper-bound: 9.376e-06

median-lower-bound: 2.156e-06

median-upper-bound: 8.624e-06

min-lower-bound: 2.156e-06

Load/1

max-upper-bound: 8.06e-05

median-lower-bound: 1.2864e-05

median-upper-bound: 5.1456e-05

min-lower-bound: 1.188e-05

Load/2

max-upper-bound: 0.00028116

median-lower-bound: 6.6552e-05

median-upper-bound: 0.000266208

min-lower-bound: 6.635e-05

Load/3

max-upper-bound: 0.000408136

median-lower-bound: 0.0001005

median-upper-bound: 0.000402

min-lower-bound: 9.8372e-05

Load/4

max-upper-bound: 0.000479504

median-lower-bound: 0.0001173

median-upper-bound: 0.0004692

min-lower-bound: 0.000114648

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 2.58e-05

median-lower-bound: 6.41e-06

median-upper-bound: 2.564e-05

min-lower-bound: 6.21e-06

FullCache

max-upper-bound: 2.124e-05

median-lower-bound: 5.31e-06

median-upper-bound: 2.124e-05

min-lower-bound: 5.31e-06

HotCache

max-upper-bound: 2.112e-05

median-lower-bound: 5.28e-06

median-upper-bound: 2.112e-05

min-lower-bound: 5.28e-06

Load/1

max-upper-bound: 3.74e-05

median-lower-bound: 9.31e-06

median-upper-bound: 3.724e-05

min-lower-bound: 9.23e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOfIz2eLDflYc=

DirtyCache

max-upper-bound: 2.9016e-05

median-lower-bound: 7.054e-06

median-upper-bound: 2.8216e-05

min-lower-bound: 6.994e-06

FullCache

max-upper-bound: 2.0568e-05

median-lower-bound: 2.712e-06

median-upper-bound: 1.0848e-05

min-lower-bound: 2.692e-06

HotCache

max-upper-bound: 4.024e-06

median-lower-bound: 1.006e-06

median-upper-bound: 4.024e-06

min-lower-bound: 1.006e-06

Load/1

max-upper-bound: 3.3432e-05

median-lower-bound: 8.27e-06

median-upper-bound: 3.308e-05

min-lower-bound: 8.204e-06

This requirement refines [spec:/rtems/event/req/perf-runtime](#).

5.2.9 spec:/rtems/event/req/perf-other-cpu

Let U and V be two tasks with distinct home schedulers. Let B be a time point measured by U right before a call to rtems_event_send which does satisfy the event condition of the waiting task V which is scheduled on another processor. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the task V waits for events, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8.2e-05

median-lower-bound: 2.05e-05

median-upper-bound: 8.2e-05

min-lower-bound: 2.05e-05

FullCache

max-upper-bound: 8.2e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 1.95e-05

HotCache

max-upper-bound: 8e-05

median-lower-bound: 1.95e-05

median-upper-bound: 7.8e-05

min-lower-bound: 1.95e-05

Load/1

max-upper-bound: 9.2e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.25e-05

Load/2

max-upper-bound: 9.2e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.2e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000218

median-lower-bound: 5.4e-05

median-upper-bound: 0.000216

min-lower-bound: 5.35e-05

FullCache

max-upper-bound: 0.000166

median-lower-bound: 1.8e-05

median-upper-bound: 7.2e-05

min-lower-bound: 1.8e-05

HotCache

max-upper-bound: 7.2e-05

median-lower-bound: 1.75e-05

median-upper-bound: 7e-05

min-lower-bound: 1.7e-05

Load/1

max-upper-bound: 0.000214

median-lower-bound: 5.35e-05

median-upper-bound: 0.000214

min-lower-bound: 5.3e-05

Load/2

max-upper-bound: 0.000244

median-lower-bound: 6.05e-05

median-upper-bound: 0.000242

min-lower-bound: 6e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.792e-05

median-lower-bound: 1.927e-05

median-upper-bound: 7.708e-05

min-lower-bound: 1.912e-05

FullCache

max-upper-bound: 7.66e-05

median-lower-bound: 1.902e-05

median-upper-bound: 7.608e-05

min-lower-bound: 1.844e-05

HotCache

max-upper-bound: 7.544e-05

median-lower-bound: 1.886e-05

median-upper-bound: 7.544e-05

min-lower-bound: 1.881e-05

Load/1

max-upper-bound: 8.528e-05

median-lower-bound: 2.127e-05

median-upper-bound: 8.508e-05

min-lower-bound: 2.092e-05

Load/2

max-upper-bound: 8.616e-05
median-lower-bound: 2.136e-05
median-upper-bound: 8.544e-05
min-lower-bound: 2.073e-05

Load/3

max-upper-bound: 8.548e-05
median-lower-bound: 2.132e-05
median-upper-bound: 8.528e-05
min-lower-bound: 2.089e-05

Load/4

max-upper-bound: 8.592e-05
median-lower-bound: 2.1e-05
median-upper-bound: 8.4e-05
min-lower-bound: 2.081e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 8.8912e-05
median-lower-bound: 1.8192e-05
median-upper-bound: 7.2768e-05
min-lower-bound: 1.7966e-05

FullCache

max-upper-bound: 9.0816e-05
median-lower-bound: 7.876e-06
median-upper-bound: 3.1504e-05
min-lower-bound: 7.806e-06

HotCache

max-upper-bound: 1.7104e-05
median-lower-bound: 4.134e-06
median-upper-bound: 1.6536e-05
min-lower-bound: 4.134e-06

Load/1

max-upper-bound: 0.000107288

median-lower-bound: 1.7154e-05

median-upper-bound: 6.8616e-05

min-lower-bound: 1.6958e-05

Load/2

max-upper-bound: 0.000369872

median-lower-bound: 9.2262e-05

median-upper-bound: 0.000369048

min-lower-bound: 9.2108e-05

Load/3

max-upper-bound: 0.000647064

median-lower-bound: 0.000157334

median-upper-bound: 0.000629336

min-lower-bound: 0.00015498

Load/4

max-upper-bound: 0.000817592

median-lower-bound: 0.00020107

median-upper-bound: 0.00080428

min-lower-bound: 0.000200576

This requirement refines [*spec:/rtems/event/req/perf-runtime*](#).

5.2.10 [*spec:/rtems/event/req/perf-other-not-satisfied*](#)

While the execution environment is below environments, while a task waits for events, while the measurement sample is the runtime of exactly one successful call to rtems_event_send which does not satisfy the event condition of the waiting task, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 1e-05

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 1.8e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 4e-06

Load/2

max-upper-bound: 1.8e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 4e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 4e-05

median-lower-bound: 9.5e-06

median-upper-bound: 3.8e-05

min-lower-bound: 9.5e-06

FullCache

max-upper-bound: 2.2e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 6e-05

median-lower-bound: 1.5e-05

median-upper-bound: 6e-05

min-lower-bound: 1.45e-05

Load/2

max-upper-bound: 6.2e-05

median-lower-bound: 1.55e-05

median-upper-bound: 6.2e-05

min-lower-bound: 1.5e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.4e-05
median-lower-bound: 5.5e-06
median-upper-bound: 2.2e-05
min-lower-bound: 5.5e-06

FullCache

max-upper-bound: 1.4e-05
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

Load/1

max-upper-bound: 3.4e-05
median-lower-bound: 8e-06
median-upper-bound: 3.2e-05
min-lower-bound: 8e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.76e-06
median-lower-bound: 1.9e-06
median-upper-bound: 7.6e-06
min-lower-bound: 1.86e-06

FullCache

max-upper-bound: 7.52e-06
median-lower-bound: 1.88e-06
median-upper-bound: 7.52e-06
min-lower-bound: 1.83e-06

HotCache

max-upper-bound: 7.48e-06
median-lower-bound: 1.87e-06
median-upper-bound: 7.48e-06
min-lower-bound: 1.87e-06

Load/1

max-upper-bound: 1.544e-05
median-lower-bound: 3.86e-06
median-upper-bound: 1.544e-05
min-lower-bound: 3.8e-06

Load/2

max-upper-bound: 1.544e-05
median-lower-bound: 3.86e-06
median-upper-bound: 1.544e-05
min-lower-bound: 3.8e-06

Load/3

max-upper-bound: 1.544e-05
median-lower-bound: 3.86e-06
median-upper-bound: 1.544e-05
min-lower-bound: 3.8e-06

Load/4

max-upper-bound: 1.544e-05
median-lower-bound: 3.86e-06
median-upper-bound: 1.544e-05
min-lower-bound: 3.8e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 1.8048e-05
median-lower-bound: 4.046e-06
median-upper-bound: 1.6184e-05
min-lower-bound: 3.83e-06

FullCache

max-upper-bound: 8.448e-06
median-lower-bound: 6.74e-07

median-upper-bound: 2.696e-06

min-lower-bound: 6.74e-07

HotCache

max-upper-bound: 2.184e-06

median-lower-bound: 4.22e-07

median-upper-bound: 1.688e-06

min-lower-bound: 4.22e-07

Load/1

max-upper-bound: 3.8352e-05

median-lower-bound: 5.642e-06

median-upper-bound: 2.2568e-05

min-lower-bound: 4.648e-06

Load/2

max-upper-bound: 6.9512e-05

median-lower-bound: 1.4056e-05

median-upper-bound: 5.6224e-05

min-lower-bound: 1.4008e-05

Load/3

max-upper-bound: 8.4144e-05

median-lower-bound: 1.9332e-05

median-upper-bound: 7.7328e-05

min-lower-bound: 1.89e-05

Load/4

max-upper-bound: 8.748e-05

median-lower-bound: 2.0526e-05

median-upper-bound: 8.2104e-05

min-lower-bound: 2.0224e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.56e-06

median-lower-bound: 1.56e-06

median-upper-bound: 6.24e-06

min-lower-bound: 1.56e-06

FullCache

max-upper-bound: 6.08e-06
median-lower-bound: 1.52e-06
median-upper-bound: 6.08e-06
min-lower-bound: 1.49e-06

HotCache

max-upper-bound: 6.12e-06
median-lower-bound: 1.53e-06
median-upper-bound: 6.12e-06
min-lower-bound: 1.53e-06

Load/1

max-upper-bound: 1.424e-05
median-lower-bound: 3.52e-06
median-upper-bound: 1.408e-05
min-lower-bound: 3.49e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFI2eLDfIYc=

DirtyCache

max-upper-bound: 1.0608e-05
median-lower-bound: 2.616e-06
median-upper-bound: 1.0464e-05
min-lower-bound: 2.582e-06

FullCache

max-upper-bound: 6.736e-06
median-lower-bound: 8.56e-07
median-upper-bound: 3.424e-06
min-lower-bound: 8.52e-07

HotCache

max-upper-bound: 1.272e-06
median-lower-bound: 3.18e-07
median-upper-bound: 1.272e-06
min-lower-bound: 3.18e-07

Load/1

max-upper-bound: 1.464e-05

median-lower-bound: 3.62e-06

median-upper-bound: 1.448e-05

min-lower-bound: 3.564e-06

This requirement refines [spec:/rtems/event/req/perf-runtime](#).

5.2.11 spec:/rtems/event/req/perf-other-preempt

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to rtems_event_send which does satisfy the event condition of the waiting task V which does preempt the caller. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the task V waits for events, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 7.8e-05

median-lower-bound: 1.95e-05

median-upper-bound: 7.8e-05

min-lower-bound: 1.9e-05

FullCache

max-upper-bound: 7.6e-05

median-lower-bound: 1.9e-05

median-upper-bound: 7.6e-05

min-lower-bound: 1.85e-05

HotCache

max-upper-bound: 7.6e-05

median-lower-bound: 1.9e-05

median-upper-bound: 7.6e-05

min-lower-bound: 1.85e-05

Load/1

max-upper-bound: 8.6e-05

median-lower-bound: 2.15e-05

median-upper-bound: 8.6e-05

min-lower-bound: 2.1e-05

Load/2

max-upper-bound: 8.6e-05

median-lower-bound: 2.15e-05

median-upper-bound: 8.6e-05

min-lower-bound: 2.15e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000188

median-lower-bound: 4.65e-05

median-upper-bound: 0.000186

min-lower-bound: 4.65e-05

FullCache

max-upper-bound: 0.00015

median-lower-bound: 1.6e-05

median-upper-bound: 6.4e-05

min-lower-bound: 1.55e-05

HotCache

max-upper-bound: 5.8e-05

median-lower-bound: 1.45e-05

median-upper-bound: 5.8e-05

min-lower-bound: 1.4e-05

Load/1

max-upper-bound: 0.000186

median-lower-bound: 4.6e-05

median-upper-bound: 0.000184

min-lower-bound: 4.6e-05

Load/2

max-upper-bound: 0.00022

median-lower-bound: 5.35e-05

median-upper-bound: 0.000214

min-lower-bound: 5.35e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000106
median-lower-bound: 2.6e-05
median-upper-bound: 0.000104
min-lower-bound: 2.6e-05

FullCache

max-upper-bound: 9.6e-05
median-lower-bound: 1.05e-05
median-upper-bound: 4.2e-05
min-lower-bound: 1e-05

HotCache

max-upper-bound: 3.8e-05

median-lower-bound: 9e-06

median-upper-bound: 3.6e-05

min-lower-bound: 9e-06

Load/1

max-upper-bound: 0.000112

median-lower-bound: 2.75e-05

median-upper-bound: 0.00011

min-lower-bound: 2.75e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.344e-05

median-lower-bound: 1.836e-05

median-upper-bound: 7.344e-05

min-lower-bound: 1.836e-05

FullCache

max-upper-bound: 7.2e-05

median-lower-bound: 1.792e-05

median-upper-bound: 7.168e-05

min-lower-bound: 1.784e-05

HotCache

max-upper-bound: 7.156e-05

median-lower-bound: 1.789e-05

median-upper-bound: 7.156e-05

min-lower-bound: 1.781e-05

Load/1

max-upper-bound: 8.188e-05

median-lower-bound: 2.047e-05

median-upper-bound: 8.188e-05

min-lower-bound: 2.043e-05

Load/2

max-upper-bound: 8.124e-05

median-lower-bound: 2.031e-05

median-upper-bound: 8.124e-05

min-lower-bound: 2.027e-05

Load/3

max-upper-bound: 8.188e-05

median-lower-bound: 2.047e-05

median-upper-bound: 8.188e-05

min-lower-bound: 2.043e-05

Load/4

max-upper-bound: 8.124e-05

median-lower-bound: 2.031e-05

median-upper-bound: 8.124e-05

min-lower-bound: 2.027e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 7.6112e-05

median-lower-bound: 1.7872e-05

median-upper-bound: 7.1488e-05

min-lower-bound: 1.6898e-05

FullCache

max-upper-bound: 9.2e-05

median-lower-bound: 7.59e-06

median-upper-bound: 3.036e-05

min-lower-bound: 7.54e-06

HotCache

max-upper-bound: 1.4784e-05

median-lower-bound: 3.634e-06

median-upper-bound: 1.4536e-05

min-lower-bound: 3.634e-06

Load/1

max-upper-bound: 9.9328e-05

median-lower-bound: 1.606e-05

median-upper-bound: 6.424e-05

min-lower-bound: 1.5634e-05

Load/2

max-upper-bound: 0.00044488

median-lower-bound: 0.000107426

median-upper-bound: 0.000429704

min-lower-bound: 0.000107254

Load/3

max-upper-bound: 0.000688336

median-lower-bound: 0.000168912

median-upper-bound: 0.000675648

min-lower-bound: 0.000167808

Load/4

max-upper-bound: 0.000835664

median-lower-bound: 0.000199834

median-upper-bound: 0.000799336

min-lower-bound: 0.000198884

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 4.852e-05

median-lower-bound: 1.209e-05

median-upper-bound: 4.836e-05

min-lower-bound: 1.181e-05

FullCache

max-upper-bound: 4.616e-05

median-lower-bound: 1.146e-05

median-upper-bound: 4.584e-05

min-lower-bound: 1.138e-05

HotCache

max-upper-bound: 4.572e-05

median-lower-bound: 1.143e-05

median-upper-bound: 4.572e-05

min-lower-bound: 1.135e-05

Load/1

max-upper-bound: 5.58e-05

median-lower-bound: 1.395e-05

median-upper-bound: 5.58e-05

min-lower-bound: 1.395e-05

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOfI2eLDfIYc=

DirtyCache

max-upper-bound: 4.304e-05

median-lower-bound: 1.022e-05

median-upper-bound: 4.088e-05

min-lower-bound: 1.016e-05

FullCache

max-upper-bound: 3.7024e-05

median-lower-bound: 5.126e-06

median-upper-bound: 2.0504e-05

min-lower-bound: 5.092e-06

HotCache

max-upper-bound: 7.632e-06

median-lower-bound: 1.908e-06

median-upper-bound: 7.632e-06

min-lower-bound: 1.908e-06

Load/1

max-upper-bound: 4.4704e-05

median-lower-bound: 1.0948e-05

median-upper-bound: 4.3792e-05

min-lower-bound: 1.0858e-05

This requirement refines [spec:/rtems/event/req/perf-runtime](#).

5.2.12 spec:/rtems/event/req/perf-runtime

spec:/rtems/event/req/perf-runtime

The runtime of @ref RTEMSAPIClassicEvent directives shall be measured.

rationale: N/A

This requirement refines [spec:/rtems/req/perf-runtime](#).

This requirement refines [spec:/rtems/event/req/group](#).

This requirement is refined by the following requirements:

- *spec:/rtems/event/req/perf-isr-preempt*
- *spec:/rtems/event/req/perf-other*
- *spec:/rtems/event/req/perf-other-cpu*
- *spec:/rtems/event/req/perf-other-not-satisfied*
- *spec:/rtems/event/req/perf-other-preempt*

5.2.13 spec:/rtems/message/req/perf-receive-try

While the execution environment is below environments, while a message queue is empty, while the measurement sample is the runtime of exactly one unsatisfied call to rtems_message_queue_send, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8e-06
median-lower-bound: 2e-06
median-upper-bound: 8e-06
min-lower-bound: 2e-06

FullCache

max-upper-bound: 8e-06
median-lower-bound: 2e-06
median-upper-bound: 8e-06
min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06
median-lower-bound: 2e-06
median-upper-bound: 8e-06
min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 1.8e-05
median-lower-bound: 4e-06
median-upper-bound: 1.6e-05

min-lower-bound: 3.5e-06

Load/2

max-upper-bound: 1.8e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 3.5e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 3.4e-05

median-lower-bound: 8.5e-06

median-upper-bound: 3.4e-05

min-lower-bound: 8e-06

FullCache

max-upper-bound: 1.6e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 5.8e-05

median-lower-bound: 1.4e-05

median-upper-bound: 5.6e-05

min-lower-bound: 1.4e-05

Load/2

max-upper-bound: 6e-05

median-lower-bound: 1.45e-05

median-upper-bound: 5.8e-05

min-lower-bound: 1.45e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.2e-05
median-lower-bound: 5e-06
median-upper-bound: 2e-05
min-lower-bound: 5e-06

FullCache

max-upper-bound: 1.6e-05
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 4e-06

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 5e-07

Load/1

max-upper-bound: 3.2e-05

median-lower-bound: 8e-06

median-upper-bound: 3.2e-05

min-lower-bound: 7.5e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.44e-06

median-lower-bound: 1.86e-06

median-upper-bound: 7.44e-06

min-lower-bound: 1.86e-06

FullCache

max-upper-bound: 7.16e-06

median-lower-bound: 1.74e-06

median-upper-bound: 6.96e-06

min-lower-bound: 1.74e-06

HotCache

max-upper-bound: 6.92e-06

median-lower-bound: 1.73e-06

median-upper-bound: 6.92e-06

min-lower-bound: 1.73e-06

Load/1

max-upper-bound: 1.48e-05

median-lower-bound: 3.7e-06

median-upper-bound: 1.48e-05

min-lower-bound: 3.66e-06

Load/2

max-upper-bound: 1.496e-05

median-lower-bound: 3.74e-06

median-upper-bound: 1.496e-05

min-lower-bound: 3.74e-06

Load/3

max-upper-bound: 1.48e-05

median-lower-bound: 3.7e-06

median-upper-bound: 1.48e-05

min-lower-bound: 3.66e-06

Load/4

max-upper-bound: 1.48e-05

median-lower-bound: 3.7e-06

median-upper-bound: 1.48e-05

min-lower-bound: 3.66e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 1.9624e-05

median-lower-bound: 4.534e-06

median-upper-bound: 1.8136e-05

min-lower-bound: 4.248e-06

FullCache

max-upper-bound: 1.296e-05

median-lower-bound: 9.64e-07

median-upper-bound: 3.856e-06

min-lower-bound: 9.64e-07

HotCache

max-upper-bound: 2.048e-06

median-lower-bound: 3.9e-07

median-upper-bound: 1.56e-06

min-lower-bound: 3.9e-07

Load/1

max-upper-bound: 4.0488e-05

median-lower-bound: 5.902e-06

median-upper-bound: 2.3608e-05

min-lower-bound: 4.858e-06

Load/2

max-upper-bound: 7.4216e-05
median-lower-bound: 1.6522e-05
median-upper-bound: 6.6088e-05
min-lower-bound: 1.6498e-05

Load/3

max-upper-bound: 8.9152e-05
median-lower-bound: 2.2162e-05
median-upper-bound: 8.8648e-05
min-lower-bound: 1.9336e-05

Load/4

max-upper-bound: 9.5696e-05
median-lower-bound: 2.307e-05
median-upper-bound: 9.228e-05
min-lower-bound: 2.1742e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.52e-06
median-lower-bound: 1.37e-06
median-upper-bound: 5.48e-06
min-lower-bound: 1.37e-06

FullCache

max-upper-bound: 5.44e-06
median-lower-bound: 1.34e-06
median-upper-bound: 5.36e-06
min-lower-bound: 1.3e-06

HotCache

max-upper-bound: 5.4e-06
median-lower-bound: 1.35e-06
median-upper-bound: 5.4e-06
min-lower-bound: 1.35e-06

Load/1

max-upper-bound: 1.324e-05
median-lower-bound: 3.31e-06

median-upper-bound: 1.324e-05

min-lower-bound: 3.22e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOfI2eLDfIYc=

DirtyCache

max-upper-bound: 1.1848e-05

median-lower-bound: 2.378e-06

median-upper-bound: 9.512e-06

min-lower-bound: 2.354e-06

FullCache

max-upper-bound: 1.0824e-05

median-lower-bound: 8.54e-07

median-upper-bound: 3.416e-06

min-lower-bound: 8.48e-07

HotCache

max-upper-bound: 1.096e-06

median-lower-bound: 2.74e-07

median-upper-bound: 1.096e-06

min-lower-bound: 2.74e-07

Load/1

max-upper-bound: 1.508e-05

median-lower-bound: 3.436e-06

median-upper-bound: 1.3744e-05

min-lower-bound: 3.394e-06

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.14 spec:/rtems/message/req/perf-receive-wait-forever

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to `rtems_message_queue_receive` which blocks on the message queue with no timeout. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while a message queue is empty, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 7.6e-05
median-lower-bound: 1.9e-05
median-upper-bound: 7.6e-05
min-lower-bound: 1.85e-05

FullCache

max-upper-bound: 7.4e-05
median-lower-bound: 1.85e-05
median-upper-bound: 7.4e-05
min-lower-bound: 1.8e-05

HotCache

max-upper-bound: 7.4e-05
median-lower-bound: 1.85e-05
median-upper-bound: 7.4e-05
min-lower-bound: 1.8e-05

Load/1

max-upper-bound: 8.8e-05
median-lower-bound: 2.2e-05
median-upper-bound: 8.8e-05
min-lower-bound: 2.15e-05

Load/2

max-upper-bound: 8.8e-05
median-lower-bound: 2.2e-05
median-upper-bound: 8.8e-05
min-lower-bound: 2.15e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000196
median-lower-bound: 4.85e-05
median-upper-bound: 0.000194
min-lower-bound: 4.85e-05

FullCache

max-upper-bound: 0.000152
median-lower-bound: 1.85e-05
median-upper-bound: 7.4e-05
min-lower-bound: 1.85e-05

HotCache

max-upper-bound: 7e-05
median-lower-bound: 1.75e-05
median-upper-bound: 7e-05
min-lower-bound: 1.7e-05

Load/1

max-upper-bound: 0.0002
median-lower-bound: 4.95e-05
median-upper-bound: 0.000198
min-lower-bound: 4.95e-05

Load/2

max-upper-bound: 0.000238
median-lower-bound: 5.8e-05
median-upper-bound: 0.000232
min-lower-bound: 5.75e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000116

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.85e-05

FullCache

max-upper-bound: 0.00011

median-lower-bound: 1.15e-05

median-upper-bound: 4.6e-05

min-lower-bound: 1.15e-05

HotCache

max-upper-bound: 4.6e-05

median-lower-bound: 1.15e-05

median-upper-bound: 4.6e-05

min-lower-bound: 1.1e-05

Load/1

max-upper-bound: 0.000122

median-lower-bound: 3.05e-05

median-upper-bound: 0.000122

min-lower-bound: 3e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.176e-05

median-lower-bound: 1.784e-05

median-upper-bound: 7.136e-05

min-lower-bound: 1.776e-05

FullCache

max-upper-bound: 6.868e-05

median-lower-bound: 1.717e-05

median-upper-bound: 6.868e-05

min-lower-bound: 1.717e-05

HotCache

max-upper-bound: 6.888e-05

median-lower-bound: 1.722e-05

median-upper-bound: 6.888e-05

min-lower-bound: 1.722e-05

Load/1

max-upper-bound: 8.356e-05

median-lower-bound: 2.077e-05

median-upper-bound: 8.308e-05

min-lower-bound: 2.069e-05

Load/2

max-upper-bound: 8.356e-05

median-lower-bound: 2.077e-05

median-upper-bound: 8.308e-05

min-lower-bound: 2.069e-05

Load/3

max-upper-bound: 8.356e-05

median-lower-bound: 2.077e-05

median-upper-bound: 8.308e-05

min-lower-bound: 2.069e-05

Load/4

max-upper-bound: 8.356e-05

median-lower-bound: 2.077e-05

median-upper-bound: 8.308e-05

min-lower-bound: 2.069e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 7.6848e-05
median-lower-bound: 1.8436e-05
median-upper-bound: 7.3744e-05
min-lower-bound: 1.7486e-05

FullCache

max-upper-bound: 7.9496e-05
median-lower-bound: 8.774e-06
median-upper-bound: 3.5096e-05
min-lower-bound: 8.734e-06

HotCache

max-upper-bound: 1.4528e-05
median-lower-bound: 3.56e-06
median-upper-bound: 1.424e-05
min-lower-bound: 3.56e-06

Load/1

max-upper-bound: 0.000120608
median-lower-bound: 1.7804e-05
median-upper-bound: 7.1216e-05
min-lower-bound: 1.7564e-05

Load/2

max-upper-bound: 0.00050312
median-lower-bound: 0.000119652
median-upper-bound: 0.000478608
min-lower-bound: 0.000119592

Load/3

max-upper-bound: 0.000780432
median-lower-bound: 0.000189674
median-upper-bound: 0.000758696
min-lower-bound: 0.000187986

Load/4

max-upper-bound: 0.000930864
median-lower-bound: 0.000223454

median-upper-bound: 0.000893816

min-lower-bound: 0.000223392

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.272e-05

median-lower-bound: 1.314e-05

median-upper-bound: 5.256e-05

min-lower-bound: 1.3e-05

FullCache

max-upper-bound: 5.036e-05

median-lower-bound: 1.259e-05

median-upper-bound: 5.036e-05

min-lower-bound: 1.255e-05

HotCache

max-upper-bound: 5.032e-05

median-lower-bound: 1.258e-05

median-upper-bound: 5.032e-05

min-lower-bound: 1.258e-05

Load/1

max-upper-bound: 6.028e-05

median-lower-bound: 1.503e-05

median-upper-bound: 6.012e-05

min-lower-bound: 1.491e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.2592e-05

median-lower-bound: 1.1908e-05

median-upper-bound: 4.7632e-05

min-lower-bound: 1.1862e-05

FullCache

max-upper-bound: 4.8024e-05

median-lower-bound: 5.576e-06

median-upper-bound: 2.2304e-05

min-lower-bound: 5.56e-06

HotCache

max-upper-bound: 8.816e-06

median-lower-bound: 2.204e-06

median-upper-bound: 8.816e-06

min-lower-bound: 2.204e-06

Load/1

max-upper-bound: 5.6224e-05

median-lower-bound: 1.2816e-05

median-upper-bound: 5.1264e-05

min-lower-bound: 1.2756e-05

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.15 spec:/rtems/message/req/perf-receive-wait-timed

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to `rtems_message_queue_receive` which blocks on the message queue with a timeout. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while a message queue is empty, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

`sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=`

DirtyCache

max-upper-bound: 8.4e-05

median-lower-bound: 2.05e-05

median-upper-bound: 8.2e-05

min-lower-bound: 2.05e-05

FullCache

max-upper-bound: 8.2e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 1.95e-05

HotCache

max-upper-bound: 8e-05
median-lower-bound: 2e-05
median-upper-bound: 8e-05
min-lower-bound: 1.95e-05

Load/1

max-upper-bound: 9.6e-05
median-lower-bound: 2.35e-05
median-upper-bound: 9.4e-05
min-lower-bound: 2.3e-05

Load/2

max-upper-bound: 9.6e-05
median-lower-bound: 2.35e-05
median-upper-bound: 9.4e-05
min-lower-bound: 2.35e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.00021
median-lower-bound: 5.2e-05
median-upper-bound: 0.000208
min-lower-bound: 5.15e-05

FullCache

max-upper-bound: 0.000156
median-lower-bound: 2.1e-05
median-upper-bound: 8.4e-05
min-lower-bound: 2.05e-05

HotCache

max-upper-bound: 7.8e-05
median-lower-bound: 1.95e-05
median-upper-bound: 7.8e-05
min-lower-bound: 1.95e-05

Load/1

max-upper-bound: 0.000212
median-lower-bound: 5.25e-05

median-upper-bound: 0.00021

min-lower-bound: 5.25e-05

Load/2

max-upper-bound: 0.000256

median-lower-bound: 6.2e-05

median-upper-bound: 0.000248

min-lower-bound: 6.2e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000126

median-lower-bound: 3.1e-05

median-upper-bound: 0.000124

min-lower-bound: 3.1e-05

FullCache

max-upper-bound: 0.00011

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.2e-05

HotCache

max-upper-bound: 5e-05

median-lower-bound: 1.2e-05

median-upper-bound: 4.8e-05

min-lower-bound: 1.2e-05

Load/1

max-upper-bound: 0.000132

median-lower-bound: 3.3e-05

median-upper-bound: 0.000132

min-lower-bound: 3.25e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.736e-05

median-lower-bound: 1.934e-05

median-upper-bound: 7.736e-05

min-lower-bound: 1.934e-05

FullCache

max-upper-bound: 7.604e-05

median-lower-bound: 1.891e-05

median-upper-bound: 7.564e-05

min-lower-bound: 1.889e-05

HotCache

max-upper-bound: 7.624e-05

median-lower-bound: 1.906e-05

median-upper-bound: 7.624e-05

min-lower-bound: 1.902e-05

Load/1

max-upper-bound: 9.02e-05
median-lower-bound: 2.255e-05
median-upper-bound: 9.02e-05
min-lower-bound: 2.243e-05

Load/2

max-upper-bound: 9.004e-05
median-lower-bound: 2.251e-05
median-upper-bound: 9.004e-05
min-lower-bound: 2.223e-05

Load/3

max-upper-bound: 9.02e-05
median-lower-bound: 2.255e-05
median-upper-bound: 9.02e-05
min-lower-bound: 2.243e-05

Load/4

max-upper-bound: 9.02e-05
median-lower-bound: 2.255e-05
median-upper-bound: 9.02e-05
min-lower-bound: 2.243e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 8.3288e-05
median-lower-bound: 1.9524e-05
median-upper-bound: 7.8096e-05
min-lower-bound: 1.872e-05

FullCache

max-upper-bound: 8.4776e-05
median-lower-bound: 9.438e-06
median-upper-bound: 3.7752e-05
min-lower-bound: 9.394e-06

HotCache

max-upper-bound: 1.6472e-05
median-lower-bound: 4.018e-06

median-upper-bound: 1.6072e-05

min-lower-bound: 4.018e-06

Load/1

max-upper-bound: 0.000124184

median-lower-bound: 1.904e-05

median-upper-bound: 7.616e-05

min-lower-bound: 1.864e-05

Load/2

max-upper-bound: 0.000534784

median-lower-bound: 0.000127222

median-upper-bound: 0.000508888

min-lower-bound: 0.000127082

Load/3

max-upper-bound: 0.000828448

median-lower-bound: 0.000202946

median-upper-bound: 0.000811784

min-lower-bound: 0.00019972

Load/4

max-upper-bound: 0.000987144

median-lower-bound: 0.000238754

median-upper-bound: 0.000955016

min-lower-bound: 0.000238604

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.676e-05

median-lower-bound: 1.411e-05

median-upper-bound: 5.644e-05

min-lower-bound: 1.401e-05

FullCache

max-upper-bound: 5.416e-05

median-lower-bound: 1.354e-05

median-upper-bound: 5.416e-05

min-lower-bound: 1.354e-05

HotCache

max-upper-bound: 5.436e-05
median-lower-bound: 1.359e-05
median-upper-bound: 5.436e-05
min-lower-bound: 1.359e-05

Load/1

max-upper-bound: 6.456e-05
median-lower-bound: 1.61e-05
median-upper-bound: 6.44e-05
min-lower-bound: 1.598e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.2896e-05
median-lower-bound: 1.32e-05
median-upper-bound: 5.28e-05
min-lower-bound: 1.3192e-05

FullCache

max-upper-bound: 4.8104e-05
median-lower-bound: 6.236e-06
median-upper-bound: 2.4944e-05
min-lower-bound: 6.196e-06

HotCache

max-upper-bound: 9.76e-06
median-lower-bound: 2.41e-06
median-upper-bound: 9.64e-06
min-lower-bound: 2.41e-06

Load/1

max-upper-bound: 5.712e-05
median-lower-bound: 1.414e-05
median-upper-bound: 5.656e-05
min-lower-bound: 1.4086e-05

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.16 spec:/rtems/message/req/perf-runtime

spec:/rtems/message/req/perf-runtime

The runtime of @ref RTEMSAPIClassicMessage directives shall be measured.

rationale: N/A

This requirement refines *spec:/rtems/req/perf-runtime*.

This requirement refines *spec:/rtems/message/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/message/req/perf-receive-try*
- *spec:/rtems/message/req/perf-receive-wait-forever*
- *spec:/rtems/message/req/perf-receive-wait-timed*
- *spec:/rtems/message/req/perf-send*
- *spec:/rtems/message/req/perf-send-other*
- *spec:/rtems/message/req/perf-send-other-cpu*
- *spec:/rtems/message/req/perf-send-preempt*

5.2.17 spec:/rtems/message/req/perf-send

While the execution environment is below environments, while a message queue is empty, while no task waits on the message queue, while the measurement sample is the runtime of exactly one successful call to rtems_message_queue_send, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 1.4e-05
median-lower-bound: 3.5e-06
median-upper-bound: 1.4e-05
min-lower-bound: 3e-06

FullCache

max-upper-bound: 1.4e-05
median-lower-bound: 3e-06
median-upper-bound: 1.2e-05

min-lower-bound: 3e-06

HotCache

max-upper-bound: 1.4e-05

median-lower-bound: 3e-06

median-upper-bound: 1.2e-05

min-lower-bound: 3e-06

Load/1

max-upper-bound: 2.6e-05

median-lower-bound: 6.5e-06

median-upper-bound: 2.6e-05

min-lower-bound: 6e-06

Load/2

max-upper-bound: 2.8e-05

median-lower-bound: 6.5e-06

median-upper-bound: 2.6e-05

min-lower-bound: 6e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 6e-05

median-lower-bound: 1.45e-05

median-upper-bound: 5.8e-05

min-lower-bound: 1.45e-05

FullCache

max-upper-bound: 3e-05

median-lower-bound: 2.5e-06

median-upper-bound: 1e-05

min-lower-bound: 2e-06

HotCache

max-upper-bound: 1e-05

median-lower-bound: 2.5e-06

median-upper-bound: 1e-05

min-lower-bound: 2e-06

Load/1

max-upper-bound: 7.6e-05
median-lower-bound: 1.85e-05
median-upper-bound: 7.4e-05
min-lower-bound: 1.8e-05

Load/2

max-upper-bound: 8.2e-05
median-lower-bound: 2e-05
median-upper-bound: 8e-05
min-lower-bound: 2e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574IY=

DirtyCache

max-upper-bound: 3.4e-05

median-lower-bound: 8e-06

median-upper-bound: 3.2e-05

min-lower-bound: 8e-06

FullCache

max-upper-bound: 2.2e-05

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 2e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 4.8e-05

median-lower-bound: 1.2e-05

median-upper-bound: 4.8e-05

min-lower-bound: 1.15e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 1.276e-05

median-lower-bound: 3.19e-06

median-upper-bound: 1.276e-05

min-lower-bound: 3.19e-06

FullCache

max-upper-bound: 1.268e-05

median-lower-bound: 3.11e-06

median-upper-bound: 1.244e-05

min-lower-bound: 3.1e-06

HotCache

max-upper-bound: 1.264e-05

median-lower-bound: 3.16e-06

median-upper-bound: 1.264e-05

min-lower-bound: 3.12e-06

Load/1

max-upper-bound: 2.436e-05

median-lower-bound: 6.09e-06

median-upper-bound: 2.436e-05

min-lower-bound: 5.97e-06

Load/2

max-upper-bound: 2.436e-05

median-lower-bound: 6.09e-06

median-upper-bound: 2.436e-05

min-lower-bound: 6.01e-06

Load/3

max-upper-bound: 2.436e-05

median-lower-bound: 6.09e-06

median-upper-bound: 2.436e-05

min-lower-bound: 5.97e-06

Load/4

max-upper-bound: 2.436e-05

median-lower-bound: 6.09e-06

median-upper-bound: 2.436e-05

min-lower-bound: 5.97e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 2.5304e-05

median-lower-bound: 6.052e-06

median-upper-bound: 2.4208e-05

min-lower-bound: 5.572e-06

FullCache

max-upper-bound: 1.6616e-05

median-lower-bound: 1.186e-06

median-upper-bound: 4.744e-06

min-lower-bound: 1.186e-06

HotCache

max-upper-bound: 2.944e-06
median-lower-bound: 6.12e-07
median-upper-bound: 2.448e-06
min-lower-bound: 6.12e-07

Load/1

max-upper-bound: 4.8064e-05
median-lower-bound: 7.092e-06
median-upper-bound: 2.8368e-05
min-lower-bound: 6.182e-06

Load/2

max-upper-bound: 0.000119096
median-lower-bound: 2.8766e-05
median-upper-bound: 0.000115064
min-lower-bound: 2.7478e-05

Load/3

max-upper-bound: 0.000168512
median-lower-bound: 4.1306e-05
median-upper-bound: 0.000165224
min-lower-bound: 3.884e-05

Load/4

max-upper-bound: 0.000194432
median-lower-bound: 4.8126e-05
median-upper-bound: 0.000192504
min-lower-bound: 4.3936e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 1.1e-05
median-lower-bound: 2.75e-06
median-upper-bound: 1.1e-05
min-lower-bound: 2.71e-06

FullCache

max-upper-bound: 1.076e-05
median-lower-bound: 2.65e-06

median-upper-bound: 1.06e-05

min-lower-bound: 2.61e-06

HotCache

max-upper-bound: 1.056e-05

median-lower-bound: 2.64e-06

median-upper-bound: 1.056e-05

min-lower-bound: 2.64e-06

Load/1

max-upper-bound: 2.292e-05

median-lower-bound: 5.67e-06

median-upper-bound: 2.268e-05

min-lower-bound: 5.63e-06

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 1.5312e-05

median-lower-bound: 3.802e-06

median-upper-bound: 1.5208e-05

min-lower-bound: 3.746e-06

FullCache

max-upper-bound: 1.2152e-05

median-lower-bound: 1.072e-06

median-upper-bound: 4.288e-06

min-lower-bound: 1.068e-06

HotCache

max-upper-bound: 1.984e-06

median-lower-bound: 4.96e-07

median-upper-bound: 1.984e-06

min-lower-bound: 4.96e-07

Load/1

max-upper-bound: 2.148e-05

median-lower-bound: 5.298e-06

median-upper-bound: 2.1192e-05

min-lower-bound: 5.264e-06

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.18 spec:/rtems/message/req/perf-send-other

While the execution environment is below environments, while a message queue is empty, while exactly one task waits on the message queue, while the measurement sample is the runtime of exactly one successful call to rtems_message_queue_send which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 6e-05

median-lower-bound: 1.5e-05

median-upper-bound: 6e-05

min-lower-bound: 1.45e-05

FullCache

max-upper-bound: 5.6e-05

median-lower-bound: 1.35e-05

median-upper-bound: 5.4e-05

min-lower-bound: 1.35e-05

HotCache

max-upper-bound: 5.6e-05

median-lower-bound: 1.35e-05

median-upper-bound: 5.4e-05

min-lower-bound: 1.35e-05

Load/1

max-upper-bound: 7.2e-05

median-lower-bound: 1.75e-05

median-upper-bound: 7e-05

min-lower-bound: 1.75e-05

Load/2

max-upper-bound: 7.2e-05

median-lower-bound: 1.8e-05

median-upper-bound: 7.2e-05

min-lower-bound: 1.75e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574IY=

DirtyCache

max-upper-bound: 0.000144

median-lower-bound: 3.55e-05

median-upper-bound: 0.000142

min-lower-bound: 3.55e-05

FullCache

max-upper-bound: 7.8e-05

median-lower-bound: 1.6e-05

median-upper-bound: 6.4e-05

min-lower-bound: 1.55e-05

HotCache

max-upper-bound: 6.2e-05

median-lower-bound: 1.5e-05

median-upper-bound: 6e-05

min-lower-bound: 1.5e-05

Load/1

max-upper-bound: 0.000156

median-lower-bound: 3.85e-05

median-upper-bound: 0.000154

min-lower-bound: 3.8e-05

Load/2

max-upper-bound: 0.000174

median-lower-bound: 4.35e-05

median-upper-bound: 0.000174

min-lower-bound: 4.3e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 8.2e-05
median-lower-bound: 2.05e-05
median-upper-bound: 8.2e-05
min-lower-bound: 2e-05

FullCache

max-upper-bound: 4.6e-05
median-lower-bound: 6.5e-06
median-upper-bound: 2.6e-05
min-lower-bound: 6.5e-06

HotCache

max-upper-bound: 2.4e-05
median-lower-bound: 6e-06
median-upper-bound: 2.4e-05
min-lower-bound: 5.5e-06

Load/1

max-upper-bound: 9.4e-05

median-lower-bound: 2.35e-05

median-upper-bound: 9.4e-05

min-lower-bound: 2.3e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcl0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.588e-05

median-lower-bound: 1.397e-05

median-upper-bound: 5.588e-05

min-lower-bound: 1.397e-05

FullCache

max-upper-bound: 5.228e-05

median-lower-bound: 1.305e-05

median-upper-bound: 5.22e-05

min-lower-bound: 1.289e-05

HotCache

max-upper-bound: 5.2e-05

median-lower-bound: 1.3e-05

median-upper-bound: 5.2e-05

min-lower-bound: 1.292e-05

Load/1

max-upper-bound: 6.732e-05

median-lower-bound: 1.683e-05

median-upper-bound: 6.732e-05

min-lower-bound: 1.671e-05

Load/2

max-upper-bound: 6.732e-05

median-lower-bound: 1.683e-05

median-upper-bound: 6.732e-05

min-lower-bound: 1.671e-05

Load/3

max-upper-bound: 6.804e-05

median-lower-bound: 1.701e-05

median-upper-bound: 6.804e-05

min-lower-bound: 1.701e-05

Load/4

max-upper-bound: 6.804e-05

median-lower-bound: 1.701e-05

median-upper-bound: 6.804e-05

min-lower-bound: 1.701e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.772e-05

median-lower-bound: 1.3486e-05

median-upper-bound: 5.3944e-05

min-lower-bound: 1.2942e-05

FullCache

max-upper-bound: 4.2232e-05

median-lower-bound: 5.844e-06

median-upper-bound: 2.3376e-05

min-lower-bound: 5.83e-06

HotCache

max-upper-bound: 1.1944e-05

median-lower-bound: 2.804e-06

median-upper-bound: 1.1216e-05

min-lower-bound: 2.804e-06

Load/1

max-upper-bound: 9.0672e-05

median-lower-bound: 1.4414e-05

median-upper-bound: 5.7656e-05

min-lower-bound: 1.3466e-05

Load/2

max-upper-bound: 0.000326056

median-lower-bound: 8.0318e-05

median-upper-bound: 0.000321272

min-lower-bound: 8.0088e-05

Load/3

max-upper-bound: 0.000494512
median-lower-bound: 0.000123134
median-upper-bound: 0.000492536
min-lower-bound: 0.000120764

Load/4

max-upper-bound: 0.000580544
median-lower-bound: 0.000144162
median-upper-bound: 0.000576648
min-lower-bound: 0.000141978

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 3.492e-05
median-lower-bound: 8.67e-06
median-upper-bound: 3.468e-05
min-lower-bound: 8.47e-06

FullCache

max-upper-bound: 3.056e-05
median-lower-bound: 7.64e-06
median-upper-bound: 3.056e-05
min-lower-bound: 7.64e-06

HotCache

max-upper-bound: 3.012e-05
median-lower-bound: 7.53e-06
median-upper-bound: 3.012e-05
min-lower-bound: 7.53e-06

Load/1

max-upper-bound: 4.684e-05
median-lower-bound: 1.155e-05
median-upper-bound: 4.62e-05
min-lower-bound: 1.153e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 3.3184e-05

median-lower-bound: 8.216e-06

median-upper-bound: 3.2864e-05

min-lower-bound: 8.18e-06

FullCache

max-upper-bound: 2.332e-05

median-lower-bound: 3.608e-06

median-upper-bound: 1.4432e-05

min-lower-bound: 3.598e-06

HotCache

max-upper-bound: 5.424e-06

median-lower-bound: 1.356e-06

median-upper-bound: 5.424e-06

min-lower-bound: 1.356e-06

Load/1

max-upper-bound: 3.784e-05

median-lower-bound: 9.392e-06

median-upper-bound: 3.7568e-05

min-lower-bound: 9.35e-06

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.19 spec:/rtems/message/req/perf-send-other-cpu

Let U and V be two tasks with distinct home schedulers. Let B be a time point measured by U right before a call to rtems_message_queue_send which unblocks the waiting task V which is scheduled on another processor. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while a message queue is empty, while only task V waits on the message queue, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 9.4e-05

median-lower-bound: 2.35e-05

median-upper-bound: 9.4e-05

min-lower-bound: 2.25e-05

FullCache

max-upper-bound: 9e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.2e-05

HotCache

max-upper-bound: 9.2e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.25e-05

Load/1

max-upper-bound: 0.000102

median-lower-bound: 2.55e-05

median-upper-bound: 0.000102

min-lower-bound: 2.5e-05

Load/2

max-upper-bound: 0.000102

median-lower-bound: 2.5e-05

median-upper-bound: 0.0001

min-lower-bound: 2.45e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000238

median-lower-bound: 5.9e-05

median-upper-bound: 0.000236

min-lower-bound: 5.9e-05

FullCache

max-upper-bound: 0.000112

median-lower-bound: 2.1e-05

median-upper-bound: 8.4e-05

min-lower-bound: 2.05e-05

HotCache

max-upper-bound: 8.2e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 2e-05

Load/1

max-upper-bound: 0.000234

median-lower-bound: 5.8e-05

median-upper-bound: 0.000232

min-lower-bound: 5.8e-05

Load/2

max-upper-bound: 0.000268

median-lower-bound: 6.6e-05

median-upper-bound: 0.000264

min-lower-bound: 6.6e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.768e-05

median-lower-bound: 2.181e-05

median-upper-bound: 8.724e-05

min-lower-bound: 2.172e-05

FullCache

max-upper-bound: 8.576e-05

median-lower-bound: 2.14e-05

median-upper-bound: 8.56e-05

min-lower-bound: 2.132e-05

HotCache

max-upper-bound: 8.476e-05

median-lower-bound: 2.119e-05

median-upper-bound: 8.476e-05

min-lower-bound: 2.117e-05

Load/1

max-upper-bound: 9.576e-05

median-lower-bound: 2.387e-05

median-upper-bound: 9.548e-05

min-lower-bound: 2.334e-05

Load/2

max-upper-bound: 9.584e-05

median-lower-bound: 2.375e-05

median-upper-bound: 9.5e-05

min-lower-bound: 2.321e-05

Load/3

max-upper-bound: 9.584e-05

median-lower-bound: 2.375e-05

median-upper-bound: 9.5e-05

min-lower-bound: 2.321e-05

Load/4

max-upper-bound: 9.576e-05

median-lower-bound: 2.386e-05

median-upper-bound: 9.544e-05

min-lower-bound: 2.317e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 9.4624e-05

median-lower-bound: 2.0306e-05

median-upper-bound: 8.1224e-05

min-lower-bound: 1.9922e-05

FullCache

max-upper-bound: 5.436e-05

median-lower-bound: 9.22e-06

median-upper-bound: 3.688e-05

min-lower-bound: 9.106e-06

HotCache

max-upper-bound: 1.9776e-05

median-lower-bound: 4.776e-06

median-upper-bound: 1.9104e-05

min-lower-bound: 4.776e-06

Load/1

max-upper-bound: 0.000108176

median-lower-bound: 1.9106e-05

median-upper-bound: 7.6424e-05

min-lower-bound: 1.8912e-05

Load/2

max-upper-bound: 0.00043296

median-lower-bound: 0.00010756

median-upper-bound: 0.00043024

min-lower-bound: 0.000107404

Load/3

max-upper-bound: 0.000766288

median-lower-bound: 0.000187162

median-upper-bound: 0.000748648

min-lower-bound: 0.000185116

Load/4

max-upper-bound: 0.000961368

median-lower-bound: 0.000237486

median-upper-bound: 0.000949944

min-lower-bound: 0.000237038

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.20 spec:/rtems/message/req/perf-send-preempt

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to rtems_message_queue_send which unblocks the waiting task V which does preempt the caller. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while a message queue is empty, while only task V waits on the message queue, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 9e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.2e-05

FullCache

max-upper-bound: 8.8e-05

median-lower-bound: 2.2e-05

median-upper-bound: 8.8e-05

min-lower-bound: 2.15e-05

HotCache

max-upper-bound: 8.6e-05

median-lower-bound: 2.15e-05

median-upper-bound: 8.6e-05

min-lower-bound: 2.15e-05

Load/1

max-upper-bound: 9.8e-05

median-lower-bound: 2.45e-05

median-upper-bound: 9.8e-05

min-lower-bound: 2.4e-05

Load/2

max-upper-bound: 9.8e-05

median-lower-bound: 2.45e-05

median-upper-bound: 9.8e-05

min-lower-bound: 2.45e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.00021

median-lower-bound: 5.2e-05

median-upper-bound: 0.000208

min-lower-bound: 5.2e-05

FullCache

max-upper-bound: 0.000102

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 2e-05

HotCache

max-upper-bound: 7.4e-05

median-lower-bound: 1.85e-05

median-upper-bound: 7.4e-05

min-lower-bound: 1.8e-05

Load/1

max-upper-bound: 0.000204

median-lower-bound: 5.05e-05

median-upper-bound: 0.000202

min-lower-bound: 5.05e-05

Load/2

max-upper-bound: 0.00024

median-lower-bound: 5.95e-05

median-upper-bound: 0.000238

min-lower-bound: 5.9e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000116

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.85e-05

FullCache

max-upper-bound: 6.4e-05

median-lower-bound: 1.15e-05

median-upper-bound: 4.6e-05

min-lower-bound: 1.1e-05

HotCache

max-upper-bound: 4.2e-05

median-lower-bound: 1e-05

median-upper-bound: 4e-05

min-lower-bound: 1e-05

Load/1

max-upper-bound: 0.000122

median-lower-bound: 3.05e-05

median-upper-bound: 0.000122

min-lower-bound: 3e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.54e-05

median-lower-bound: 2.135e-05

median-upper-bound: 8.54e-05

min-lower-bound: 2.111e-05

FullCache

max-upper-bound: 8.344e-05

median-lower-bound: 2.082e-05

median-upper-bound: 8.328e-05

min-lower-bound: 2.072e-05

HotCache

max-upper-bound: 8.348e-05

median-lower-bound: 2.087e-05

median-upper-bound: 8.348e-05

min-lower-bound: 2.087e-05

Load/1

max-upper-bound: 9.264e-05

median-lower-bound: 2.316e-05

median-upper-bound: 9.264e-05

min-lower-bound: 2.316e-05

Load/2

max-upper-bound: 9.392e-05

median-lower-bound: 2.348e-05

median-upper-bound: 9.392e-05

min-lower-bound: 2.348e-05

Load/3

max-upper-bound: 9.376e-05

median-lower-bound: 2.344e-05

median-upper-bound: 9.376e-05

min-lower-bound: 2.344e-05

Load/4

max-upper-bound: 9.376e-05

median-lower-bound: 2.344e-05

median-upper-bound: 9.376e-05

min-lower-bound: 2.344e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFlZ2eLDflYc=

DirtyCache

max-upper-bound: 8.4272e-05

median-lower-bound: 1.9424e-05

median-upper-bound: 7.7696e-05

min-lower-bound: 1.8548e-05

FullCache

max-upper-bound: 5.0152e-05

median-lower-bound: 8.968e-06

median-upper-bound: 3.5872e-05

min-lower-bound: 8.934e-06

HotCache

max-upper-bound: 1.6576e-05

median-lower-bound: 4.144e-06

median-upper-bound: 1.6576e-05

min-lower-bound: 4.144e-06

Load/1

max-upper-bound: 0.000103112

median-lower-bound: 1.7594e-05

median-upper-bound: 7.0376e-05

min-lower-bound: 1.7324e-05

Load/2

max-upper-bound: 0.000494264

median-lower-bound: 0.000121442

median-upper-bound: 0.000485768

min-lower-bound: 0.000121322

Load/3

max-upper-bound: 0.000780928

median-lower-bound: 0.00019489

median-upper-bound: 0.00077956

min-lower-bound: 0.000191446

Load/4

max-upper-bound: 0.000935568

median-lower-bound: 0.00023072

median-upper-bound: 0.00092288

min-lower-bound: 0.00022803

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.42e-05
median-lower-bound: 1.345e-05
median-upper-bound: 5.38e-05
min-lower-bound: 1.333e-05

FullCache

max-upper-bound: 5.192e-05
median-lower-bound: 1.298e-05
median-upper-bound: 5.192e-05
min-lower-bound: 1.294e-05

HotCache

max-upper-bound: 5.172e-05
median-lower-bound: 1.293e-05
median-upper-bound: 5.172e-05
min-lower-bound: 1.293e-05

Load/1

max-upper-bound: 6.2e-05
median-lower-bound: 1.538e-05
median-upper-bound: 6.152e-05
min-lower-bound: 1.534e-05

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 4.7744e-05
median-lower-bound: 1.146e-05
median-upper-bound: 4.584e-05
min-lower-bound: 1.137e-05

FullCache

max-upper-bound: 3.1288e-05
median-lower-bound: 6.03e-06
median-upper-bound: 2.412e-05
min-lower-bound: 6.024e-06

HotCache

max-upper-bound: 8.568e-06
median-lower-bound: 2.142e-06

median-upper-bound: 8.568e-06

min-lower-bound: 2.142e-06

Load/1

max-upper-bound: 4.948e-05

median-lower-bound: 1.2076e-05

median-upper-bound: 4.8304e-05

min-lower-bound: 1.1982e-05

This requirement refines [spec:/rtems/message/req/perf-runtime](#).

5.2.21 spec:/rtems/part/req/perf-get-buffer

While the execution environment is below environments, while a partition has exactly 100 free buffers, while the measurement sample is the runtime of exactly one successful call to rtems_partition_get_buffer, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 8e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 1.4e-05

median-lower-bound: 3.5e-06

median-upper-bound: 1.4e-05

min-lower-bound: 3e-06

Load/2

max-upper-bound: 1.4e-05

median-lower-bound: 3e-06

median-upper-bound: 1.2e-05

min-lower-bound: 3e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574IY=

DirtyCache

max-upper-bound: 2.8e-05

median-lower-bound: 6.5e-06

median-upper-bound: 2.6e-05

min-lower-bound: 6.5e-06

FullCache

max-upper-bound: 1.2e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 5e-05

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.2e-05

Load/2

max-upper-bound: 5e-05

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.2e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 1.8e-05

median-lower-bound: 4.5e-06

median-upper-bound: 1.8e-05

min-lower-bound: 4e-06

FullCache

max-upper-bound: 1e-05

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 1e-06

HotCache

max-upper-bound: 4e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 5e-07

Load/1

max-upper-bound: 2.6e-05
median-lower-bound: 6.5e-06
median-upper-bound: 2.6e-05
min-lower-bound: 6e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.32e-06
median-lower-bound: 1.58e-06
median-upper-bound: 6.32e-06
min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 6.2e-06
median-lower-bound: 1.48e-06
median-upper-bound: 5.92e-06
min-lower-bound: 1.48e-06

HotCache

max-upper-bound: 5.88e-06
median-lower-bound: 1.47e-06
median-upper-bound: 5.88e-06
min-lower-bound: 1.47e-06

Load/1

max-upper-bound: 1.204e-05
median-lower-bound: 2.97e-06
median-upper-bound: 1.188e-05
min-lower-bound: 2.93e-06

Load/2

max-upper-bound: 1.204e-05

median-lower-bound: 2.97e-06

median-upper-bound: 1.188e-05

min-lower-bound: 2.93e-06

Load/3

max-upper-bound: 1.204e-05

median-lower-bound: 2.97e-06

median-upper-bound: 1.188e-05

min-lower-bound: 2.93e-06

Load/4

max-upper-bound: 1.204e-05

median-lower-bound: 2.97e-06

median-upper-bound: 1.188e-05

min-lower-bound: 2.93e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFlZ2eLDfIYc=

DirtyCache

max-upper-bound: 1.7024e-05

median-lower-bound: 3.712e-06

median-upper-bound: 1.4848e-05

min-lower-bound: 3.192e-06

FullCache

max-upper-bound: 8.136e-06

median-lower-bound: 9.18e-07

median-upper-bound: 3.672e-06

min-lower-bound: 8e-07

HotCache

max-upper-bound: 1.864e-06

median-lower-bound: 3.42e-07

median-upper-bound: 1.368e-06

min-lower-bound: 3.42e-07

Load/1

max-upper-bound: 3.5736e-05

median-lower-bound: 5.126e-06

median-upper-bound: 2.0504e-05

min-lower-bound: 4.184e-06

Load/2

max-upper-bound: 5.7856e-05

median-lower-bound: 1.221e-05

median-upper-bound: 4.884e-05

min-lower-bound: 1.1982e-05

Load/3

max-upper-bound: 5.972e-05

median-lower-bound: 1.3606e-05

median-upper-bound: 5.4424e-05

min-lower-bound: 1.0836e-05

Load/4

max-upper-bound: 5.6312e-05

median-lower-bound: 1.3302e-05

median-upper-bound: 5.3208e-05

min-lower-bound: 1.0662e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.08e-06

median-lower-bound: 1.27e-06

median-upper-bound: 5.08e-06

min-lower-bound: 1.26e-06

FullCache

max-upper-bound: 5e-06

median-lower-bound: 1.23e-06

median-upper-bound: 4.92e-06

min-lower-bound: 1.18e-06

HotCache

max-upper-bound: 4.88e-06

median-lower-bound: 1.22e-06

median-upper-bound: 4.88e-06

min-lower-bound: 1.22e-06

Load/1

max-upper-bound: 1.104e-05

median-lower-bound: 2.76e-06

median-upper-bound: 1.104e-05

min-lower-bound: 2.6e-06

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOfI2eLDfIYc=

DirtyCache

max-upper-bound: 9.32e-06

median-lower-bound: 2.19e-06

median-upper-bound: 8.76e-06

min-lower-bound: 2.04e-06

FullCache

max-upper-bound: 7.624e-06

median-lower-bound: 8.18e-07

median-upper-bound: 3.272e-06

min-lower-bound: 7.08e-07

HotCache

max-upper-bound: 1.08e-06

median-lower-bound: 2.48e-07

median-upper-bound: 9.92e-07

min-lower-bound: 2.48e-07

Load/1

max-upper-bound: 1.1848e-05

median-lower-bound: 2.812e-06

median-upper-bound: 1.1248e-05

min-lower-bound: 2.742e-06

This requirement refines [spec:/rtems/part/req/perf-runtime](#).

5.2.22 spec:rtems/part/req/perf-get-no-buffer

While the execution environment is below environments, while a partition has no free buffers, while the measurement sample is the runtime of exactly one unsatisfied call to rtems_partition_get_buffer, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 6e-06
median-lower-bound: 1.5e-06
median-upper-bound: 6e-06
min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 6e-06
median-lower-bound: 1.5e-06
median-upper-bound: 6e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06
median-lower-bound: 1.5e-06
median-upper-bound: 6e-06
min-lower-bound: 1e-06

Load/1

max-upper-bound: 1.4e-05
median-lower-bound: 3.5e-06
median-upper-bound: 1.4e-05
min-lower-bound: 3e-06

Load/2

max-upper-bound: 1.4e-05
median-lower-bound: 3e-06
median-upper-bound: 1.2e-05
min-lower-bound: 3e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.6e-05
median-lower-bound: 6e-06
median-upper-bound: 2.4e-05
min-lower-bound: 6e-06

FullCache

max-upper-bound: 1.2e-05
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

Load/1

max-upper-bound: 5e-05
median-lower-bound: 1.2e-05
median-upper-bound: 4.8e-05
min-lower-bound: 1.2e-05

Load/2

max-upper-bound: 5e-05
median-lower-bound: 1.2e-05
median-upper-bound: 4.8e-05
min-lower-bound: 1.2e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 1.8e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 4e-06

FullCache

max-upper-bound: 6e-06

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 5e-07

HotCache

max-upper-bound: 4e-06

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 5e-07

Load/1

max-upper-bound: 2.4e-05

median-lower-bound: 6e-06

median-upper-bound: 2.4e-05

min-lower-bound: 5.5e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.64e-06

median-lower-bound: 1.41e-06

median-upper-bound: 5.64e-06

min-lower-bound: 1.33e-06

FullCache

max-upper-bound: 5.36e-06

median-lower-bound: 1.29e-06

median-upper-bound: 5.16e-06

min-lower-bound: 1.29e-06

HotCache

max-upper-bound: 5.44e-06

median-lower-bound: 1.36e-06

median-upper-bound: 5.44e-06

min-lower-bound: 1.36e-06

Load/1

max-upper-bound: 1.136e-05

median-lower-bound: 2.8e-06

median-upper-bound: 1.12e-05

min-lower-bound: 2.76e-06

Load/2

max-upper-bound: 1.136e-05

median-lower-bound: 2.8e-06

median-upper-bound: 1.12e-05

min-lower-bound: 2.76e-06

Load/3

max-upper-bound: 1.136e-05

median-lower-bound: 2.8e-06

median-upper-bound: 1.12e-05

min-lower-bound: 2.76e-06

Load/4

max-upper-bound: 1.136e-05

median-lower-bound: 2.8e-06

median-upper-bound: 1.12e-05

min-lower-bound: 2.76e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 1.3712e-05

median-lower-bound: 3.21e-06

median-upper-bound: 1.284e-05

min-lower-bound: 3.022e-06

FullCache

max-upper-bound: 5.408e-06

median-lower-bound: 7.82e-07

median-upper-bound: 3.128e-06

min-lower-bound: 7.76e-07

HotCache

max-upper-bound: 1.768e-06

median-lower-bound: 3.18e-07

median-upper-bound: 1.272e-06

min-lower-bound: 3.18e-07

Load/1

max-upper-bound: 3.4616e-05

median-lower-bound: 4.812e-06

median-upper-bound: 1.9248e-05

min-lower-bound: 3.79e-06

Load/2

max-upper-bound: 5.0184e-05

median-lower-bound: 1.0182e-05

median-upper-bound: 4.0728e-05

min-lower-bound: 9.528e-06

Load/3

max-upper-bound: 4.3536e-05

median-lower-bound: 1.0662e-05

median-upper-bound: 4.2648e-05

min-lower-bound: 1.0012e-05

Load/4

max-upper-bound: 4.2232e-05

median-lower-bound: 1.0276e-05

median-upper-bound: 4.1104e-05

min-lower-bound: 9.11e-06

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 4.2e-06

median-lower-bound: 1.05e-06

median-upper-bound: 4.2e-06

min-lower-bound: 1.05e-06

FullCache

max-upper-bound: 4.12e-06

median-lower-bound: 1.02e-06

median-upper-bound: 4.08e-06

min-lower-bound: 9.8e-07

HotCache

max-upper-bound: 4.28e-06

median-lower-bound: 1.03e-06

median-upper-bound: 4.12e-06

min-lower-bound: 1.03e-06

Load/1

max-upper-bound: 1.02e-05

median-lower-bound: 2.54e-06

median-upper-bound: 1.016e-05

min-lower-bound: 2.51e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOfIZ2eLDfIYc=

DirtyCache

max-upper-bound: 7.752e-06

median-lower-bound: 1.912e-06

median-upper-bound: 7.648e-06

min-lower-bound: 1.886e-06

FullCache

max-upper-bound: 3.744e-06

median-lower-bound: 5.7e-07

median-upper-bound: 2.28e-06

min-lower-bound: 5.66e-07

HotCache

max-upper-bound: 8.96e-07

median-lower-bound: 2.24e-07

median-upper-bound: 8.96e-07

min-lower-bound: 2.24e-07

Load/1

max-upper-bound: 1.056e-05

median-lower-bound: 2.594e-06

median-upper-bound: 1.0376e-05

min-lower-bound: 2.548e-06

This requirement refines [spec:/rtems/part/req/perf-runtime](#).

5.2.23 spec:/rtems/part/req/perf-return-buffer

While the execution environment is below environments, while a partition has exactly 100 minus one free buffers, while the measurement sample is the runtime of exactly one successful call to rtems_partition_return_buffer, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 1e-05

median-lower-bound: 2.5e-06

median-upper-bound: 1e-05

min-lower-bound: 2e-06

FullCache

max-upper-bound: 1e-05

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 2e-06

HotCache

max-upper-bound: 1e-05

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 2e-06

Load/1

max-upper-bound: 1.6e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 3.5e-06

Load/2

max-upper-bound: 1.6e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 3.5e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 3.4e-05

median-lower-bound: 8.5e-06

median-upper-bound: 3.4e-05

min-lower-bound: 8e-06

FullCache

max-upper-bound: 1.2e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 5.4e-05
median-lower-bound: 1.35e-05
median-upper-bound: 5.4e-05
min-lower-bound: 1.3e-05

Load/2

max-upper-bound: 5.6e-05
median-lower-bound: 1.35e-05
median-upper-bound: 5.4e-05
min-lower-bound: 1.35e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.2e-05

median-lower-bound: 5e-06

median-upper-bound: 2e-05

min-lower-bound: 5e-06

FullCache

max-upper-bound: 8e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 3e-05

median-lower-bound: 7e-06

median-upper-bound: 2.8e-05

min-lower-bound: 7e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.56e-06

median-lower-bound: 2.14e-06

median-upper-bound: 8.56e-06

min-lower-bound: 2.14e-06

FullCache

max-upper-bound: 8.36e-06

median-lower-bound: 2.09e-06

median-upper-bound: 8.36e-06

min-lower-bound: 2.01e-06

HotCache

max-upper-bound: 8.24e-06

median-lower-bound: 2.06e-06

median-upper-bound: 8.24e-06

min-lower-bound: 2.02e-06

Load/1

max-upper-bound: 1.428e-05

median-lower-bound: 3.57e-06

median-upper-bound: 1.428e-05

min-lower-bound: 3.57e-06

Load/2

max-upper-bound: 1.428e-05

median-lower-bound: 3.57e-06

median-upper-bound: 1.428e-05

min-lower-bound: 3.57e-06

Load/3

max-upper-bound: 1.428e-05

median-lower-bound: 3.57e-06

median-upper-bound: 1.428e-05

min-lower-bound: 3.57e-06

Load/4

max-upper-bound: 1.428e-05

median-lower-bound: 3.57e-06

median-upper-bound: 1.428e-05

min-lower-bound: 3.53e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 1.724e-05

median-lower-bound: 3.766e-06

median-upper-bound: 1.5064e-05

min-lower-bound: 3.5e-06

FullCache

max-upper-bound: 4.992e-06

median-lower-bound: 7.86e-07

median-upper-bound: 3.144e-06

min-lower-bound: 7.86e-07

HotCache

max-upper-bound: 2.272e-06
median-lower-bound: 4.46e-07
median-upper-bound: 1.784e-06
min-lower-bound: 4.46e-07

Load/1

max-upper-bound: 3.5808e-05
median-lower-bound: 5.322e-06
median-upper-bound: 2.1288e-05
min-lower-bound: 4.57e-06

Load/2

max-upper-bound: 5.872e-05
median-lower-bound: 1.3486e-05
median-upper-bound: 5.3944e-05
min-lower-bound: 1.2666e-05

Load/3

max-upper-bound: 5.972e-05
median-lower-bound: 1.4692e-05
median-upper-bound: 5.8768e-05
min-lower-bound: 1.2684e-05

Load/4

max-upper-bound: 6.204e-05
median-lower-bound: 1.5324e-05
median-upper-bound: 6.1296e-05
min-lower-bound: 1.2416e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.32e-06
median-lower-bound: 1.83e-06
median-upper-bound: 7.32e-06
min-lower-bound: 1.77e-06

FullCache

max-upper-bound: 6.76e-06

median-lower-bound: 1.69e-06

median-upper-bound: 6.76e-06

min-lower-bound: 1.69e-06

HotCache

max-upper-bound: 6.96e-06

median-lower-bound: 1.74e-06

median-upper-bound: 6.96e-06

min-lower-bound: 1.74e-06

Load/1

max-upper-bound: 1.36e-05

median-lower-bound: 3.34e-06

median-upper-bound: 1.336e-05

min-lower-bound: 3.22e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 9.288e-06

median-lower-bound: 2.142e-06

median-upper-bound: 8.568e-06

min-lower-bound: 1.998e-06

FullCache

max-upper-bound: 4.528e-06

median-lower-bound: 6.96e-07

median-upper-bound: 2.784e-06

min-lower-bound: 6.92e-07

HotCache

max-upper-bound: 1.408e-06

median-lower-bound: 3.52e-07

median-upper-bound: 1.408e-06

min-lower-bound: 3.52e-07

Load/1

max-upper-bound: 1.276e-05

median-lower-bound: 2.994e-06

median-upper-bound: 1.1976e-05

min-lower-bound: 2.846e-06

This requirement refines *spec:/rtems/part/req/perf-runtime*.

5.2.24 spec:/rtems/part/req/perf-runtime

spec:/rtems/part/req/perf-runtime

The runtime of @ref RTEMSAPIClassicPart directives shall be measured.

rationale: N/A

This requirement refines *spec:/rtems/req/perf-runtime*.

This requirement refines *spec:/rtems/part/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/part/req/perf-get-buffer*
- *spec:/rtems/part/req/perf-get-no-buffer*
- *spec:/rtems/part/req/perf-return-buffer*

5.2.25 spec:/rtems/req/perf-runtime

spec:/rtems/req/perf-runtime

The runtime of the @ref RTEMSAPIClassic directives shall be measured.

rationale: N/A

This requirement refines *spec:/req/perf-runtime*.

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/barrier/req/perf-runtime*
- *spec:/rtems/event/req/perf-runtime*
- *spec:/rtems/message/req/perf-runtime*
- *spec:/rtems/part/req/perf-runtime*
- *spec:/rtems/sem/req/perf-runtime*
- *spec:/rtems/task/req/perf-runtime*

5.2.26 spec:rtems/sem/req/perf-mtx-pi-obtain

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while the measurement sample is the runtime of exactly one successful call to rtems_semaphore_obtain which immediately obtains the mutex, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 1.6e-05

median-lower-bound: 3.5e-06

median-upper-bound: 1.4e-05

min-lower-bound: 3e-06

Load/2

max-upper-bound: 1.4e-05

median-lower-bound: 3.5e-06

median-upper-bound: 1.4e-05

min-lower-bound: 3e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 3.6e-05
median-lower-bound: 8.5e-06
median-upper-bound: 3.4e-05
min-lower-bound: 8.5e-06

FullCache

max-upper-bound: 1.8e-05
median-lower-bound: 1.5e-06
median-upper-bound: 6e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

Load/1

max-upper-bound: 5.4e-05
median-lower-bound: 1.35e-05
median-upper-bound: 5.4e-05
min-lower-bound: 1.3e-05

Load/2

max-upper-bound: 5.6e-05
median-lower-bound: 1.35e-05
median-upper-bound: 5.4e-05
min-lower-bound: 1.35e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.2e-05

median-lower-bound: 5e-06

median-upper-bound: 2e-05

min-lower-bound: 5e-06

FullCache

max-upper-bound: 1.6e-05

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 5e-07

HotCache

max-upper-bound: 4e-06

median-lower-bound: 1e-06

median-upper-bound: 4e-06

min-lower-bound: 5e-07

Load/1

max-upper-bound: 3e-05

median-lower-bound: 7e-06

median-upper-bound: 2.8e-05

min-lower-bound: 7e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcl0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.08e-06

median-lower-bound: 1.77e-06

median-upper-bound: 7.08e-06

min-lower-bound: 1.73e-06

FullCache

max-upper-bound: 6.48e-06

median-lower-bound: 1.59e-06

median-upper-bound: 6.36e-06

min-lower-bound: 1.59e-06

HotCache

max-upper-bound: 6.32e-06

median-lower-bound: 1.58e-06

median-upper-bound: 6.32e-06

min-lower-bound: 1.58e-06

Load/1

max-upper-bound: 1.28e-05

median-lower-bound: 3.16e-06

median-upper-bound: 1.264e-05

min-lower-bound: 3.16e-06

Load/2

max-upper-bound: 1.28e-05

median-lower-bound: 3.16e-06

median-upper-bound: 1.264e-05

min-lower-bound: 3.16e-06

Load/3

max-upper-bound: 1.28e-05

median-lower-bound: 3.16e-06

median-upper-bound: 1.264e-05

min-lower-bound: 3.16e-06

Load/4

max-upper-bound: 1.28e-05

median-lower-bound: 3.16e-06

median-upper-bound: 1.264e-05

min-lower-bound: 3.16e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 2.052e-05

median-lower-bound: 4.506e-06

median-upper-bound: 1.8024e-05

min-lower-bound: 4.25e-06

FullCache

max-upper-bound: 1.176e-05

median-lower-bound: 7.18e-07

median-upper-bound: 2.872e-06

min-lower-bound: 7.12e-07

HotCache

max-upper-bound: 1.968e-06

median-lower-bound: 3.66e-07

median-upper-bound: 1.464e-06

min-lower-bound: 3.66e-07

Load/1

max-upper-bound: 3.9384e-05

median-lower-bound: 5.59e-06

median-upper-bound: 2.236e-05

min-lower-bound: 4.662e-06

Load/2

max-upper-bound: 6.8328e-05

median-lower-bound: 1.576e-05

median-upper-bound: 6.304e-05

min-lower-bound: 1.5654e-05

Load/3

max-upper-bound: 7.6264e-05

median-lower-bound: 1.8946e-05

median-upper-bound: 7.5784e-05

min-lower-bound: 1.6724e-05

Load/4

max-upper-bound: 8.4928e-05

median-lower-bound: 1.8642e-05

median-upper-bound: 7.4568e-05

min-lower-bound: 1.7214e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.52e-06

median-lower-bound: 1.34e-06

median-upper-bound: 5.36e-06

min-lower-bound: 1.34e-06

FullCache

max-upper-bound: 5.28e-06

median-lower-bound: 1.28e-06

median-upper-bound: 5.12e-06

min-lower-bound: 1.22e-06

HotCache

max-upper-bound: 5.08e-06

median-lower-bound: 1.27e-06

median-upper-bound: 5.08e-06

min-lower-bound: 1.23e-06

Load/1

max-upper-bound: 1.14e-05

median-lower-bound: 2.85e-06

median-upper-bound: 1.14e-05

min-lower-bound: 2.79e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOfIz2eLDflYc=

DirtyCache

max-upper-bound: 1.0912e-05

median-lower-bound: 2.262e-06

median-upper-bound: 9.048e-06

min-lower-bound: 2.238e-06

FullCache

max-upper-bound: 8.192e-06

median-lower-bound: 2.64e-07

median-upper-bound: 1.056e-06

min-lower-bound: 2.64e-07

HotCache

max-upper-bound: 1.056e-06

median-lower-bound: 2.64e-07

median-upper-bound: 1.056e-06

min-lower-bound: 2.64e-07

Load/1

max-upper-bound: 1.2552e-05

median-lower-bound: 2.96e-06

median-upper-bound: 1.184e-05

min-lower-bound: 2.92e-06

This requirement refines [spec:/rtems/sem/req/perf-runtime](#).

5.2.27 spec:/rtems/sem/req/perf-mtx-pi-release

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while the measurement sample is the runtime of exactly one successful call to rtems_semaphore_release which does not unblock a waiting task, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 1.6e-05

median-lower-bound: 3.5e-06

median-upper-bound: 1.4e-05

min-lower-bound: 3.5e-06

Load/2

max-upper-bound: 1.6e-05

median-lower-bound: 3.5e-06

median-upper-bound: 1.4e-05

min-lower-bound: 3e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 3.2e-05

median-lower-bound: 8e-06

median-upper-bound: 3.2e-05

min-lower-bound: 7.5e-06

FullCache

max-upper-bound: 1.4e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 5.4e-05
median-lower-bound: 1.35e-05
median-upper-bound: 5.4e-05
min-lower-bound: 1.3e-05

Load/2

max-upper-bound: 5.6e-05
median-lower-bound: 1.35e-05
median-upper-bound: 5.4e-05
min-lower-bound: 1.3e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.2e-05
median-lower-bound: 5e-06
median-upper-bound: 2e-05
min-lower-bound: 5e-06

FullCache

max-upper-bound: 1.2e-05
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 5e-07

HotCache

max-upper-bound: 4e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 5e-07

Load/1

max-upper-bound: 3e-05
median-lower-bound: 7e-06
median-upper-bound: 2.8e-05
min-lower-bound: 7e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.2e-06
median-lower-bound: 1.76e-06
median-upper-bound: 7.04e-06
min-lower-bound: 1.72e-06

FullCache

max-upper-bound: 6.92e-06
median-lower-bound: 1.73e-06
median-upper-bound: 6.92e-06
min-lower-bound: 1.73e-06

HotCache

max-upper-bound: 6.64e-06
median-lower-bound: 1.66e-06

median-upper-bound: 6.64e-06

min-lower-bound: 1.66e-06

Load/1

max-upper-bound: 1.292e-05

median-lower-bound: 3.19e-06

median-upper-bound: 1.276e-05

min-lower-bound: 3.15e-06

Load/2

max-upper-bound: 1.292e-05

median-lower-bound: 3.19e-06

median-upper-bound: 1.276e-05

min-lower-bound: 3.15e-06

Load/3

max-upper-bound: 1.292e-05

median-lower-bound: 3.19e-06

median-upper-bound: 1.276e-05

min-lower-bound: 3.15e-06

Load/4

max-upper-bound: 1.292e-05

median-lower-bound: 3.19e-06

median-upper-bound: 1.276e-05

min-lower-bound: 3.15e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 1.8408e-05

median-lower-bound: 4.376e-06

median-upper-bound: 1.7504e-05

min-lower-bound: 4.276e-06

FullCache

max-upper-bound: 8.176e-06

median-lower-bound: 8.48e-07

median-upper-bound: 3.392e-06

min-lower-bound: 8.48e-07

HotCache

max-upper-bound: 2.056e-06
median-lower-bound: 3.92e-07
median-upper-bound: 1.568e-06
min-lower-bound: 3.92e-07

Load/1

max-upper-bound: 3.7456e-05
median-lower-bound: 5.768e-06
median-upper-bound: 2.3072e-05
min-lower-bound: 4.886e-06

Load/2

max-upper-bound: 6.5328e-05
median-lower-bound: 1.248e-05
median-upper-bound: 4.992e-05
min-lower-bound: 1.244e-05

Load/3

max-upper-bound: 7.2928e-05
median-lower-bound: 1.6402e-05
median-upper-bound: 6.5608e-05
min-lower-bound: 1.4684e-05

Load/4

max-upper-bound: 8.0488e-05
median-lower-bound: 1.7816e-05
median-upper-bound: 7.1264e-05
min-lower-bound: 1.515e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.48e-06
median-lower-bound: 1.37e-06
median-upper-bound: 5.48e-06
min-lower-bound: 1.37e-06

FullCache

max-upper-bound: 5.16e-06

median-lower-bound: 1.27e-06

median-upper-bound: 5.08e-06

min-lower-bound: 1.23e-06

HotCache

max-upper-bound: 5.04e-06

median-lower-bound: 1.26e-06

median-upper-bound: 5.04e-06

min-lower-bound: 1.26e-06

Load/1

max-upper-bound: 1.088e-05

median-lower-bound: 2.72e-06

median-upper-bound: 1.088e-05

min-lower-bound: 2.72e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 7.536e-06

median-lower-bound: 1.86e-06

median-upper-bound: 7.44e-06

min-lower-bound: 1.834e-06

FullCache

max-upper-bound: 2.92e-06

median-lower-bound: 3.84e-07

median-upper-bound: 1.536e-06

min-lower-bound: 3.84e-07

HotCache

max-upper-bound: 1.088e-06

median-lower-bound: 2.72e-07

median-upper-bound: 1.088e-06

min-lower-bound: 2.72e-07

Load/1

max-upper-bound: 1.0576e-05

median-lower-bound: 2.596e-06

median-upper-bound: 1.0384e-05

min-lower-bound: 2.552e-06

This requirement refines *spec:/rtems/sem/req/perf-runtime*.

5.2.28 spec:/rtems/sem/req/perf-mtx-pi-release-one

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while the measurement sample is the runtime of exactly one successful call to rtems_semaphore_release which does unblock exactly one waiting task which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 6.2e-05

median-lower-bound: 1.55e-05

median-upper-bound: 6.2e-05

min-lower-bound: 1.5e-05

FullCache

max-upper-bound: 5.8e-05

median-lower-bound: 1.45e-05

median-upper-bound: 5.8e-05

min-lower-bound: 1.4e-05

HotCache

max-upper-bound: 5.8e-05

median-lower-bound: 1.45e-05

median-upper-bound: 5.8e-05

min-lower-bound: 1.4e-05

Load/1

max-upper-bound: 7.6e-05

median-lower-bound: 1.9e-05

median-upper-bound: 7.6e-05

min-lower-bound: 1.85e-05

Load/2

max-upper-bound: 7.6e-05

median-lower-bound: 1.85e-05

median-upper-bound: 7.4e-05

min-lower-bound: 1.85e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000164

median-lower-bound: 4e-05

median-upper-bound: 0.00016

min-lower-bound: 4e-05

FullCache

max-upper-bound: 9.8e-05

median-lower-bound: 1.95e-05

median-upper-bound: 7.8e-05

min-lower-bound: 1.9e-05

HotCache

max-upper-bound: 7e-05

median-lower-bound: 1.75e-05

median-upper-bound: 7e-05

min-lower-bound: 1.7e-05

Load/1

max-upper-bound: 0.000176

median-lower-bound: 4.3e-05

median-upper-bound: 0.000172

min-lower-bound: 4.3e-05

Load/2

max-upper-bound: 0.000196

median-lower-bound: 4.85e-05

median-upper-bound: 0.000194

min-lower-bound: 4.8e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 8.8e-05

median-lower-bound: 2.2e-05

median-upper-bound: 8.8e-05

min-lower-bound: 2.15e-05

FullCache

max-upper-bound: 4.4e-05

median-lower-bound: 5.5e-06

median-upper-bound: 2.2e-05

min-lower-bound: 5.5e-06

HotCache

max-upper-bound: 2.2e-05

median-lower-bound: 5e-06

median-upper-bound: 2e-05

min-lower-bound: 5e-06

Load/1

max-upper-bound: 0.0001
median-lower-bound: 2.5e-05
median-upper-bound: 0.0001
min-lower-bound: 2.45e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.912e-05
median-lower-bound: 1.478e-05
median-upper-bound: 5.912e-05
min-lower-bound: 1.466e-05

FullCache

max-upper-bound: 5.524e-05
median-lower-bound: 1.373e-05
median-upper-bound: 5.492e-05
min-lower-bound: 1.369e-05

HotCache

max-upper-bound: 5.52e-05
median-lower-bound: 1.38e-05
median-upper-bound: 5.52e-05
min-lower-bound: 1.372e-05

Load/1

max-upper-bound: 7.112e-05
median-lower-bound: 1.778e-05
median-upper-bound: 7.112e-05
min-lower-bound: 1.778e-05

Load/2

max-upper-bound: 7.096e-05
median-lower-bound: 1.774e-05
median-upper-bound: 7.096e-05
min-lower-bound: 1.774e-05

Load/3

max-upper-bound: 7.024e-05

median-lower-bound: 1.756e-05

median-upper-bound: 7.024e-05

min-lower-bound: 1.756e-05

Load/4

max-upper-bound: 6.992e-05

median-lower-bound: 1.748e-05

median-upper-bound: 6.992e-05

min-lower-bound: 1.748e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 6.4032e-05

median-lower-bound: 1.5704e-05

median-upper-bound: 6.2816e-05

min-lower-bound: 1.5162e-05

FullCache

max-upper-bound: 5.064e-05

median-lower-bound: 7.956e-06

median-upper-bound: 3.1824e-05

min-lower-bound: 7.91e-06

HotCache

max-upper-bound: 1.644e-05

median-lower-bound: 3.95e-06

median-upper-bound: 1.58e-05

min-lower-bound: 3.95e-06

Load/1

max-upper-bound: 0.000102024

median-lower-bound: 1.6424e-05

median-upper-bound: 6.5696e-05

min-lower-bound: 1.5496e-05

Load/2

max-upper-bound: 0.00038412

median-lower-bound: 9.2088e-05

median-upper-bound: 0.000368352

min-lower-bound: 9.2044e-05

Load/3

max-upper-bound: 0.000585536

median-lower-bound: 0.000144302

median-upper-bound: 0.000577208

min-lower-bound: 0.000142448

Load/4

max-upper-bound: 0.000676192

median-lower-bound: 0.000166508

median-upper-bound: 0.000666032

min-lower-bound: 0.00016302

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 3.236e-05

median-lower-bound: 8.01e-06

median-upper-bound: 3.204e-05

min-lower-bound: 8.01e-06

FullCache

max-upper-bound: 2.78e-05

median-lower-bound: 6.95e-06

median-upper-bound: 2.78e-05

min-lower-bound: 6.95e-06

HotCache

max-upper-bound: 2.8e-05

median-lower-bound: 7e-06

median-upper-bound: 2.8e-05

min-lower-bound: 7e-06

Load/1

max-upper-bound: 4.38e-05

median-lower-bound: 1.091e-05

median-upper-bound: 4.364e-05

min-lower-bound: 1.079e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDf1Yc=

DirtyCache

max-upper-bound: 3.7536e-05
median-lower-bound: 8.984e-06
median-upper-bound: 3.5936e-05
min-lower-bound: 8.934e-06

FullCache

max-upper-bound: 2.204e-05
median-lower-bound: 3.048e-06
median-upper-bound: 1.2192e-05
min-lower-bound: 3.042e-06

HotCache

max-upper-bound: 5.744e-06
median-lower-bound: 1.412e-06
median-upper-bound: 5.648e-06
min-lower-bound: 1.412e-06

Load/1

max-upper-bound: 4.2712e-05
median-lower-bound: 1.0254e-05
median-upper-bound: 4.1016e-05
min-lower-bound: 1.0204e-05

This requirement refines [spec:/rtems/sem/req/perf-runtime](#).

5.2.29 spec:/rtems/sem/req/perf-mtx-pi-release-other-cpu

Let U and V be two tasks with distinct home schedulers. Let B be a time point measured by U right before a call to rtems_semaphore_release which unblocks task V which is scheduled on another processor. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while exactly task V waits on the mutex, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000126
median-lower-bound: 3.1e-05
median-upper-bound: 0.000124
min-lower-bound: 3.05e-05

FullCache

max-upper-bound: 0.000124
median-lower-bound: 3.05e-05
median-upper-bound: 0.000122
min-lower-bound: 3e-05

HotCache

max-upper-bound: 0.000124
median-lower-bound: 3.1e-05
median-upper-bound: 0.000124
min-lower-bound: 3.05e-05

Load/1

max-upper-bound: 0.000134
median-lower-bound: 3.35e-05
median-upper-bound: 0.000134
min-lower-bound: 3.3e-05

Load/2

max-upper-bound: 0.000134
median-lower-bound: 3.3e-05
median-upper-bound: 0.000132
min-lower-bound: 3.25e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000318
median-lower-bound: 7.9e-05
median-upper-bound: 0.000316
min-lower-bound: 7.85e-05

FullCache

max-upper-bound: 0.000196
median-lower-bound: 3e-05

median-upper-bound: 0.00012

min-lower-bound: 2.95e-05

HotCache

max-upper-bound: 0.000118

median-lower-bound: 2.9e-05

median-upper-bound: 0.000116

min-lower-bound: 2.85e-05

Load/1

max-upper-bound: 0.000314

median-lower-bound: 7.8e-05

median-upper-bound: 0.000312

min-lower-bound: 7.8e-05

Load/2

max-upper-bound: 0.000366

median-lower-bound: 9.1e-05

median-upper-bound: 0.000364

min-lower-bound: 9.05e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.00011944

median-lower-bound: 2.975e-05

median-upper-bound: 0.000119

min-lower-bound: 2.926e-05

FullCache

max-upper-bound: 0.00011764

median-lower-bound: 2.941e-05

median-upper-bound: 0.00011764

min-lower-bound: 2.886e-05

HotCache

max-upper-bound: 0.00011684

median-lower-bound: 2.921e-05

median-upper-bound: 0.00011684

min-lower-bound: 2.914e-05

Load/1

max-upper-bound: 0.00012576
median-lower-bound: 3.132e-05
median-upper-bound: 0.00012528
min-lower-bound: 3.124e-05

Load/2

max-upper-bound: 0.0001258
median-lower-bound: 3.145e-05
median-upper-bound: 0.0001258
min-lower-bound: 3.127e-05

Load/3

max-upper-bound: 0.0001258
median-lower-bound: 3.145e-05
median-upper-bound: 0.0001258
min-lower-bound: 3.118e-05

Load/4

max-upper-bound: 0.00012564
median-lower-bound: 3.127e-05
median-upper-bound: 0.00012508
min-lower-bound: 3.117e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 0.0001264
median-lower-bound: 2.8146e-05
median-upper-bound: 0.000112584
min-lower-bound: 2.7626e-05

FullCache

max-upper-bound: 9.1288e-05
median-lower-bound: 1.225e-05
median-upper-bound: 4.9e-05
min-lower-bound: 1.21e-05

HotCache

max-upper-bound: 2.8448e-05

median-lower-bound: 6.928e-06

median-upper-bound: 2.7712e-05

min-lower-bound: 6.928e-06

Load/1

max-upper-bound: 0.000145552

median-lower-bound: 2.6358e-05

median-upper-bound: 0.000105432

min-lower-bound: 2.6086e-05

Load/2

max-upper-bound: 0.000639944

median-lower-bound: 0.0001571

median-upper-bound: 0.0006284

min-lower-bound: 0.000156856

Load/3

max-upper-bound: 0.001071048

median-lower-bound: 0.000264426

median-upper-bound: 0.001057704

min-lower-bound: 0.000263772

Load/4

max-upper-bound: 0.001344168

median-lower-bound: 0.000333384

median-upper-bound: 0.001333536

min-lower-bound: 0.00032833

This requirement refines [*spec:/rtems/sem/req/perf-runtime*](#).

5.2.30 spec:/rtems/sem/req/perf-mtx-pi-release-preempt

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to `rtems_semaphore_release` which unblocks task V which does preempt the caller. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while exactly task V waits for the mutex, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000118
median-lower-bound: 2.95e-05
median-upper-bound: 0.000118
min-lower-bound: 2.9e-05

FullCache

max-upper-bound: 0.000116
median-lower-bound: 2.9e-05
median-upper-bound: 0.000116
min-lower-bound: 2.9e-05

HotCache

max-upper-bound: 0.000116
median-lower-bound: 2.9e-05
median-upper-bound: 0.000116
min-lower-bound: 2.85e-05

Load/1

max-upper-bound: 0.000132
median-lower-bound: 3.3e-05
median-upper-bound: 0.000132
min-lower-bound: 3.25e-05

Load/2

max-upper-bound: 0.000132
median-lower-bound: 3.25e-05
median-upper-bound: 0.00013
min-lower-bound: 3.25e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000278
median-lower-bound: 6.9e-05
median-upper-bound: 0.000276
min-lower-bound: 6.9e-05

FullCache

max-upper-bound: 0.000148
median-lower-bound: 2.9e-05
median-upper-bound: 0.000116
min-lower-bound: 2.85e-05

HotCache

max-upper-bound: 0.000106
median-lower-bound: 2.65e-05
median-upper-bound: 0.000106
min-lower-bound: 2.6e-05

Load/1

max-upper-bound: 0.000286
median-lower-bound: 7.1e-05
median-upper-bound: 0.000284
min-lower-bound: 7.05e-05

Load/2

max-upper-bound: 0.000334
median-lower-bound: 8.3e-05
median-upper-bound: 0.000332
min-lower-bound: 8.25e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000158

median-lower-bound: 3.9e-05

median-upper-bound: 0.000156

min-lower-bound: 3.9e-05

FullCache

max-upper-bound: 8.4e-05

median-lower-bound: 1.35e-05

median-upper-bound: 5.4e-05

min-lower-bound: 1.35e-05

HotCache

max-upper-bound: 5.2e-05

median-lower-bound: 1.25e-05

median-upper-bound: 5e-05

min-lower-bound: 1.25e-05

Load/1

max-upper-bound: 0.000164

median-lower-bound: 4.1e-05

median-upper-bound: 0.000164

min-lower-bound: 4.05e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.000113

median-lower-bound: 2.811e-05

median-upper-bound: 0.00011244

min-lower-bound: 2.793e-05

FullCache

max-upper-bound: 0.0001104

median-lower-bound: 2.76e-05

median-upper-bound: 0.0001104

min-lower-bound: 2.756e-05

HotCache

max-upper-bound: 0.00011076

median-lower-bound: 2.769e-05

median-upper-bound: 0.00011076

min-lower-bound: 2.769e-05

Load/1

max-upper-bound: 0.0001256

median-lower-bound: 3.132e-05

median-upper-bound: 0.00012528

min-lower-bound: 3.128e-05

Load/2

max-upper-bound: 0.0001256

median-lower-bound: 3.132e-05

median-upper-bound: 0.00012528

min-lower-bound: 3.128e-05

Load/3

max-upper-bound: 0.0001256

median-lower-bound: 3.132e-05

median-upper-bound: 0.00012528

min-lower-bound: 3.128e-05

Load/4

max-upper-bound: 0.0001256

median-lower-bound: 3.132e-05

median-upper-bound: 0.00012528

min-lower-bound: 3.128e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 0.000109656
median-lower-bound: 2.6024e-05
median-upper-bound: 0.000104096
min-lower-bound: 2.511e-05

FullCache

max-upper-bound: 6.7032e-05
median-lower-bound: 1.1418e-05
median-upper-bound: 4.5672e-05
min-lower-bound: 1.1368e-05

HotCache

max-upper-bound: 2.4144e-05
median-lower-bound: 6.018e-06
median-upper-bound: 2.4072e-05
min-lower-bound: 6.018e-06

Load/1

max-upper-bound: 0.000137808
median-lower-bound: 2.4774e-05
median-upper-bound: 9.9096e-05
min-lower-bound: 2.4548e-05

Load/2

max-upper-bound: 0.000701616
median-lower-bound: 0.00017535
median-upper-bound: 0.0007014
min-lower-bound: 0.000174094

Load/3

max-upper-bound: 0.0010992
median-lower-bound: 0.000273658
median-upper-bound: 0.001094632
min-lower-bound: 0.000271588

Load/4

max-upper-bound: 0.001299624
median-lower-bound: 0.000323146

median-upper-bound: 0.001292584

min-lower-bound: 0.000318906

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.764e-05

median-lower-bound: 1.689e-05

median-upper-bound: 6.756e-05

min-lower-bound: 1.677e-05

FullCache

max-upper-bound: 6.584e-05

median-lower-bound: 1.646e-05

median-upper-bound: 6.584e-05

min-lower-bound: 1.646e-05

HotCache

max-upper-bound: 6.58e-05

median-lower-bound: 1.645e-05

median-upper-bound: 6.58e-05

min-lower-bound: 1.645e-05

Load/1

max-upper-bound: 7.568e-05

median-lower-bound: 1.892e-05

median-upper-bound: 7.568e-05

min-lower-bound: 1.872e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 6.1384e-05

median-lower-bound: 1.4904e-05

median-upper-bound: 5.9616e-05

min-lower-bound: 1.478e-05

FullCache

max-upper-bound: 3.3224e-05

median-lower-bound: 6.172e-06

median-upper-bound: 2.4688e-05

min-lower-bound: 6.16e-06

HotCache

max-upper-bound: 1.152e-05

median-lower-bound: 2.88e-06

median-upper-bound: 1.152e-05

min-lower-bound: 2.88e-06

Load/1

max-upper-bound: 6.5792e-05

median-lower-bound: 1.5866e-05

median-upper-bound: 6.3464e-05

min-lower-bound: 1.5806e-05

This requirement refines *spec:/rtems/sem/req/perf-runtime*.

5.2.31 spec:/rtems/sem/req/perf-mtx-pi-try

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while the measurement sample is the runtime of exactly one unsatisfied call to rtems_semaphore_obtain which immediately returns, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

FullCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

HotCache

max-upper-bound: 8e-06

median-lower-bound: 2e-06

median-upper-bound: 8e-06

min-lower-bound: 1.5e-06

Load/1

max-upper-bound: 1.6e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 4e-06

Load/2

max-upper-bound: 1.8e-05

median-lower-bound: 4e-06

median-upper-bound: 1.6e-05

min-lower-bound: 4e-06

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 4e-05

median-lower-bound: 9.5e-06

median-upper-bound: 3.8e-05

min-lower-bound: 9e-06

FullCache

max-upper-bound: 1.8e-05

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06

median-lower-bound: 1.5e-06

median-upper-bound: 6e-06

min-lower-bound: 1e-06

Load/1

max-upper-bound: 6e-05

median-lower-bound: 1.45e-05

median-upper-bound: 5.8e-05

min-lower-bound: 1.45e-05

Load/2

max-upper-bound: 6.2e-05
median-lower-bound: 1.5e-05
median-upper-bound: 6e-05
min-lower-bound: 1.5e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 2.4e-05
median-lower-bound: 6e-06
median-upper-bound: 2.4e-05
min-lower-bound: 6e-06

FullCache

max-upper-bound: 1.4e-05
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

HotCache

max-upper-bound: 6e-06
median-lower-bound: 1e-06
median-upper-bound: 4e-06
min-lower-bound: 1e-06

Load/1

max-upper-bound: 3.6e-05
median-lower-bound: 8.5e-06
median-upper-bound: 3.4e-05
min-lower-bound: 8.5e-06

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.48e-06
median-lower-bound: 1.87e-06
median-upper-bound: 7.48e-06
min-lower-bound: 1.79e-06

FullCache

max-upper-bound: 7.2e-06
median-lower-bound: 1.73e-06
median-upper-bound: 6.92e-06
min-lower-bound: 1.73e-06

HotCache

max-upper-bound: 7.2e-06
median-lower-bound: 1.8e-06
median-upper-bound: 7.2e-06
min-lower-bound: 1.8e-06

Load/1

max-upper-bound: 1.532e-05
median-lower-bound: 3.77e-06

median-upper-bound: 1.508e-05

min-lower-bound: 3.69e-06

Load/2

max-upper-bound: 1.532e-05

median-lower-bound: 3.77e-06

median-upper-bound: 1.508e-05

min-lower-bound: 3.69e-06

Load/3

max-upper-bound: 1.532e-05

median-lower-bound: 3.77e-06

median-upper-bound: 1.508e-05

min-lower-bound: 3.69e-06

Load/4

max-upper-bound: 1.532e-05

median-lower-bound: 3.77e-06

median-upper-bound: 1.508e-05

min-lower-bound: 3.69e-06

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 1.8456e-05

median-lower-bound: 4.476e-06

median-upper-bound: 1.7904e-05

min-lower-bound: 4.076e-06

FullCache

max-upper-bound: 9.352e-06

median-lower-bound: 7.62e-07

median-upper-bound: 3.048e-06

min-lower-bound: 7.58e-07

HotCache

max-upper-bound: 2.152e-06

median-lower-bound: 4.12e-07

median-upper-bound: 1.648e-06

min-lower-bound: 4.12e-07

Load/1

max-upper-bound: 4.0864e-05
median-lower-bound: 5.69e-06
median-upper-bound: 2.276e-05
min-lower-bound: 4.626e-06

Load/2

max-upper-bound: 7.6544e-05
median-lower-bound: 1.7774e-05
median-upper-bound: 7.1096e-05
min-lower-bound: 1.7694e-05

Load/3

max-upper-bound: 9.896e-05
median-lower-bound: 2.4178e-05
median-upper-bound: 9.6712e-05
min-lower-bound: 2.0998e-05

Load/4

max-upper-bound: 0.000107128
median-lower-bound: 2.4432e-05
median-upper-bound: 9.7728e-05
min-lower-bound: 2.1314e-05

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.04e-06
median-lower-bound: 1.46e-06
median-upper-bound: 5.84e-06
min-lower-bound: 1.46e-06

FullCache

max-upper-bound: 5.8e-06
median-lower-bound: 1.41e-06
median-upper-bound: 5.64e-06
min-lower-bound: 1.37e-06

HotCache

max-upper-bound: 5.76e-06

median-lower-bound: 1.44e-06

median-upper-bound: 5.76e-06

min-lower-bound: 1.44e-06

Load/1

max-upper-bound: 1.4e-05

median-lower-bound: 3.45e-06

median-upper-bound: 1.38e-05

min-lower-bound: 3.34e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 9.6e-06

median-lower-bound: 2.358e-06

median-upper-bound: 9.432e-06

min-lower-bound: 2.334e-06

FullCache

max-upper-bound: 4.856e-06

median-lower-bound: 3.1e-07

median-upper-bound: 1.24e-06

min-lower-bound: 3.1e-07

HotCache

max-upper-bound: 1.24e-06

median-lower-bound: 3.1e-07

median-upper-bound: 1.24e-06

min-lower-bound: 3.1e-07

Load/1

max-upper-bound: 1.2512e-05

median-lower-bound: 3.09e-06

median-upper-bound: 1.236e-05

min-lower-bound: 3.05e-06

This requirement refines [spec:/rtems/sem/req/perf-runtime](#).

5.2.32 spec:rtems/sem/req/perf-mtx-pi-wait-forever

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to rtems_semaphore_obtain which blocks on the mutex with no timeout. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

<sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=>

DirtyCache

max-upper-bound: 0.00011
median-lower-bound: 2.75e-05
median-upper-bound: 0.00011
min-lower-bound: 2.75e-05

FullCache

max-upper-bound: 0.000108
median-lower-bound: 2.7e-05
median-upper-bound: 0.000108
min-lower-bound: 2.65e-05

HotCache

max-upper-bound: 0.000108
median-lower-bound: 2.7e-05
median-upper-bound: 0.000108
min-lower-bound: 2.65e-05

Load/1

max-upper-bound: 0.000124
median-lower-bound: 3.1e-05
median-upper-bound: 0.000124
min-lower-bound: 3.05e-05

Load/2

max-upper-bound: 0.000124
median-lower-bound: 3.1e-05
median-upper-bound: 0.000124

min-lower-bound: 3.05e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574IY=

DirtyCache

max-upper-bound: 0.000276

median-lower-bound: 6.9e-05

median-upper-bound: 0.000276

min-lower-bound: 6.85e-05

FullCache

max-upper-bound: 0.000158

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.85e-05

HotCache

max-upper-bound: 0.000108

median-lower-bound: 2.7e-05

median-upper-bound: 0.000108

min-lower-bound: 2.65e-05

Load/1

max-upper-bound: 0.000284

median-lower-bound: 7.05e-05

median-upper-bound: 0.000282

min-lower-bound: 7.05e-05

Load/2

max-upper-bound: 0.000336

median-lower-bound: 8.3e-05

median-upper-bound: 0.000332

min-lower-bound: 8.3e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.00017
median-lower-bound: 4.2e-05
median-upper-bound: 0.000168
min-lower-bound: 4.2e-05

FullCache

max-upper-bound: 9.8e-05
median-lower-bound: 1.6e-05
median-upper-bound: 6.4e-05
min-lower-bound: 1.55e-05

HotCache

max-upper-bound: 6e-05
median-lower-bound: 1.45e-05
median-upper-bound: 5.8e-05
min-lower-bound: 1.45e-05

Load/1

max-upper-bound: 0.000176

median-lower-bound: 4.4e-05

median-upper-bound: 0.000176

min-lower-bound: 4.35e-05

[sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=](#)

DirtyCache

max-upper-bound: 0.000105

median-lower-bound: 2.625e-05

median-upper-bound: 0.000105

min-lower-bound: 2.625e-05

FullCache

max-upper-bound: 0.00010296

median-lower-bound: 2.568e-05

median-upper-bound: 0.00010272

min-lower-bound: 2.564e-05

HotCache

max-upper-bound: 0.00010284

median-lower-bound: 2.571e-05

median-upper-bound: 0.00010284

min-lower-bound: 2.563e-05

Load/1

max-upper-bound: 0.00011816

median-lower-bound: 2.954e-05

median-upper-bound: 0.00011816

min-lower-bound: 2.946e-05

Load/2

max-upper-bound: 0.00011816

median-lower-bound: 2.954e-05

median-upper-bound: 0.00011816

min-lower-bound: 2.946e-05

Load/3

max-upper-bound: 0.00011832

median-lower-bound: 2.958e-05

median-upper-bound: 0.00011832

min-lower-bound: 2.934e-05

Load/4

max-upper-bound: 0.00011832

median-lower-bound: 2.958e-05

median-upper-bound: 0.00011832

min-lower-bound: 2.934e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 0.000112208

median-lower-bound: 2.7194e-05

median-upper-bound: 0.000108776

min-lower-bound: 2.621e-05

FullCache

max-upper-bound: 8.3136e-05

median-lower-bound: 1.2762e-05

median-upper-bound: 5.1048e-05

min-lower-bound: 1.2678e-05

HotCache

max-upper-bound: 2.4432e-05

median-lower-bound: 5.964e-06

median-upper-bound: 2.3856e-05

min-lower-bound: 5.964e-06

Load/1

max-upper-bound: 0.000161576

median-lower-bound: 2.6626e-05

median-upper-bound: 0.000106504

min-lower-bound: 2.6004e-05

Load/2

max-upper-bound: 0.000718024

median-lower-bound: 0.000178464

median-upper-bound: 0.000713856

min-lower-bound: 0.000178106

Load/3

max-upper-bound: 0.001148736

median-lower-bound: 0.000284654

median-upper-bound: 0.001138616

min-lower-bound: 0.00028356

Load/4

max-upper-bound: 0.001350688

median-lower-bound: 0.000336698

median-upper-bound: 0.001346792

min-lower-bound: 0.000334676

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.396e-05

median-lower-bound: 1.849e-05

median-upper-bound: 7.396e-05

min-lower-bound: 1.849e-05

FullCache

max-upper-bound: 7.192e-05

median-lower-bound: 1.798e-05

median-upper-bound: 7.192e-05

min-lower-bound: 1.764e-05

HotCache

max-upper-bound: 7.108e-05

median-lower-bound: 1.777e-05

median-upper-bound: 7.108e-05

min-lower-bound: 1.773e-05

Load/1

max-upper-bound: 8.192e-05

median-lower-bound: 2.048e-05

median-upper-bound: 8.192e-05

min-lower-bound: 2.048e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 6.7592e-05

median-lower-bound: 1.658e-05

median-upper-bound: 6.632e-05

min-lower-bound: 1.6494e-05

FullCache

max-upper-bound: 3.8328e-05

median-lower-bound: 6.492e-06

median-upper-bound: 2.5968e-05

min-lower-bound: 6.472e-06

HotCache

max-upper-bound: 1.1928e-05

median-lower-bound: 2.95e-06

median-upper-bound: 1.18e-05

min-lower-bound: 2.95e-06

Load/1

max-upper-bound: 7.1848e-05

median-lower-bound: 1.7368e-05

median-upper-bound: 6.9472e-05

min-lower-bound: 1.7292e-05

This requirement refines [spec:/rtems/sem/req/perf-runtime](#).

5.2.33 spec:/rtems/sem/req/perf-mtx-pi-wait-timed

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to rtems_semaphore_obtain which blocks on the mutex with a timeout. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the semaphore is a priority inheritance mutex, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 0.000116

median-lower-bound: 2.9e-05

median-upper-bound: 0.000116

min-lower-bound: 2.9e-05

FullCache

max-upper-bound: 0.000116

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.8e-05

HotCache

max-upper-bound: 0.000116

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.8e-05

Load/1

max-upper-bound: 0.000132

median-lower-bound: 3.25e-05

median-upper-bound: 0.00013

min-lower-bound: 3.25e-05

Load/2

max-upper-bound: 0.00013

median-lower-bound: 3.25e-05

median-upper-bound: 0.00013

min-lower-bound: 3.2e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000286

median-lower-bound: 7.15e-05

median-upper-bound: 0.000286

min-lower-bound: 7.1e-05

FullCache

max-upper-bound: 0.000168

median-lower-bound: 3.2e-05

median-upper-bound: 0.000128

min-lower-bound: 3.15e-05

HotCache

max-upper-bound: 0.00012
median-lower-bound: 3e-05
median-upper-bound: 0.00012
min-lower-bound: 2.95e-05

Load/1

max-upper-bound: 0.000294
median-lower-bound: 7.3e-05
median-upper-bound: 0.000292
min-lower-bound: 7.25e-05

Load/2

max-upper-bound: 0.000348
median-lower-bound: 8.6e-05
median-upper-bound: 0.000344
min-lower-bound: 8.55e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000178

median-lower-bound: 4.4e-05

median-upper-bound: 0.000176

min-lower-bound: 4.4e-05

FullCache

max-upper-bound: 0.000108

median-lower-bound: 1.7e-05

median-upper-bound: 6.8e-05

min-lower-bound: 1.65e-05

HotCache

max-upper-bound: 6.4e-05

median-lower-bound: 1.55e-05

median-upper-bound: 6.2e-05

min-lower-bound: 1.55e-05

Load/1

max-upper-bound: 0.000184

median-lower-bound: 4.6e-05

median-upper-bound: 0.000184

min-lower-bound: 4.55e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.0001118

median-lower-bound: 2.785e-05

median-upper-bound: 0.0001114

min-lower-bound: 2.773e-05

FullCache

max-upper-bound: 0.00010992

median-lower-bound: 2.748e-05

median-upper-bound: 0.00010992

min-lower-bound: 2.724e-05

HotCache

max-upper-bound: 0.00010884

median-lower-bound: 2.721e-05

median-upper-bound: 0.00010884

min-lower-bound: 2.721e-05

Load/1

max-upper-bound: 0.00012496

median-lower-bound: 3.116e-05

median-upper-bound: 0.00012464

min-lower-bound: 3.1e-05

Load/2

max-upper-bound: 0.00012496

median-lower-bound: 3.116e-05

median-upper-bound: 0.00012464

min-lower-bound: 3.1e-05

Load/3

max-upper-bound: 0.00012496

median-lower-bound: 3.116e-05

median-upper-bound: 0.00012464

min-lower-bound: 3.1e-05

Load/4

max-upper-bound: 0.00012496

median-lower-bound: 3.116e-05

median-upper-bound: 0.00012464

min-lower-bound: 3.1e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFlZ2eLDfIYc=

DirtyCache

max-upper-bound: 0.000120264

median-lower-bound: 2.7956e-05

median-upper-bound: 0.000111824

min-lower-bound: 2.6916e-05

FullCache

max-upper-bound: 8.8176e-05

median-lower-bound: 1.3208e-05

median-upper-bound: 5.2832e-05

min-lower-bound: 1.3148e-05

HotCache

max-upper-bound: 2.6376e-05

median-lower-bound: 6.526e-06

median-upper-bound: 2.6104e-05

min-lower-bound: 6.526e-06

Load/1

max-upper-bound: 0.000162904

median-lower-bound: 2.7162e-05

median-upper-bound: 0.000108648

min-lower-bound: 2.6772e-05

Load/2

max-upper-bound: 0.000741368

median-lower-bound: 0.000184708

median-upper-bound: 0.000738832

min-lower-bound: 0.000184256

Load/3

max-upper-bound: 0.001186688

median-lower-bound: 0.000294466

median-upper-bound: 0.001177864

min-lower-bound: 0.000293062

Load/4

max-upper-bound: 0.001400728

median-lower-bound: 0.000348276

median-upper-bound: 0.001393104

min-lower-bound: 0.000347286

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.68e-05

median-lower-bound: 1.92e-05

median-upper-bound: 7.68e-05

min-lower-bound: 1.92e-05

FullCache

max-upper-bound: 7.58e-05

median-lower-bound: 1.883e-05

median-upper-bound: 7.532e-05

min-lower-bound: 1.869e-05

HotCache

max-upper-bound: 7.616e-05

median-lower-bound: 1.904e-05

median-upper-bound: 7.616e-05

min-lower-bound: 1.888e-05

Load/1

max-upper-bound: 8.524e-05

median-lower-bound: 2.131e-05

median-upper-bound: 8.524e-05

min-lower-bound: 2.111e-05

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 7.2024e-05

median-lower-bound: 1.7732e-05

median-upper-bound: 7.0928e-05

min-lower-bound: 1.7626e-05

FullCache

max-upper-bound: 4.2584e-05

median-lower-bound: 6.806e-06

median-upper-bound: 2.7224e-05

min-lower-bound: 6.752e-06

HotCache

max-upper-bound: 1.2608e-05

median-lower-bound: 3.12e-06

median-upper-bound: 1.248e-05

min-lower-bound: 3.12e-06

Load/1

max-upper-bound: 7.3976e-05

median-lower-bound: 1.8364e-05

median-upper-bound: 7.3456e-05

min-lower-bound: 1.8308e-05

This requirement refines *spec:/rtems/sem/req/perf-runtime*.

5.2.34 spec:/rtems/sem/req/perf-runtime

spec:/rtems/sem/req/perf-runtime

The runtime of @ref RTEMSAPIClassicSem directives shall be measured.

rationale: N/A

This requirement refines *spec:/rtems/sem/req/perf-runtime*.

This requirement refines *spec:/rtems/sem/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/sem/req/perf-mtx-pi-obtain*
- *spec:/rtems/sem/req/perf-mtx-pi-release*
- *spec:/rtems/sem/req/perf-mtx-pi-release-one*
- *spec:/rtems/sem/req/perf-mtx-pi-release-other-cpu*
- *spec:/rtems/sem/req/perf-mtx-pi-release-preempt*
- *spec:/rtems/sem/req/perf-mtx-pi-try*
- *spec:/rtems/sem/req/perf-mtx-pi-wait-forever*
- *spec:/rtems/sem/req/perf-mtx-pi-wait-timed*

5.2.35 spec:/rtems/task/req/perf-construct

While the execution environment is below environments, while the measurement sample is the runtime of exactly one successful call to rtems_task_construct, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000166

median-lower-bound: 4.15e-05

median-upper-bound: 0.000166

min-lower-bound: 4.1e-05

FullCache

max-upper-bound: 0.000164

median-lower-bound: 4.1e-05

median-upper-bound: 0.000164

min-lower-bound: 4.1e-05

HotCache

max-upper-bound: 0.000164

median-lower-bound: 4.1e-05

median-upper-bound: 0.000164

min-lower-bound: 4.1e-05

Load/1

max-upper-bound: 0.00018

median-lower-bound: 4.45e-05

median-upper-bound: 0.000178

min-lower-bound: 4.45e-05

Load/2

max-upper-bound: 0.000178

median-lower-bound: 4.45e-05

median-upper-bound: 0.000178

min-lower-bound: 4.4e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.00027

median-lower-bound: 6.65e-05

median-upper-bound: 0.000266

min-lower-bound: 6.6e-05

FullCache

max-upper-bound: 0.000196

median-lower-bound: 4.1e-05

median-upper-bound: 0.000164

min-lower-bound: 4.1e-05

HotCache

max-upper-bound: 0.00016

median-lower-bound: 4e-05

median-upper-bound: 0.00016

min-lower-bound: 3.95e-05

Load/1

max-upper-bound: 0.000282

median-lower-bound: 7e-05

median-upper-bound: 0.00028

min-lower-bound: 6.95e-05

Load/2

max-upper-bound: 0.000368

median-lower-bound: 9.1e-05

median-upper-bound: 0.000364

min-lower-bound: 9.1e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000162

median-lower-bound: 4e-05

median-upper-bound: 0.00016

min-lower-bound: 4e-05

FullCache

max-upper-bound: 0.000102

median-lower-bound: 1.8e-05

median-upper-bound: 7.2e-05

min-lower-bound: 1.75e-05

HotCache

max-upper-bound: 7e-05

median-lower-bound: 1.75e-05

median-upper-bound: 7e-05

min-lower-bound: 1.7e-05

Load/1

max-upper-bound: 0.000178

median-lower-bound: 4.4e-05

median-upper-bound: 0.000176

min-lower-bound: 4.35e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcl0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.000151

median-lower-bound: 3.759e-05

median-upper-bound: 0.00015036

min-lower-bound: 3.751e-05

FullCache

max-upper-bound: 0.00015052

median-lower-bound: 3.76e-05

median-upper-bound: 0.0001504

min-lower-bound: 3.743e-05

HotCache

max-upper-bound: 0.00014976

median-lower-bound: 3.744e-05

median-upper-bound: 0.00014976

min-lower-bound: 3.744e-05

Load/1

max-upper-bound: 0.0001626

median-lower-bound: 4.065e-05

median-upper-bound: 0.0001626

min-lower-bound: 4.049e-05

Load/2

max-upper-bound: 0.0001618

median-lower-bound: 4.045e-05

median-upper-bound: 0.0001618

min-lower-bound: 4.041e-05

Load/3

max-upper-bound: 0.0001626

median-lower-bound: 4.065e-05

median-upper-bound: 0.0001626

min-lower-bound: 4.049e-05

Load/4

max-upper-bound: 0.0001626

median-lower-bound: 4.065e-05

median-upper-bound: 0.0001626

min-lower-bound: 4.049e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFlZ2eLDflYc=

DirtyCache

max-upper-bound: 9.1664e-05

median-lower-bound: 2.2482e-05

median-upper-bound: 8.9928e-05

min-lower-bound: 2.1906e-05

FullCache

max-upper-bound: 9.3696e-05

median-lower-bound: 1.3934e-05

median-upper-bound: 5.5736e-05

min-lower-bound: 1.3726e-05

HotCache

max-upper-bound: 3.7592e-05

median-lower-bound: 6.908e-06

median-upper-bound: 2.7632e-05

min-lower-bound: 6.864e-06

Load/1

max-upper-bound: 0.000156224

median-lower-bound: 2.3548e-05

median-upper-bound: 9.4192e-05

min-lower-bound: 2.2446e-05

Load/2

max-upper-bound: 0.00074872

median-lower-bound: 0.00018456

median-upper-bound: 0.00073824

min-lower-bound: 0.00018355

Load/3

max-upper-bound: 0.00129904

median-lower-bound: 0.000321152

median-upper-bound: 0.001284608

min-lower-bound: 0.000318308

Load/4

max-upper-bound: 0.001719288

median-lower-bound: 0.00042442

median-upper-bound: 0.00169768

min-lower-bound: 0.000422798

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 7.816e-05
median-lower-bound: 1.954e-05
median-upper-bound: 7.816e-05
min-lower-bound: 1.916e-05

FullCache

max-upper-bound: 7.784e-05
median-lower-bound: 1.946e-05
median-upper-bound: 7.784e-05
min-lower-bound: 1.941e-05

HotCache

max-upper-bound: 7.716e-05
median-lower-bound: 1.929e-05
median-upper-bound: 7.716e-05
min-lower-bound: 1.929e-05

Load/1

max-upper-bound: 8.984e-05
median-lower-bound: 2.24e-05
median-upper-bound: 8.96e-05
min-lower-bound: 2.208e-05

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 6.4072e-05
median-lower-bound: 1.4646e-05
median-upper-bound: 5.8584e-05
min-lower-bound: 1.4472e-05

FullCache

max-upper-bound: 4.368e-05
median-lower-bound: 6.198e-06
median-upper-bound: 2.4792e-05
min-lower-bound: 6.068e-06

HotCache

max-upper-bound: 1.756e-05
median-lower-bound: 3.358e-06

median-upper-bound: 1.3432e-05

min-lower-bound: 3.358e-06

Load/1

max-upper-bound: 6.7776e-05

median-lower-bound: 1.6114e-05

median-upper-bound: 6.4456e-05

min-lower-bound: 1.59e-05

This requirement refines [spec:/rtems/task/req/perf-runtime](#).

5.2.36 spec:/rtems/task/req/perf-restart

While the execution environment is below environments, while the measurement sample is the runtime of exactly one successful call to rtems_task_restart which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 8.6e-05

median-lower-bound: 2.15e-05

median-upper-bound: 8.6e-05

min-lower-bound: 2.1e-05

FullCache

max-upper-bound: 8.2e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 2e-05

HotCache

max-upper-bound: 8.2e-05

median-lower-bound: 2.05e-05

median-upper-bound: 8.2e-05

min-lower-bound: 2e-05

Load/1

max-upper-bound: 9.8e-05

median-lower-bound: 2.45e-05

median-upper-bound: 9.8e-05

min-lower-bound: 2.3e-05

Load/2

max-upper-bound: 9.8e-05

median-lower-bound: 2.4e-05

median-upper-bound: 9.6e-05

min-lower-bound: 2.3e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000216

median-lower-bound: 5.3e-05

median-upper-bound: 0.000212

min-lower-bound: 5.3e-05

FullCache

max-upper-bound: 0.000174

median-lower-bound: 1.65e-05

median-upper-bound: 6.6e-05

min-lower-bound: 1.65e-05

HotCache

max-upper-bound: 6.2e-05

median-lower-bound: 1.5e-05

median-upper-bound: 6e-05

min-lower-bound: 1.5e-05

Load/1

max-upper-bound: 0.000224

median-lower-bound: 5.6e-05

median-upper-bound: 0.000224

min-lower-bound: 5.55e-05

Load/2

max-upper-bound: 0.000258

median-lower-bound: 6.4e-05

median-upper-bound: 0.000256

min-lower-bound: 6.2e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 0.0
median-upper-bound: 0.0
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000134
median-lower-bound: 3.3e-05
median-upper-bound: 0.000132
min-lower-bound: 3.25e-05

FullCache

max-upper-bound: 0.000116
median-lower-bound: 9.5e-06
median-upper-bound: 3.8e-05
min-lower-bound: 9e-06

HotCache

max-upper-bound: 3.8e-05
median-lower-bound: 9e-06
median-upper-bound: 3.6e-05
min-lower-bound: 9e-06

Load/1

max-upper-bound: 0.000144
median-lower-bound: 3.6e-05
median-upper-bound: 0.000144
min-lower-bound: 3.45e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.192e-05
median-lower-bound: 2.046e-05
median-upper-bound: 8.184e-05
min-lower-bound: 2.036e-05

FullCache

max-upper-bound: 7.824e-05
median-lower-bound: 1.943e-05
median-upper-bound: 7.772e-05
min-lower-bound: 1.943e-05

HotCache

max-upper-bound: 7.792e-05
median-lower-bound: 1.948e-05
median-upper-bound: 7.792e-05
min-lower-bound: 1.948e-05

Load/1

max-upper-bound: 9.352e-05
median-lower-bound: 2.328e-05
median-upper-bound: 9.312e-05
min-lower-bound: 2.215e-05

Load/2

max-upper-bound: 9.352e-05
median-lower-bound: 2.328e-05

median-upper-bound: 9.312e-05

min-lower-bound: 2.213e-05

Load/3

max-upper-bound: 9.352e-05

median-lower-bound: 2.322e-05

median-upper-bound: 9.288e-05

min-lower-bound: 2.223e-05

Load/4

max-upper-bound: 9.352e-05

median-lower-bound: 2.322e-05

median-upper-bound: 9.288e-05

min-lower-bound: 2.223e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 8.8136e-05

median-lower-bound: 2.0934e-05

median-upper-bound: 8.3736e-05

min-lower-bound: 2.0448e-05

FullCache

max-upper-bound: 7.3936e-05

median-lower-bound: 6.13e-06

median-upper-bound: 2.452e-05

min-lower-bound: 6.106e-06

HotCache

max-upper-bound: 1.6152e-05

median-lower-bound: 3.91e-06

median-upper-bound: 1.564e-05

min-lower-bound: 3.91e-06

Load/1

max-upper-bound: 0.000132472

median-lower-bound: 2.1702e-05

median-upper-bound: 8.6808e-05

min-lower-bound: 2.068e-05

Load/2

max-upper-bound: 0.000525616
median-lower-bound: 0.000129916
median-upper-bound: 0.000519664
min-lower-bound: 0.000126052

Load/3

max-upper-bound: 0.00080428
median-lower-bound: 0.000199864
median-upper-bound: 0.000799456
min-lower-bound: 0.00018731

Load/4

max-upper-bound: 0.000929216
median-lower-bound: 0.000229488
median-upper-bound: 0.000917952
min-lower-bound: 0.000220234

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 5.184e-05
median-lower-bound: 1.291e-05
median-upper-bound: 5.164e-05
min-lower-bound: 1.291e-05

FullCache

max-upper-bound: 4.86e-05
median-lower-bound: 1.198e-05
median-upper-bound: 4.792e-05
min-lower-bound: 1.19e-05

HotCache

max-upper-bound: 4.788e-05
median-lower-bound: 1.197e-05
median-upper-bound: 4.788e-05
min-lower-bound: 1.189e-05

Load/1

max-upper-bound: 6.372e-05

median-lower-bound: 1.565e-05

median-upper-bound: 6.26e-05

min-lower-bound: 1.518e-05

sparc/gr740/uni/qual-only/c1ZkBOsUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.6328e-05

median-lower-bound: 1.323e-05

median-upper-bound: 5.292e-05

min-lower-bound: 1.3156e-05

FullCache

max-upper-bound: 4.6808e-05

median-lower-bound: 4.326e-06

median-upper-bound: 1.7304e-05

min-lower-bound: 4.306e-06

HotCache

max-upper-bound: 8.464e-06

median-lower-bound: 2.116e-06

median-upper-bound: 8.464e-06

min-lower-bound: 2.116e-06

Load/1

max-upper-bound: 6.1064e-05

median-lower-bound: 1.429e-05

median-upper-bound: 5.716e-05

min-lower-bound: 1.423e-05

This requirement refines [spec:/rtems/task/req/perf-runtime](#).

5.2.37 spec:/rtems/task/req/perf-restart-preempt

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to `rtems_task_restart` which starts task V which preempts the caller. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the measurement sample is $E - B$, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000156
median-lower-bound: 3.9e-05
median-upper-bound: 0.000156
min-lower-bound: 3.9e-05

FullCache

max-upper-bound: 0.000156
median-lower-bound: 3.85e-05
median-upper-bound: 0.000154
min-lower-bound: 3.8e-05

HotCache

max-upper-bound: 0.000156
median-lower-bound: 3.85e-05
median-upper-bound: 0.000154
min-lower-bound: 3.8e-05

Load/1

max-upper-bound: 0.000166
median-lower-bound: 4.15e-05
median-upper-bound: 0.000166
min-lower-bound: 4.1e-05

Load/2

max-upper-bound: 0.000166
median-lower-bound: 4.15e-05
median-upper-bound: 0.000166
min-lower-bound: 4.1e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000326
median-lower-bound: 8.15e-05
median-upper-bound: 0.000326
min-lower-bound: 8.1e-05

FullCache

max-upper-bound: 0.000286
median-lower-bound: 3.55e-05
median-upper-bound: 0.000142
min-lower-bound: 3.5e-05

HotCache

max-upper-bound: 0.000134
median-lower-bound: 3.35e-05
median-upper-bound: 0.000134
min-lower-bound: 3.3e-05

Load/1

max-upper-bound: 0.000322
median-lower-bound: 8e-05
median-upper-bound: 0.00032
min-lower-bound: 7.95e-05

Load/2

max-upper-bound: 0.000382
median-lower-bound: 9.35e-05
median-upper-bound: 0.000374
min-lower-bound: 9.3e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 5e-05

median-upper-bound: 0.0002

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 5e-05

median-upper-bound: 0.0002

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000212

median-lower-bound: 5.3e-05

median-upper-bound: 0.000212

min-lower-bound: 5.25e-05

FullCache

max-upper-bound: 0.000196

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.25e-05

HotCache

max-upper-bound: 8.4e-05

median-lower-bound: 2.1e-05

median-upper-bound: 8.4e-05

min-lower-bound: 2.05e-05

Load/1

max-upper-bound: 0.00022

median-lower-bound: 5.45e-05

median-upper-bound: 0.000218

min-lower-bound: 5.45e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.00014988

median-lower-bound: 3.715e-05

median-upper-bound: 0.0001486

min-lower-bound: 3.703e-05

FullCache

max-upper-bound: 0.00014644

median-lower-bound: 3.661e-05

median-upper-bound: 0.00014644

min-lower-bound: 3.659e-05

HotCache

max-upper-bound: 0.00014632

median-lower-bound: 3.658e-05

median-upper-bound: 0.00014632

min-lower-bound: 3.658e-05

Load/1

max-upper-bound: 0.00015744

median-lower-bound: 3.926e-05

median-upper-bound: 0.00015704

min-lower-bound: 3.9e-05

Load/2

max-upper-bound: 0.00015744

median-lower-bound: 3.926e-05

median-upper-bound: 0.00015704

min-lower-bound: 3.9e-05

Load/3

max-upper-bound: 0.00015744

median-lower-bound: 3.926e-05

median-upper-bound: 0.00015704

min-lower-bound: 3.9e-05

Load/4

max-upper-bound: 0.00015744

median-lower-bound: 3.926e-05

median-upper-bound: 0.00015704

min-lower-bound: 3.9e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 0.000131184
median-lower-bound: 3.1192e-05
median-upper-bound: 0.000124768
min-lower-bound: 3.0298e-05

FullCache

max-upper-bound: 0.000123064
median-lower-bound: 1.37e-05
median-upper-bound: 5.48e-05
min-lower-bound: 1.365e-05

HotCache

max-upper-bound: 3.336e-05
median-lower-bound: 8.238e-06
median-upper-bound: 3.2952e-05
min-lower-bound: 8.238e-06

Load/1

max-upper-bound: 0.000184344
median-lower-bound: 2.9632e-05
median-upper-bound: 0.000118528
min-lower-bound: 2.8994e-05

Load/2

max-upper-bound: 0.000826384
median-lower-bound: 0.00020054
median-upper-bound: 0.00080216
min-lower-bound: 0.000200332

Load/3

max-upper-bound: 0.001288688
median-lower-bound: 0.000314378
median-upper-bound: 0.001257512
min-lower-bound: 0.00031287

Load/4

max-upper-bound: 0.001554912
median-lower-bound: 0.000371362

median-upper-bound: 0.001485448

min-lower-bound: 0.000371018

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.00010084

median-lower-bound: 2.519e-05

median-upper-bound: 0.00010076

min-lower-bound: 2.519e-05

FullCache

max-upper-bound: 9.92e-05

median-lower-bound: 2.464e-05

median-upper-bound: 9.856e-05

min-lower-bound: 2.448e-05

HotCache

max-upper-bound: 9.812e-05

median-lower-bound: 2.453e-05

median-upper-bound: 9.812e-05

min-lower-bound: 2.449e-05

Load/1

max-upper-bound: 0.00010836

median-lower-bound: 2.709e-05

median-upper-bound: 0.00010836

min-lower-bound: 2.705e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 8.0904e-05

median-lower-bound: 1.9816e-05

median-upper-bound: 7.9264e-05

min-lower-bound: 1.9676e-05

FullCache

max-upper-bound: 7.06e-05

median-lower-bound: 8.264e-06

median-upper-bound: 3.3056e-05

min-lower-bound: 8.224e-06

HotCache

max-upper-bound: 1.8264e-05

median-lower-bound: 4.49e-06

median-upper-bound: 1.796e-05

min-lower-bound: 4.49e-06

Load/1

max-upper-bound: 8.4168e-05

median-lower-bound: 2.0602e-05

median-upper-bound: 8.2408e-05

min-lower-bound: 2.0482e-05

This requirement refines [*spec:/rtems/task/req/perf-runtime*](#).

5.2.38 spec:/rtems/task/req/perf-restart-self

Let B be a time point measured by a task right before a call to rtems_task_restart which restarts the caller. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the measurement sample is $E - B$, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.00016

median-lower-bound: 4e-05

median-upper-bound: 0.00016

min-lower-bound: 4e-05

FullCache

max-upper-bound: 0.000162

median-lower-bound: 4e-05

median-upper-bound: 0.00016

min-lower-bound: 4e-05

HotCache

max-upper-bound: 0.000162

median-lower-bound: 4.05e-05

median-upper-bound: 0.000162

min-lower-bound: 4e-05

Load/1

max-upper-bound: 0.000162

median-lower-bound: 4.05e-05

median-upper-bound: 0.000162

min-lower-bound: 4.05e-05

Load/2

max-upper-bound: 0.000162

median-lower-bound: 4.05e-05

median-upper-bound: 0.000162

min-lower-bound: 4e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.00024

median-lower-bound: 6e-05

median-upper-bound: 0.00024

min-lower-bound: 5.95e-05

FullCache

max-upper-bound: 0.00019

median-lower-bound: 3.4e-05

median-upper-bound: 0.000136

min-lower-bound: 3.35e-05

HotCache

max-upper-bound: 0.000132

median-lower-bound: 3.3e-05

median-upper-bound: 0.000132

min-lower-bound: 3.25e-05

Load/1

max-upper-bound: 0.000242

median-lower-bound: 6.05e-05

median-upper-bound: 0.000242

min-lower-bound: 6e-05

Load/2

max-upper-bound: 0.000284
median-lower-bound: 7.05e-05
median-upper-bound: 0.000282
min-lower-bound: 7.05e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002
median-lower-bound: 5e-05
median-upper-bound: 0.0002
min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000172
median-lower-bound: 4.25e-05
median-upper-bound: 0.00017
min-lower-bound: 4.25e-05

FullCache

max-upper-bound: 0.000124

median-lower-bound: 2.1e-05

median-upper-bound: 8.4e-05

min-lower-bound: 2.1e-05

HotCache

max-upper-bound: 8.2e-05

median-lower-bound: 2.05e-05

median-upper-bound: 8.2e-05

min-lower-bound: 2e-05

Load/1

max-upper-bound: 0.000172

median-lower-bound: 4.25e-05

median-upper-bound: 0.00017

min-lower-bound: 4.25e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.00015568

median-lower-bound: 3.892e-05

median-upper-bound: 0.00015568

min-lower-bound: 3.864e-05

FullCache

max-upper-bound: 0.00015568

median-lower-bound: 3.868e-05

median-upper-bound: 0.00015472

min-lower-bound: 3.852e-05

HotCache

max-upper-bound: 0.00015568

median-lower-bound: 3.892e-05

median-upper-bound: 0.00015568

min-lower-bound: 3.864e-05

Load/1

max-upper-bound: 0.0001552

median-lower-bound: 3.88e-05

median-upper-bound: 0.0001552

min-lower-bound: 3.88e-05

Load/2

max-upper-bound: 0.00015584

median-lower-bound: 3.896e-05

median-upper-bound: 0.00015584

min-lower-bound: 3.896e-05

Load/3

max-upper-bound: 0.00015584

median-lower-bound: 3.896e-05

median-upper-bound: 0.00015584

min-lower-bound: 3.896e-05

Load/4

max-upper-bound: 0.00015584

median-lower-bound: 3.896e-05

median-upper-bound: 0.00015584

min-lower-bound: 3.896e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 8.8128e-05

median-lower-bound: 2.1856e-05

median-upper-bound: 8.7424e-05

min-lower-bound: 2.1742e-05

FullCache

max-upper-bound: 9.2496e-05

median-lower-bound: 1.0216e-05

median-upper-bound: 4.0864e-05

min-lower-bound: 1.0206e-05

HotCache

max-upper-bound: 3.3648e-05

median-lower-bound: 8.344e-06

median-upper-bound: 3.3376e-05

min-lower-bound: 8.344e-06

Load/1

max-upper-bound: 0.000133184
median-lower-bound: 2.2052e-05
median-upper-bound: 8.8208e-05
min-lower-bound: 2.1962e-05

Load/2

max-upper-bound: 0.000636544
median-lower-bound: 0.00015723
median-upper-bound: 0.00062892
min-lower-bound: 0.000157052

Load/3

max-upper-bound: 0.001013752
median-lower-bound: 0.000251854
median-upper-bound: 0.001007416
min-lower-bound: 0.00025183

Load/4

max-upper-bound: 0.001230072
median-lower-bound: 0.000303726
median-upper-bound: 0.001214904
min-lower-bound: 0.000303236

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 9.784e-05
median-lower-bound: 2.434e-05
median-upper-bound: 9.736e-05
min-lower-bound: 2.408e-05

FullCache

max-upper-bound: 9.736e-05
median-lower-bound: 2.434e-05
median-upper-bound: 9.736e-05
min-lower-bound: 2.434e-05

HotCache

max-upper-bound: 9.736e-05

median-lower-bound: 2.434e-05

median-upper-bound: 9.736e-05

min-lower-bound: 2.434e-05

Load/1

max-upper-bound: 9.784e-05

median-lower-bound: 2.434e-05

median-upper-bound: 9.736e-05

min-lower-bound: 2.408e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 6.0552e-05

median-lower-bound: 1.5064e-05

median-upper-bound: 6.0256e-05

min-lower-bound: 1.4968e-05

FullCache

max-upper-bound: 3.7472e-05

median-lower-bound: 5.686e-06

median-upper-bound: 2.2744e-05

min-lower-bound: 5.682e-06

HotCache

max-upper-bound: 1.7608e-05

median-lower-bound: 4.338e-06

median-upper-bound: 1.7352e-05

min-lower-bound: 4.338e-06

Load/1

max-upper-bound: 6.0576e-05

median-lower-bound: 1.506e-05

median-upper-bound: 6.024e-05

min-lower-bound: 1.4954e-05

This requirement refines [spec:/rtems/task/req/perf-runtime](#).

5.2.39 spec:/rtems/task/req/perf-runtime

spec:/rtems/task/req/perf-runtime

The runtime of @ref RTEMSAPIClassicTasks directives shall be measured.

rationale: N/A

This requirement refines *spec:/rtems/req/perf-runtime*.

This requirement refines *spec:/rtems/task/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/task/req/perf-construct*
- *spec:/rtems/task/req/perf-restart*
- *spec:/rtems/task/req/perf-restart-preempt*
- *spec:/rtems/task/req/perf-restart-self*
- *spec:/rtems/task/req/perf-set-scheduler-move*
- *spec:/rtems/task/req/perf-set-scheduler-nop*
- *spec:/rtems/task/req/perf-set-scheduler-other*
- *spec:/rtems/task/req/perf-set-scheduler-preempt*
- *spec:/rtems/task/req/perf-start*
- *spec:/rtems/task/req/perf-start-preempt*

5.2.40 spec:/rtems/task/req/perf-set-scheduler-move

While the execution environment is below environments, while the measurement sample is the runtime of exactly one successful call to rtems_task_set_scheduler which changes the scheduler of the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000116

median-lower-bound: 2.85e-05

median-upper-bound: 0.000114

min-lower-bound: 2.8e-05

FullCache

max-upper-bound: 0.000108
median-lower-bound: 2.7e-05
median-upper-bound: 0.000108
min-lower-bound: 2.7e-05

HotCache

max-upper-bound: 0.000108
median-lower-bound: 2.7e-05
median-upper-bound: 0.000108
min-lower-bound: 2.7e-05

Load/1

max-upper-bound: 0.00013
median-lower-bound: 3.25e-05
median-upper-bound: 0.00013
min-lower-bound: 3.2e-05

Load/2

max-upper-bound: 0.00013
median-lower-bound: 3.25e-05
median-upper-bound: 0.00013
min-lower-bound: 3.2e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000338
median-lower-bound: 8.4e-05
median-upper-bound: 0.000336
min-lower-bound: 8.4e-05

FullCache

max-upper-bound: 0.000184
median-lower-bound: 3e-05
median-upper-bound: 0.00012
min-lower-bound: 3e-05

HotCache

max-upper-bound: 0.00012
median-lower-bound: 2.95e-05

median-upper-bound: 0.000118

min-lower-bound: 2.9e-05

Load/1

max-upper-bound: 0.000362

median-lower-bound: 9e-05

median-upper-bound: 0.00036

min-lower-bound: 8.75e-05

Load/2

max-upper-bound: 0.00041

median-lower-bound: 0.0001025

median-upper-bound: 0.00041

min-lower-bound: 0.000101

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 0.00010836

median-lower-bound: 2.703e-05

median-upper-bound: 0.00010812

min-lower-bound: 2.683e-05

FullCache

max-upper-bound: 0.0001044

median-lower-bound: 2.586e-05

median-upper-bound: 0.00010344

min-lower-bound: 2.578e-05

HotCache

max-upper-bound: 0.0001046

median-lower-bound: 2.615e-05

median-upper-bound: 0.0001046

min-lower-bound: 2.607e-05

Load/1

max-upper-bound: 0.0001234

median-lower-bound: 3.043e-05

median-upper-bound: 0.00012172

min-lower-bound: 3.034e-05

Load/2

max-upper-bound: 0.00012264
median-lower-bound: 3.054e-05
median-upper-bound: 0.00012216
min-lower-bound: 3.045e-05

Load/3

max-upper-bound: 0.00012264
median-lower-bound: 3.052e-05
median-upper-bound: 0.00012208
min-lower-bound: 3.041e-05

Load/4

max-upper-bound: 0.00012288
median-lower-bound: 3.05e-05
median-upper-bound: 0.000122
min-lower-bound: 3.033e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 0.00011996
median-lower-bound: 2.7476e-05
median-upper-bound: 0.000109904
min-lower-bound: 2.719e-05

FullCache

max-upper-bound: 0.00011556
median-lower-bound: 1.3236e-05
median-upper-bound: 5.2944e-05
min-lower-bound: 1.3172e-05

HotCache

max-upper-bound: 2.7168e-05
median-lower-bound: 6.552e-06
median-upper-bound: 2.6208e-05
min-lower-bound: 6.552e-06

Load/1

max-upper-bound: 0.000190232

median-lower-bound: 3.0796e-05

median-upper-bound: 0.000123184

min-lower-bound: 3.0084e-05

Load/2

max-upper-bound: 0.00064236

median-lower-bound: 0.00015924

median-upper-bound: 0.00063696

min-lower-bound: 0.000158816

Load/3

max-upper-bound: 0.00110592

median-lower-bound: 0.0002725

median-upper-bound: 0.00109

min-lower-bound: 0.00027201

Load/4

max-upper-bound: 0.00136536

median-lower-bound: 0.000340782

median-upper-bound: 0.001363128

min-lower-bound: 0.000336014

This requirement refines [*spec:/rtems/task/req/perf-runtime*](#).

5.2.41 spec:/rtems/task/req/perf-set-scheduler-nop

While the execution environment is below environments, while the measurement sample is the runtime of exactly one successful call to rtems_task_set_scheduler which does not change the scheduler or priority, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 9.2e-05

median-lower-bound: 2.25e-05

median-upper-bound: 9e-05

min-lower-bound: 2.25e-05

FullCache

max-upper-bound: 8.8e-05
median-lower-bound: 2.15e-05
median-upper-bound: 8.6e-05
min-lower-bound: 2.15e-05

HotCache

max-upper-bound: 8.8e-05
median-lower-bound: 2.15e-05
median-upper-bound: 8.6e-05
min-lower-bound: 2.15e-05

Load/1

max-upper-bound: 0.000112
median-lower-bound: 2.8e-05
median-upper-bound: 0.000112
min-lower-bound: 2.75e-05

Load/2

max-upper-bound: 0.000112
median-lower-bound: 2.75e-05
median-upper-bound: 0.00011
min-lower-bound: 2.75e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000244
median-lower-bound: 6.05e-05
median-upper-bound: 0.000242
min-lower-bound: 6.05e-05

FullCache

max-upper-bound: 0.00019
median-lower-bound: 1.85e-05
median-upper-bound: 7.4e-05
min-lower-bound: 1.85e-05

HotCache

max-upper-bound: 7.2e-05
median-lower-bound: 1.75e-05

median-upper-bound: 7e-05

min-lower-bound: 1.75e-05

Load/1

max-upper-bound: 0.000272

median-lower-bound: 6.8e-05

median-upper-bound: 0.000272

min-lower-bound: 6.75e-05

Load/2

max-upper-bound: 0.000324

median-lower-bound: 7.85e-05

median-upper-bound: 0.000314

min-lower-bound: 7.8e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 5.6e-05
median-lower-bound: 1.4e-05
median-upper-bound: 5.6e-05
min-lower-bound: 1.35e-05

FullCache

max-upper-bound: 4.6e-05
median-lower-bound: 4e-06
median-upper-bound: 1.6e-05
min-lower-bound: 3.5e-06

HotCache

max-upper-bound: 1.6e-05
median-lower-bound: 3.5e-06
median-upper-bound: 1.4e-05
min-lower-bound: 3.5e-06

Load/1

max-upper-bound: 6.8e-05
median-lower-bound: 1.7e-05
median-upper-bound: 6.8e-05
min-lower-bound: 1.65e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 8.796e-05
median-lower-bound: 2.199e-05
median-upper-bound: 8.796e-05
min-lower-bound: 2.195e-05

FullCache

max-upper-bound: 8.368e-05
median-lower-bound: 2.088e-05
median-upper-bound: 8.352e-05
min-lower-bound: 2.088e-05

HotCache

max-upper-bound: 8.372e-05

median-lower-bound: 2.093e-05

median-upper-bound: 8.372e-05

min-lower-bound: 2.085e-05

Load/1

max-upper-bound: 0.0001074

median-lower-bound: 2.685e-05

median-upper-bound: 0.0001074

min-lower-bound: 2.685e-05

Load/2

max-upper-bound: 0.0001074

median-lower-bound: 2.685e-05

median-upper-bound: 0.0001074

min-lower-bound: 2.685e-05

Load/3

max-upper-bound: 0.0001074

median-lower-bound: 2.685e-05

median-upper-bound: 0.0001074

min-lower-bound: 2.685e-05

Load/4

max-upper-bound: 0.0001074

median-lower-bound: 2.685e-05

median-upper-bound: 0.0001074

min-lower-bound: 2.685e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 9.3416e-05

median-lower-bound: 2.2962e-05

median-upper-bound: 9.1848e-05

min-lower-bound: 2.2446e-05

FullCache

max-upper-bound: 9.2504e-05

median-lower-bound: 7.334e-06

median-upper-bound: 2.9336e-05

min-lower-bound: 7.324e-06

HotCache

max-upper-bound: 1.8912e-05

median-lower-bound: 4.606e-06

median-upper-bound: 1.8424e-05

min-lower-bound: 4.606e-06

Load/1

max-upper-bound: 0.00015488

median-lower-bound: 2.5356e-05

median-upper-bound: 0.000101424

min-lower-bound: 2.423e-05

Load/2

max-upper-bound: 0.000658456

median-lower-bound: 0.000159664

median-upper-bound: 0.000638656

min-lower-bound: 0.00015957

Load/3

max-upper-bound: 0.00102956

median-lower-bound: 0.00025375

median-upper-bound: 0.001015

min-lower-bound: 0.000249198

Load/4

max-upper-bound: 0.001216392

median-lower-bound: 0.000292572

median-upper-bound: 0.001170288

min-lower-bound: 0.00029127

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 1.976e-05

median-lower-bound: 4.9e-06

median-upper-bound: 1.96e-05

min-lower-bound: 4.82e-06

FullCache

max-upper-bound: 1.924e-05
median-lower-bound: 4.81e-06
median-upper-bound: 1.924e-05
min-lower-bound: 4.79e-06

HotCache

max-upper-bound: 1.952e-05
median-lower-bound: 4.88e-06
median-upper-bound: 1.952e-05
min-lower-bound: 4.88e-06

Load/1

max-upper-bound: 2.948e-05
median-lower-bound: 7.33e-06
median-upper-bound: 2.932e-05
min-lower-bound: 7.19e-06

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFI2eLDfIYc=

DirtyCache

max-upper-bound: 2.2192e-05
median-lower-bound: 5.484e-06
median-upper-bound: 2.1936e-05
min-lower-bound: 5.428e-06

FullCache

max-upper-bound: 1.74e-05
median-lower-bound: 1.716e-06
median-upper-bound: 6.864e-06
min-lower-bound: 1.706e-06

HotCache

max-upper-bound: 3.336e-06
median-lower-bound: 8.34e-07
median-upper-bound: 3.336e-06
min-lower-bound: 8.34e-07

Load/1

max-upper-bound: 2.636e-05

median-lower-bound: 6.526e-06

median-upper-bound: 2.6104e-05

min-lower-bound: 6.46e-06

This requirement refines *spec:/rtems/task/req/perf-runtime*.

5.2.42 spec:/rtems/task/req/perf-set-scheduler-other

While the execution environment is below environments, while the measurement sample is the runtime of exactly one successful call to rtems_task_set_scheduler which moves the task to the home scheduler of the caller which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 3.8e-05

median-lower-bound: 9e-06

median-upper-bound: 3.6e-05

min-lower-bound: 9e-06

FullCache

max-upper-bound: 3.8e-05

median-lower-bound: 9e-06

median-upper-bound: 3.6e-05

min-lower-bound: 9e-06

HotCache

max-upper-bound: 3.8e-05

median-lower-bound: 9e-06

median-upper-bound: 3.6e-05

min-lower-bound: 9e-06

Load/1

max-upper-bound: 4.8e-05

median-lower-bound: 1.2e-05

median-upper-bound: 4.8e-05

min-lower-bound: 1.15e-05

Load/2

max-upper-bound: 4.8e-05

median-lower-bound: 1.15e-05

median-upper-bound: 4.6e-05

min-lower-bound: 1.15e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000118

median-lower-bound: 2.95e-05

median-upper-bound: 0.000118

min-lower-bound: 2.9e-05

FullCache

max-upper-bound: 8.6e-05

median-lower-bound: 7e-06

median-upper-bound: 2.8e-05

min-lower-bound: 7e-06

HotCache

max-upper-bound: 2.8e-05

median-lower-bound: 6.5e-06

median-upper-bound: 2.6e-05

min-lower-bound: 6.5e-06

Load/1

max-upper-bound: 0.000132

median-lower-bound: 3.3e-05

median-upper-bound: 0.000132

min-lower-bound: 3.25e-05

Load/2

max-upper-bound: 0.00015

median-lower-bound: 3.75e-05

median-upper-bound: 0.00015

min-lower-bound: 3.7e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVci0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 3.524e-05

median-lower-bound: 8.81e-06

median-upper-bound: 3.524e-05

min-lower-bound: 8.81e-06

FullCache

max-upper-bound: 3.512e-05

median-lower-bound: 8.72e-06

median-upper-bound: 3.488e-05

min-lower-bound: 8.72e-06

HotCache

max-upper-bound: 3.516e-05

median-lower-bound: 8.79e-06

median-upper-bound: 3.516e-05

min-lower-bound: 8.71e-06

Load/1

max-upper-bound: 4.472e-05

median-lower-bound: 1.118e-05

median-upper-bound: 4.472e-05

min-lower-bound: 1.118e-05

Load/2

max-upper-bound: 4.504e-05

median-lower-bound: 1.126e-05

median-upper-bound: 4.504e-05

min-lower-bound: 1.114e-05

Load/3

max-upper-bound: 4.472e-05

median-lower-bound: 1.118e-05

median-upper-bound: 4.472e-05

min-lower-bound: 1.118e-05

Load/4

max-upper-bound: 4.472e-05

median-lower-bound: 1.118e-05

median-upper-bound: 4.472e-05

min-lower-bound: 1.118e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 4.852e-05
median-lower-bound: 1.1518e-05
median-upper-bound: 4.6072e-05
min-lower-bound: 1.1026e-05

FullCache

max-upper-bound: 3.2592e-05
median-lower-bound: 3.48e-06
median-upper-bound: 1.392e-05
min-lower-bound: 3.466e-06

HotCache

max-upper-bound: 7.368e-06
median-lower-bound: 1.716e-06
median-upper-bound: 6.864e-06
min-lower-bound: 1.716e-06

Load/1

max-upper-bound: 8.4336e-05
median-lower-bound: 1.246e-05
median-upper-bound: 4.984e-05
min-lower-bound: 1.1598e-05

Load/2

max-upper-bound: 0.000268072
median-lower-bound: 6.4952e-05
median-upper-bound: 0.000259808
min-lower-bound: 6.4352e-05

Load/3

max-upper-bound: 0.000415408
median-lower-bound: 0.000103482
median-upper-bound: 0.000413928
min-lower-bound: 9.996e-05

Load/4

max-upper-bound: 0.000485824

median-lower-bound: 0.00011831

median-upper-bound: 0.00047324

min-lower-bound: 0.000117336

This requirement refines [spec:/rtems/task/req/perf-runtime](#).

5.2.43 spec:/rtems/task/req/perf-set-scheduler-preempt

Let U and V be two tasks with the distinct home schedulers. Let B be a time point measured by U right before a call to rtems_task_set_scheduler which moves task V to the home scheduler of U which preempts the caller. Let E be a time point measured by V right after the first context switch after B .

While the execution environment is below environments, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.000104

median-lower-bound: 2.6e-05

median-upper-bound: 0.000104

min-lower-bound: 2.6e-05

FullCache

max-upper-bound: 0.000102

median-lower-bound: 2.55e-05

median-upper-bound: 0.000102

min-lower-bound: 2.5e-05

HotCache

max-upper-bound: 0.000102

median-lower-bound: 2.55e-05

median-upper-bound: 0.000102

min-lower-bound: 2.5e-05

Load/1

max-upper-bound: 0.000114

median-lower-bound: 2.8e-05

median-upper-bound: 0.000112

min-lower-bound: 2.8e-05

Load/2

max-upper-bound: 0.000114

median-lower-bound: 2.8e-05

median-upper-bound: 0.000112

min-lower-bound: 2.8e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000254

median-lower-bound: 6.35e-05

median-upper-bound: 0.000254

min-lower-bound: 6.35e-05

FullCache

max-upper-bound: 0.000144

median-lower-bound: 2.2e-05

median-upper-bound: 8.8e-05

min-lower-bound: 2.2e-05

HotCache

max-upper-bound: 8.2e-05

median-lower-bound: 2e-05

median-upper-bound: 8e-05

min-lower-bound: 2e-05

Load/1

max-upper-bound: 0.000252

median-lower-bound: 6.3e-05

median-upper-bound: 0.000252

min-lower-bound: 6.25e-05

Load/2

max-upper-bound: 0.000254

median-lower-bound: 6.3e-05

median-upper-bound: 0.000252

min-lower-bound: 6.25e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcl0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 9.896e-05
median-lower-bound: 2.474e-05
median-upper-bound: 9.896e-05
min-lower-bound: 2.466e-05

FullCache

max-upper-bound: 9.724e-05
median-lower-bound: 2.427e-05
median-upper-bound: 9.708e-05
min-lower-bound: 2.414e-05

HotCache

max-upper-bound: 9.712e-05
median-lower-bound: 2.428e-05
median-upper-bound: 9.712e-05
min-lower-bound: 2.412e-05

Load/1

max-upper-bound: 0.00010684
median-lower-bound: 2.671e-05
median-upper-bound: 0.00010684
min-lower-bound: 2.671e-05

Load/2

max-upper-bound: 0.00010708
median-lower-bound: 2.677e-05
median-upper-bound: 0.00010708
min-lower-bound: 2.677e-05

Load/3

max-upper-bound: 0.000107
median-lower-bound: 2.675e-05
median-upper-bound: 0.000107
min-lower-bound: 2.675e-05

Load/4

max-upper-bound: 0.000107
median-lower-bound: 2.675e-05

median-upper-bound: 0.000107

min-lower-bound: 2.675e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 9.6896e-05

median-lower-bound: 2.2452e-05

median-upper-bound: 8.9808e-05

min-lower-bound: 2.2332e-05

FullCache

max-upper-bound: 5.752e-05

median-lower-bound: 9.792e-06

median-upper-bound: 3.9168e-05

min-lower-bound: 9.768e-06

HotCache

max-upper-bound: 1.9096e-05

median-lower-bound: 4.774e-06

median-upper-bound: 1.9096e-05

min-lower-bound: 4.774e-06

Load/1

max-upper-bound: 0.000118752

median-lower-bound: 2.1774e-05

median-upper-bound: 8.7096e-05

min-lower-bound: 2.1704e-05

Load/2

max-upper-bound: 0.000116392

median-lower-bound: 2.1798e-05

median-upper-bound: 8.7192e-05

min-lower-bound: 2.1674e-05

Load/3

max-upper-bound: 0.000549704

median-lower-bound: 0.000118466

median-upper-bound: 0.000473864

min-lower-bound: 0.000118102

Load/4

max-upper-bound: 0.00082376
median-lower-bound: 0.000177756
median-upper-bound: 0.000711024
min-lower-bound: 0.000176602

This requirement refines [spec:/rtems/task/req/perf-runtime](#).

5.2.44 spec:/rtems/task/req/perf-start

While the execution environment is below environments, while the measurement sample is the runtime of exactly one successful call to rtems_task_start which does not preempt the caller, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

[sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=](#)

DirtyCache

max-upper-bound: 5e-05
median-lower-bound: 1.25e-05
median-upper-bound: 5e-05
min-lower-bound: 1.25e-05

FullCache

max-upper-bound: 4.6e-05
median-lower-bound: 1.15e-05
median-upper-bound: 4.6e-05
min-lower-bound: 1.1e-05

HotCache

max-upper-bound: 4.6e-05
median-lower-bound: 1.15e-05
median-upper-bound: 4.6e-05
min-lower-bound: 1.1e-05

Load/1

max-upper-bound: 6.4e-05
median-lower-bound: 1.55e-05
median-upper-bound: 6.2e-05
min-lower-bound: 1.55e-05

Load/2

max-upper-bound: 6.4e-05
median-lower-bound: 1.55e-05
median-upper-bound: 6.2e-05
min-lower-bound: 1.55e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000142
median-lower-bound: 3.5e-05
median-upper-bound: 0.00014
min-lower-bound: 3.45e-05

FullCache

max-upper-bound: 7.4e-05
median-lower-bound: 1.55e-05
median-upper-bound: 6.2e-05
min-lower-bound: 1.5e-05

HotCache

max-upper-bound: 6.2e-05
median-lower-bound: 1.5e-05
median-upper-bound: 6e-05
min-lower-bound: 1.45e-05

Load/1

max-upper-bound: 0.000156
median-lower-bound: 3.8e-05
median-upper-bound: 0.000152
min-lower-bound: 3.75e-05

Load/2

max-upper-bound: 0.000178
median-lower-bound: 4.35e-05
median-upper-bound: 0.000174
min-lower-bound: 4.25e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 9e-05

median-lower-bound: 2.2e-05

median-upper-bound: 8.8e-05

min-lower-bound: 2.15e-05

FullCache

max-upper-bound: 6e-05

median-lower-bound: 8.5e-06

median-upper-bound: 3.4e-05

min-lower-bound: 8e-06

HotCache

max-upper-bound: 3.4e-05

median-lower-bound: 8e-06

median-upper-bound: 3.2e-05

min-lower-bound: 7.5e-06

Load/1

max-upper-bound: 0.000102

median-lower-bound: 2.55e-05

median-upper-bound: 0.000102

min-lower-bound: 2.5e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 4.816e-05

median-lower-bound: 1.196e-05

median-upper-bound: 4.784e-05

min-lower-bound: 1.194e-05

FullCache

max-upper-bound: 4.364e-05

median-lower-bound: 1.091e-05

median-upper-bound: 4.364e-05

min-lower-bound: 1.091e-05

HotCache

max-upper-bound: 4.376e-05

median-lower-bound: 1.094e-05

median-upper-bound: 4.376e-05

min-lower-bound: 1.094e-05

Load/1

max-upper-bound: 5.92e-05

median-lower-bound: 1.48e-05

median-upper-bound: 5.92e-05

min-lower-bound: 1.48e-05

Load/2

max-upper-bound: 5.92e-05

median-lower-bound: 1.48e-05

median-upper-bound: 5.92e-05

min-lower-bound: 1.48e-05

Load/3

max-upper-bound: 5.976e-05
median-lower-bound: 1.494e-05
median-upper-bound: 5.976e-05
min-lower-bound: 1.494e-05

Load/4

max-upper-bound: 5.976e-05
median-lower-bound: 1.494e-05
median-upper-bound: 5.976e-05
min-lower-bound: 1.494e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDflYc=

DirtyCache

max-upper-bound: 5.692e-05
median-lower-bound: 1.3832e-05
median-upper-bound: 5.5328e-05
min-lower-bound: 1.3136e-05

FullCache

max-upper-bound: 3.7328e-05
median-lower-bound: 7.062e-06
median-upper-bound: 2.8248e-05
min-lower-bound: 6.846e-06

HotCache

max-upper-bound: 1.4312e-05
median-lower-bound: 3.53e-06
median-upper-bound: 1.412e-05
min-lower-bound: 3.442e-06

Load/1

max-upper-bound: 9.212e-05
median-lower-bound: 1.4698e-05
median-upper-bound: 5.8792e-05
min-lower-bound: 1.3556e-05

Load/2

max-upper-bound: 0.000332176

median-lower-bound: 7.9596e-05

median-upper-bound: 0.000318384

min-lower-bound: 7.825e-05

Load/3

max-upper-bound: 0.000491864

median-lower-bound: 0.00012155

median-upper-bound: 0.0004862

min-lower-bound: 0.00011898

Load/4

max-upper-bound: 0.00058892

median-lower-bound: 0.0001444

median-upper-bound: 0.0005776

min-lower-bound: 0.000140016

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 3.324e-05

median-lower-bound: 8.31e-06

median-upper-bound: 3.324e-05

min-lower-bound: 8.23e-06

FullCache

max-upper-bound: 2.912e-05

median-lower-bound: 7.26e-06

median-upper-bound: 2.904e-05

min-lower-bound: 7.2e-06

HotCache

max-upper-bound: 2.9e-05

median-lower-bound: 7.25e-06

median-upper-bound: 2.9e-05

min-lower-bound: 7.21e-06

Load/1

max-upper-bound: 4.508e-05

median-lower-bound: 1.127e-05

median-upper-bound: 4.508e-05

min-lower-bound: 1.127e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 3.7784e-05

median-lower-bound: 9.236e-06

median-upper-bound: 3.6944e-05

min-lower-bound: 8.82e-06

FullCache

max-upper-bound: 2.6104e-05

median-lower-bound: 3.702e-06

median-upper-bound: 1.4808e-05

min-lower-bound: 3.518e-06

HotCache

max-upper-bound: 6.816e-06

median-lower-bound: 1.682e-06

median-upper-bound: 6.728e-06

min-lower-bound: 1.636e-06

Load/1

max-upper-bound: 4.288e-05

median-lower-bound: 1.0508e-05

median-upper-bound: 4.2032e-05

min-lower-bound: 1.0138e-05

This requirement refines [spec:/rtems/task/req/perf-runtime](#).

5.2.45 spec:/rtems/task/req/perf-start-preempt

Let U and V be two tasks with the same home scheduler. Let B be a time point measured by U right before a call to rtems_task_start which starts task V which preempts the caller. Let E be a time point measured by V right after the first context switch after B.

While the execution environment is below environments, while the measurement sample is $E - B$, when exactly 100 samples are collected, the below limit kinds shall be below limit conditions.

rationale: N/A

sparc/gr712rc/smp/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCeq9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 9.8e-05
median-lower-bound: 2.45e-05
median-upper-bound: 9.8e-05
min-lower-bound: 2.4e-05

FullCache

max-upper-bound: 9.6e-05
median-lower-bound: 2.35e-05
median-upper-bound: 9.4e-05
min-lower-bound: 2.35e-05

HotCache

max-upper-bound: 9.6e-05
median-lower-bound: 2.4e-05
median-upper-bound: 9.6e-05
min-lower-bound: 2.35e-05

Load/1

max-upper-bound: 0.000106
median-lower-bound: 2.65e-05
median-upper-bound: 0.000106
min-lower-bound: 2.65e-05

Load/2

max-upper-bound: 0.000106
median-lower-bound: 2.65e-05
median-upper-bound: 0.000106
min-lower-bound: 2.6e-05

sparc/gr712rc/smp/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.00023
median-lower-bound: 5.7e-05
median-upper-bound: 0.000228
min-lower-bound: 5.6e-05

FullCache

max-upper-bound: 0.000182

median-lower-bound: 2.65e-05

median-upper-bound: 0.000106

min-lower-bound: 2.55e-05

HotCache

max-upper-bound: 0.000108

median-lower-bound: 2.65e-05

median-upper-bound: 0.000106

min-lower-bound: 2.55e-05

Load/1

max-upper-bound: 0.000224

median-lower-bound: 5.55e-05

median-upper-bound: 0.000222

min-lower-bound: 5.5e-05

Load/2

max-upper-bound: 0.000274

median-lower-bound: 6.7e-05

median-upper-bound: 0.000268

min-lower-bound: 6.5e-05

sparc/gr712rc/uni/qual-only/5zrudgccHYC6azWEU3SRYUnkVXCe9MufBd5zmfMeEg=

DirtyCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

FullCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

HotCache

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

Load/1

max-upper-bound: 0.0002

median-lower-bound: 0.0

median-upper-bound: 0.0

min-lower-bound: 0.0

sparc/gr712rc/uni/qual-only/NDvyW8PTjD_thgV3P8TV_7MXVFP9EFvrevMzUX574lY=

DirtyCache

max-upper-bound: 0.000144

median-lower-bound: 3.5e-05

median-upper-bound: 0.00014

min-lower-bound: 3.4e-05

FullCache

max-upper-bound: 0.000122

median-lower-bound: 1.8e-05

median-upper-bound: 7.2e-05

min-lower-bound: 1.7e-05

HotCache

max-upper-bound: 7e-05

median-lower-bound: 1.7e-05

median-upper-bound: 6.8e-05

min-lower-bound: 1.55e-05

Load/1

max-upper-bound: 0.00015

median-lower-bound: 3.7e-05

median-upper-bound: 0.000148

min-lower-bound: 3.6e-05

sparc/gr740/smp/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 9.22e-05

median-lower-bound: 2.293e-05

median-upper-bound: 9.172e-05

min-lower-bound: 2.293e-05

FullCache

max-upper-bound: 8.992e-05
median-lower-bound: 2.248e-05
median-upper-bound: 8.992e-05
min-lower-bound: 2.248e-05

HotCache

max-upper-bound: 9.036e-05
median-lower-bound: 2.259e-05
median-upper-bound: 9.036e-05
min-lower-bound: 2.259e-05

Load/1

max-upper-bound: 9.952e-05
median-lower-bound: 2.488e-05
median-upper-bound: 9.952e-05
min-lower-bound: 2.488e-05

Load/2

max-upper-bound: 9.952e-05
median-lower-bound: 2.488e-05
median-upper-bound: 9.952e-05
min-lower-bound: 2.488e-05

Load/3

max-upper-bound: 0.00010048
median-lower-bound: 2.512e-05
median-upper-bound: 0.00010048
min-lower-bound: 2.512e-05

Load/4

max-upper-bound: 0.00010008
median-lower-bound: 2.502e-05
median-upper-bound: 0.00010008
min-lower-bound: 2.502e-05

sparc/gr740/smp/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFlZ2eLDfIYc=

DirtyCache

max-upper-bound: 8.7264e-05

median-lower-bound: 2.119e-05

median-upper-bound: 8.476e-05

min-lower-bound: 1.9954e-05

FullCache

max-upper-bound: 0.000101616

median-lower-bound: 1.153e-05

median-upper-bound: 4.612e-05

min-lower-bound: 1.1272e-05

HotCache

max-upper-bound: 2.404e-05

median-lower-bound: 5.986e-06

median-upper-bound: 2.3944e-05

min-lower-bound: 5.936e-06

Load/1

max-upper-bound: 0.000114688

median-lower-bound: 1.9606e-05

median-upper-bound: 7.8424e-05

min-lower-bound: 1.8946e-05

Load/2

max-upper-bound: 0.000541704

median-lower-bound: 0.000132084

median-upper-bound: 0.000528336

min-lower-bound: 0.000130782

Load/3

max-upper-bound: 0.000845152

median-lower-bound: 0.000208334

median-upper-bound: 0.000833336

min-lower-bound: 0.000204658

Load/4

max-upper-bound: 0.001045232

median-lower-bound: 0.000250862

median-upper-bound: 0.001003448

min-lower-bound: 0.000249272

sparc/gr740/uni/qual-only/_xQeTNJwSla2bVbhWPVcI0emLk2bE_GVQfvzt9CN84k=

DirtyCache

max-upper-bound: 6.244e-05
median-lower-bound: 1.561e-05
median-upper-bound: 6.244e-05
min-lower-bound: 1.557e-05

FullCache

max-upper-bound: 6.072e-05
median-lower-bound: 1.518e-05
median-upper-bound: 6.072e-05
min-lower-bound: 1.51e-05

HotCache

max-upper-bound: 6.084e-05
median-lower-bound: 1.521e-05
median-upper-bound: 6.084e-05
min-lower-bound: 1.513e-05

Load/1

max-upper-bound: 7.072e-05
median-lower-bound: 1.768e-05
median-upper-bound: 7.072e-05
min-lower-bound: 1.768e-05

sparc/gr740/uni/qual-only/c1ZkB0sUIJ-siPI7pK7knk0z6uni1pxOFIZ2eLDfIYc=

DirtyCache

max-upper-bound: 5.5664e-05
median-lower-bound: 1.3536e-05
median-upper-bound: 5.4144e-05
min-lower-bound: 1.3156e-05

FullCache

max-upper-bound: 4.704e-05
median-lower-bound: 6.65e-06
median-upper-bound: 2.66e-05
min-lower-bound: 6.38e-06

HotCache

max-upper-bound: 1.352e-05

median-lower-bound: 3.264e-06

median-upper-bound: 1.3056e-05

min-lower-bound: 3.19e-06

Load/1

max-upper-bound: 5.8496e-05

median-lower-bound: 1.4302e-05

median-upper-bound: 5.7208e-05

min-lower-bound: 1.3912e-05

This requirement refines [*spec:/rtems/task/req/perf-runtime*](#).

5.3 Interface requirements

See Software Interface Control Document (ICD)

5.4 Operational requirements

There are no operational requirements

5.5 Resources requirements

There are no resources requirements

5.6 Design requirements and implementation constraints

5.6.1 spec:/acfg/if/group

spec:/acfg/if/group

The API shall contain the application configuration interface.

rationale: N/A

identifier: RTEMSApplConfig

5.6.2 spec:/acfg/req/group

spec:/acfg/req/group

The implementation software architecture component have a component containing the application configuration implementation.

rationale: N/A

identifier: RTEMSImplApplConfig

This requirement refines *spec:/req/impl*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/appl-disable-filesystem*
- *spec:/acfg/req/appl-does-not-need-clock-driver*
- *spec:/acfg/req/appl-needs-clock-driver*
- *spec:/acfg/req/disable-bsp-settings*
- *spec:/acfg/req/disable-newlib-reentrancy*
- *spec:/acfg/req/idle-task-body*
- *spec:/acfg/req/idle-task-init-appl*
- *spec:/acfg/req/idle-task-stack-size*
- *spec:/acfg/req/init*
- *spec:/acfg/req/init-task-arguments*
- *spec:/acfg/req/init-task-attributes*
- *spec:/acfg/req/init-task-construct-storage-size*
- *spec:/acfg/req/init-task-entrypoint*
- *spec:/acfg/req/init-task-initial-modes*
- *spec:/acfg/req/init-task-name*
- *spec:/acfg/req/init-task-priority*
- *spec:/acfg/req/initial-extensions*
- *spec:/acfg/req/interrupt-stack-size*
- *spec:/acfg/req/max-barriers*
- *spec:/acfg/req/max-file-descriptors*
- *spec:/acfg/req/max-message-queues*
- *spec:/acfg/req/max-partitions*
- *spec:/acfg/req/max-periods*

- *spec:/acfg/req/max-priority*
- *spec:/acfg/req/max-processors*
- *spec:/acfg/req/max-semaphores*
- *spec:/acfg/req/max-tasks*
- *spec:/acfg/req/max-thread-local-storage-size*
- *spec:/acfg/req/max-timers*
- *spec:/acfg/req/max-user-extensions*
- *spec:/acfg/req/microseconds-per-tick*
- *spec:/acfg/req/min-task-stack-size*
- *spec:/acfg/req/min-tasks-with-user-provided-storage*
- *spec:/acfg/req/rtems-init-tasks-table*
- *spec:/acfg/req/scheduler-assignments*
- *spec:/acfg/req/scheduler-edf-smp*
- *spec:/acfg/req/scheduler-name*
- *spec:/acfg/req/scheduler-priority*
- *spec:/acfg/req/scheduler-table-entries*
- *spec:/acfg/req/task-stack-allocator*
- *spec:/acfg/req/task-stack-allocator-for-idle*
- *spec:/acfg/req/task-stack-deallocator*
- *spec:/acfg/req/task-stack-no-workspace*
- *spec:/acfg/req/ticks-per-time-slice*

5.6.3 spec:/bsp/if/group

spec:/bsp/if/group

The BSPs shall be a software architecture component containing the implementation specific to targets and devices.

rationale: N/A

identifier: RTEMSBSPs

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/if/group*

5.6.4 spec:/bsp/if/group-clock

spec:/bsp/if/group-clock

The shared BSP interrupt support shall be a component containing functions used to implement a Clock Driver.

rationale: N/A

identifier: bsp_clock

This requirement is refined by the following requirements:

- *spec:/bsp/req/mem-clock*

5.6.5 spec:/bsp/if/group-fatal

spec:/bsp/if/group-fatal

The fatal BSP support shall be a component containing functions used to indicate target-specific fatal errors.

rationale: N/A

identifier: RTEMSBSPsFatal

5.6.6 spec:/bsp/if/group-interrupt

spec:/bsp/if/group-interrupt

The shared BSP interrupt support shall be a component containing functions used to manage and process interrupts.

rationale: N/A

identifier: bsp_interrupt

5.6.7 spec:/bsp/if/group-shared

spec:/bsp/if/group-shared

The shared BSP support shall be a component containing functions used by multiple targets.

rationale: N/A

identifier: RTEMSBSPsShared

5.6.8 spec:/bsp/if/group-startup

spec:/bsp/if/group-startup

The shared BSP startup support shall be a component containing functions used to initialize the system.

rationale: N/A

identifier: RTEMSBSPsSharedStartup

5.6.9 spec:/bsp/sparc/if/group

spec:/bsp/sparc/if/group

The BSP component shall have a component containing the SPARC BSPs.

rationale: N/A

identifier: RTEMSBSPsSPARC

This requirement refines *spec:/bsp/if/group*.

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/leon3/if/group*

5.6.10 spec:/bsp/sparc/leon3/if/group

spec:/bsp/sparc/leon3/if/group

The SPARC BSP component shall have a component containing the LEON3 compatible BSPs.

rationale: N/A

identifier: RTEMSBSPsSPARCLEON3

This requirement refines [*spec:/bsp/sparc/if/group*](#).

This requirement is refined by the following requirements:

- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-01*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-02*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-03*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-04*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-05*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-06*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-07*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-08*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-09*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-10*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-11*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-12*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-13*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-14*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-15*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-16*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-17*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-18*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-19*](#)
- [*spec:/bsp/sparc/leon3/req/errata-gr712rc-20*](#)
- [*spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-boot*](#)
- [*spec:/bsp/sparc/leon3/req/fatal-cache-snooping-disabled-secondary*](#)
- [*spec:/bsp/sparc/leon3/req/fatal-clock-initialization*](#)

5.6.11 spec:/bsp/sparc/leon3/req/errata-b2bst-nop

spec:/bsp/sparc/leon3/req/errata-b2bst-nop

The software product shall use the define SPARC_LEON3FT_B2BST_NOP in assembler language source code to insert nop instructions required by the errata workaround.

rationale: N/A

This requirement refines [spec:/bsp/sparc/leon3/req/errata-gr712rc-13](#).

5.6.12 spec:/bsp/sparc/leon3/req/errata-gr712rc-01

spec:/bsp/sparc/leon3/req/errata-gr712rc-01

The software product shall take the errata 1 (FTAHBRA: On-chip Memory not cacheable) into account.

rationale: N/A

This requirement refines [spec:/req/errata](#).

This requirement refines [spec:/bsp/sparc/leon3/if/group](#).

5.6.13 spec:/bsp/sparc/leon3/req/errata-gr712rc-02

spec:/bsp/sparc/leon3/req/errata-gr712rc-02

The software product shall take the errata 2 (CAN OC: interrupt can be cleared before read) into account.

rationale: N/A

This requirement refines [spec:/req/errata](#).

This requirement refines [spec:/bsp/sparc/leon3/if/group](#).

5.6.14 spec:/bsp/sparc/leon3/req/errata-gr712rc-03

spec:/bsp/sparc/leon3/req/errata-gr712rc-03

The software product shall take the errata 3 (GRSPW2: interrupt can be lost) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.15 spec:/bsp/sparc/leon3/req/errata-gr712rc-04

spec:/bsp/sparc/leon3/req/errata-gr712rc-04

The software product shall take the errata 4 (GRSPW2: CRC calculation partially incorrect) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.16 spec:/bsp/sparc/leon3/req/errata-gr712rc-05

spec:/bsp/sparc/leon3/req/errata-gr712rc-05

The software product shall take the errata 5 (SPICTRL: transfers in progress bit not cleared) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.17 spec:/bsp/sparc/leon3/req/errata-gr712rc-06

spec:/bsp/sparc/leon3/req/errata-gr712rc-06

The software product shall take the errata 6 (SPICTRL: back-to-back transfers) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.18 spec:/bsp/sparc/leon3/req/errata-gr712rc-07

spec:/bsp/sparc/leon3/req/errata-gr712rc-07

The software product shall take the errata 7 (FTMCTLR: EDAC usage with 8-bit wide memory) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.19 spec:/bsp/sparc/leon3/req/errata-gr712rc-08

spec:/bsp/sparc/leon3/req/errata-gr712rc-08

The software product shall take the errata 8 (LEON3FT Cache Controller: Incorrect Bus Access After Power-Down) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.20 spec:/bsp/sparc/leon3/req/errata-gr712rc-09

spec:/bsp/sparc/leon3/req/errata-gr712rc-09

The software product shall take the errata 9 (Failing SDRAM Access After Uncorrectable EDAC Error) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.21 spec:/bsp/sparc/leon3/req/errata-gr712rc-10

spec:/bsp/sparc/leon3/req/errata-gr712rc-10

The software product shall take the errata 10 (MIL-STD-1553B core duplicate interrupt assertion) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.22 spec:/bsp/sparc/leon3/req/errata-gr712rc-11

spec:/bsp/sparc/leon3/req/errata-gr712rc-11

The software product shall take the errata 11 (Technical Note on LEON SRMMU Behaviour) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.23 spec:/bsp/sparc/leon3/req/errata-gr712rc-12

spec:/bsp/sparc/leon3/req/errata-gr712rc-12

The software product shall take the errata 12 (Technical Note on GRETH Ethernet Controller Behaviour) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.24 spec:/bsp/sparc/leon3/req/errata-gr712rc-13

spec:/bsp/sparc/leon3/req/errata-gr712rc-13

The software product shall take the errata 13 (Technical Note on Stale Cache Entry After Store with Data Tag Parity Error) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/leon3/req/errata-b2bst-nop*
- *spec:/bsp/sparc/leon3/req/errata-gr712rc-compiler-fix*

5.6.25 spec:/bsp/sparc/leon3/req/errata-gr712rc-14

spec:/bsp/sparc/leon3/req/errata-gr712rc-14

The software product shall take the errata 14 (Never disable the TLB when the MMU is enabled) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.26 spec:/bsp/sparc/leon3/req/errata-gr712rc-15

spec:/bsp/sparc/leon3/req/errata-gr712rc-15

The software product shall take the errata 15 (Technical Note on LEON3/FT AHB Lock Release During Atomic Operation) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.27 spec:/bsp/sparc/leon3/req/errata-gr712rc-16

spec:/bsp/sparc/leon3/req/errata-gr712rc-16

The software product shall take the errata 16 (Technical Note on GR712RC Incorrect Annulation of Floating-point Operation on) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/leon3/req/errata-gr712rc-compiler-fix*

5.6.28 spec:/bsp/sparc/leon3/req/errata-gr712rc-17

spec:/bsp/sparc/leon3/req/errata-gr712rc-17

The software product shall take the errata 17 (Technical Note on GRFPU Floating-point controller: Missing FDIV/FSQRT Result) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/leon3/req/errata-gr712rc-compiler-fix*

5.6.29 spec:/bsp/sparc/leon3/req/errata-gr712rc-18

spec:/bsp/sparc/leon3/req/errata-gr712rc-18

The software product shall take the errata 18 (Don't break into debug mode on RETT instructions) into account.

rationale: N/A

This requirement refines *spec:/req/errata*.

This requirement refines *spec:/bsp/sparc/leon3/if/group*.

5.6.30 spec:/bsp/sparc/leon3/req/errata-gr712rc-19

spec:/bsp/sparc/leon3/req/errata-gr712rc-19

The software product shall take the errata 19 (Stores to ASI 0x1C may update data cache) into account.

rationale: N/A

This requirement refines [spec:/req/errata](#).

This requirement refines [spec:/bsp/sparc/leon3/if/group](#).

5.6.31 spec:/bsp/sparc/leon3/req/errata-gr712rc-20

spec:/bsp/sparc/leon3/req/errata-gr712rc-20

The software product shall take the errata 20 (Technical Note on LEON3FT RETT Restart Errata) into account.

rationale: N/A

This requirement refines [spec:/req/errata](#).

This requirement refines [spec:/bsp/sparc/leon3/if/group](#).

This requirement is refined by the following requirements:

- [spec:/bsp/sparc/leon3/req/errata-gr712rc-compiler-fix](#)
- [spec:/bsp/sparc/leon3/req/errata-tn-0018-macros](#)

5.6.32 spec:/bsp/sparc/leon3/req/errata-gr712rc-compiler-fix

spec:/bsp/sparc/leon3/req/errata-gr712rc-compiler-fix

The software product shall be compiled using the -mfix-gr712rc compiler option.

rationale: This compiler option enables workarounds for chip errata.

This requirement refines [spec:/bsp/sparc/leon3/req/errata-gr712rc-13](#).

This requirement refines [spec:/bsp/sparc/leon3/req/errata-gr712rc-16](#).

This requirement refines [spec:/bsp/sparc/leon3/req/errata-gr712rc-17](#).

This requirement refines [spec:/bsp/sparc/leon3/req/errata-gr712rc-20](#).

5.6.33 spec:/bsp/sparc/leon3/req/errata-tn-0018-macros

spec:/bsp/sparc/leon3/req/errata-tn-0018-macros

The software product shall provide assembler language macros to help implementing workarounds for the errata.

rationale: N/A

This requirement refines *spec:/bsp/sparc/leon3/req/errata-gr712rc-20*.

This requirement is refined by the following requirements:

- *spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-fix*
- *spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-wait-iflush*
- *spec:/bsp/sparc/leon3/req/errata-tn-0018-macro-write-psr*

5.6.34 spec:/newlib/req/futex

spec:/newlib/req/futex

The super core shall have a component containing the futex implementation those interfaces are defined by Newlib.

rationale: With the futex support, highly efficient SMP synchronization primitives which offer random fairness can be provided by RTEMS. The futex support is required by the OpenMP implementation from GCC. It is used for barriers which are a core synchronization primitive of OpenMP.

identifier: RTEMSScoreFutex

This requirement is refined by the following requirements:

- *spec:/newlib/req/futex-wait*
- *spec:/newlib/req/futex-wake*

5.6.35 spec:/newlib/req/sys-lock

spec:/newlib/req/sys-lock

The super core shall have a component containing the system lock mutex implementation those interfaces are defined by Newlib.

rationale: The system lock mutexes are used for Newlib internal locking, the C++ standard library provided by GCC, the OpenMP support provided by GCC, the self-contained objects

API provided by RTEMS, and the super core internal mutexes such as the objects allocator lock.

identifier: RTEMSScoreSysLockMutex

This requirement is refined by the following requirements:

- *spec:/newlib/req/sys-lock-mutex-acquire*
- *spec:/newlib/req/sys-lock-mutex-acquire-timed*
- *spec:/newlib/req/sys-lock-mutex-recursive-acquire*
- *spec:/newlib/req/sys-lock-mutex-recursive-acquire-timed*
- *spec:/newlib/req/sys-lock-mutex-recursive-release*
- *spec:/newlib/req/sys-lock-mutex-recursive-try-acquire*
- *spec:/newlib/req/sys-lock-mutex-release*
- *spec:/newlib/req/sys-lock-mutex-try-acquire*

5.6.36 spec:/req/dev

spec:/req/dev

The software product shall have software architecture components containing the device drivers.

rationale: N/A

identifier: RTEMSDeviceDrivers

This requirement refines *spec:/req/root*.

5.6.37 spec:/req/domains

spec:/req/domains

The interfaces used or provided by the system shall be contained in interface domains.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/acfg/if/domain*
- *spec:/bsp/if/domain*
- *spec:/build-options/if/domain*

- spec:/c/if/domain
- spec:/compiler/if/domain
- spec:/dev/if/domain
- spec:/doc/if/domain
- spec:/if/domain
- spec:/newlib/if/domain
- spec:/score/if/domain
- spec:/user/if/domain

5.6.38 spec:/req/errata

spec:/req/errata

The software product shall take target errata into account.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- spec:/bsp/sparc/leon3/req/errata-gr712rc-01
- spec:/bsp/sparc/leon3/req/errata-gr712rc-02
- spec:/bsp/sparc/leon3/req/errata-gr712rc-03
- spec:/bsp/sparc/leon3/req/errata-gr712rc-04
- spec:/bsp/sparc/leon3/req/errata-gr712rc-05
- spec:/bsp/sparc/leon3/req/errata-gr712rc-06
- spec:/bsp/sparc/leon3/req/errata-gr712rc-07
- spec:/bsp/sparc/leon3/req/errata-gr712rc-08
- spec:/bsp/sparc/leon3/req/errata-gr712rc-09
- spec:/bsp/sparc/leon3/req/errata-gr712rc-10
- spec:/bsp/sparc/leon3/req/errata-gr712rc-11
- spec:/bsp/sparc/leon3/req/errata-gr712rc-12
- spec:/bsp/sparc/leon3/req/errata-gr712rc-13
- spec:/bsp/sparc/leon3/req/errata-gr712rc-14
- spec:/bsp/sparc/leon3/req/errata-gr712rc-15
- spec:/bsp/sparc/leon3/req/errata-gr712rc-16

- *spec:/bsp/sparc/leon3/req/errata-gr712rc-17*
- *spec:/bsp/sparc/leon3/req/errata-gr712rc-18*
- *spec:/bsp/sparc/leon3/req/errata-gr712rc-19*
- *spec:/bsp/sparc/leon3/req/errata-gr712rc-20*

5.6.39 spec:/req/fine-grained-locking

spec:/req/fine-grained-locking

If the system shall be implemented using fine grained locking at the lowest level.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/score/tq/req/lock*

5.6.40 spec:/req/impl

spec:/req/impl

The software product shall have software architecture components containing the implementation.

rationale: N/A

identifier: RTEMSImpl

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/acfg/req/group*
- *spec:/rtems/req/group*

5.6.41 spec:/req/root

spec:/req/root

The software product shall be a real-time operating system.

rationale: N/A

This requirement is refined by the following requirements:

- *spec:/bsp/if/group*
- *spec:/req/dev*
- *spec:/req/domains*
- *spec:/req/errata*
- *spec:/req/fine-grained-locking*
- *spec:/req/impl*
- *spec:/req/mem-benchmark*
- *spec:/req/perf-runtime*
- *spec:/req/terminate*
- *spec:/req/test-suites*
- *spec:/req/usage-constraints*

5.6.42 spec:/req/terminate

spec:/req/terminate

The system shall have a procedure to terminate the system.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/req/fatal-error*

5.6.43 spec:/req/test-suites

spec:/req/test-suites

The validation and unit tests shall be contained in test suites.

rationale: N/A

This requirement refines *spec:/req/root*.

5.6.44 spec:/req/usage-constraints

spec:/req/usage-constraints

The system shall document usage constraints of interfaces.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/acfg/constraint/clock-ticks-per-second*
- *spec:/acfg/constraint/imfs-memfile-bytes-per-block*
- *spec:/acfg/constraint/init-driver-table*
- *spec:/acfg/constraint/init-idle-task-body*
- *spec:/acfg/constraint/init-mpci-table*
- *spec:/acfg/constraint/init-posix-thread-entry-point*
- *spec:/acfg/constraint/init-task-entry-point*
- *spec:/acfg/constraint/init-task-stack-allocator*
- *spec:/acfg/constraint/init-task-stack-allocator-for-idle*
- *spec:/acfg/constraint/init-task-stack-allocator-init*
- *spec:/acfg/constraint/init-task-stack-deallocator*
- *spec:/acfg/constraint/init-userext-table*
- *spec:/acfg/constraint/max-clock-ticks*
- *spec:/acfg/constraint/max-drivers*
- *spec:/acfg/constraint/max-obj*
- *spec:/acfg/constraint/max-priority*
- *spec:/acfg/constraint/max-processors*
- *spec:/acfg/constraint/max-size*
- *spec:/acfg/constraint/max-tasks*
- *spec:/acfg/constraint/max-uint32*
- *spec:/acfg/constraint/max-uintptr*
- *spec:/acfg/constraint/memory-size*
- *spec:/acfg/constraint/min-16*
- *spec:/acfg/constraint/min-bsp-app*
- *spec:/acfg/constraint/min-clock-ticks*

- *spec:/acfg/constraint/min-one*
- *spec:/acfg/constraint/min-task-stack-size*
- *spec:/acfg/constraint/min-zero*
- *spec:/acfg/constraint/mpci-receive-server-stack*
- *spec:/acfg/constraint/multiple-of-bdbuf-buffer-min-size*
- *spec:/acfg/constraint/multiple-of-task-storage-alignment*
- *spec:/acfg/constraint/option-not-pre-qualified*
- *spec:/acfg/constraint/option-pre-qualified-mandatory*
- *spec:/acfg/constraint/option-pre-qualified-zero*
- *spec:/acfg/constraint posix-api*
- *spec:/acfg/constraint/power-of-two*
- *spec:/acfg/constraint/scheduler-assignments-count*
- *spec:/acfg/constraint/scheduler-assignments-macros*
- *spec:/acfg/constraint/scheduler-table-entry-count*
- *spec:/acfg/constraint/scheduler-table-entry-macros*
- *spec:/acfg/constraint/stack-space*
- *spec:/acfg/constraint/task-stack-from-alloc*
- *spec:/acfg/constraint/type-rtems-name*
- *spec:/acfg/constraint/type-task-argument*
- *spec:/acfg/constraint/unlimited*
- *spec:/acfg/constraint/unlimited-allocation-size*
- *spec:/acfg/constraint/use-task-storage-size*
- *spec:/acfg/constraint/valid-idle-stack-area*
- *spec:/acfg/constraint/valid-isr-stack-alignment*
- *spec:/acfg/constraint/valid-isr-stack-area*
- *spec:/acfg/constraint/valid-task-attributes*
- *spec:/acfg/constraint/valid-task-modes*
- *spec:/acfg/constraint/valid-task-priority*
- *spec:/acfg/constraint/wkspace*
- *spec:/constraint/affinity-may-preempt*
- *spec:/constraint/api-only*
- *spec:/constraint/clock-driver*
- *spec:/constraint/clock-tick*

- *spec:/constraint/constant-not-pre-qualified*
- *spec:/constraint/cpu-simple-vectored-interrupts*
- *spec:/constraint/directive-creator-only*
- *spec:/constraint/directive-ctx-any*
- *spec:/constraint/directive-ctx-const*
- *spec:/constraint/directive-ctx-devinit*
- *spec:/constraint/directive-ctx-fatal*
- *spec:/constraint/directive-ctx-isr*
- *spec:/constraint/directive-ctx-isr-local*
- *spec:/constraint/directive-ctx-task*
- *spec:/constraint/directive-no-preempt*
- *spec:/constraint/directive-no-preempt-local*
- *spec:/constraint/directive-no-return*
- *spec:/constraint/directive-not-pre-qualified*
- *spec:/constraint/directive-remote*
- *spec:/constraint/extensions-fatal*
- *spec:/constraint/fatal-bad-thread-dispatch-disable-level*
- *spec:/constraint/fatal-bad-thread-dispatch-environment*
- *spec:/constraint/interrupt-enable*
- *spec:/constraint/interrupts-disabled-smp*
- *spec:/constraint/mp-send*
- *spec:/constraint/no-atexit*
- *spec:/constraint/no-impl*
- *spec:/constraint/no-underscore-interfaces*
- *spec:/constraint/obj-unlimited-alloc*
- *spec:/constraint/obj-unlimited-free*
- *spec:/constraint/object-allocator*
- *spec:/constraint/object-allocator-not-owner*
- *spec:/constraint/object-allocator-owner*
- *spec:/constraint/priority-may-preempt*
- *spec:/constraint/request-response*
- *spec:/constraint/terminate*
- *spec:/constraint/thread-dispatching-disabled*

RTEMS Qualification Software Requirement Specification [sparc/gr712rc/smp/4]

Release 3

ESA Contract No. 4000125572/18/NL/GLC/as

- *spec:/constraint/thread-dispatching-disabled-init*
- *spec:/constraint/thread-dispatching-enabled*
- *spec:/constraint/thread-dispatching-enabled-multitasking*
- *spec:/constraint/thread-life-protected*
- *spec:/constraint/tls-alignment-power-of-2*
- *spec:/constraint/unblock-may-preempt*
- *spec:/rtems/barrier/constraint/max*
- *spec:/rtems/clock/constraint/tod-begin*
- *spec:/rtems/clock/constraint/tod-end*
- *spec:/rtems/constraint/delete-by-any-task*
- *spec:/rtems/constraint/directive-ctx-isr-no-wait*
- *spec:/rtems/constraint/mp-max-global-objects*
- *spec:/rtems/constraint/request-may-block*
- *spec:/rtems/dpmem/constraint/max*
- *spec:/rtems/init/constraint/directive-ctx-boot*
- *spec:/rtems/intr/constraint/entry-initialized*
- *spec:/rtems/intr/constraint/entry-installed*
- *spec:/rtems/intr/constraint/no-smp*
- *spec:/rtems/intr/constraint/not-within-server*
- *spec:/rtems/intr/constraint/server-entry-initialized*
- *spec:/rtems/intr/constraint/server-entry-not-while-action-prepend*
- *spec:/rtems/intr/constraint/server-entry-not-while-move*
- *spec:/rtems/intr/constraint/server-entry-not-while-pending*
- *spec:/rtems/intr/constraint/server-entry-not-while-submit*
- *spec:/rtems/intr/constraint/server-request-initialized*
- *spec:/rtems/intr/constraint/server-request-not-while-set-vector*
- *spec:/rtems/intr/constraint/server-request-not-while-submit*
- *spec:/rtems/io/constraint/no-floating-point*
- *spec:/rtems/message/constraint/max*
- *spec:/rtems/message/constraint/receive-isr*
- *spec:/rtems/part/constraint/max*
- *spec:/rtems/ratemon/constraint/max*
- *spec:/rtems/region/constraint/max*

- *spec:/rtems/sem/constraint/max*
- *spec:/rtems/sem/constraint/obtain-devinit*
- *spec:/rtems/sem/constraint/obtain-isr*
- *spec:/rtems/sem/constraint/release-devinit*
- *spec:/rtems/sem/constraint/release-isr*
- *spec:/rtems/task/constraint/max*
- *spec:/rtems/task/constraint/preempt-enable*
- *spec:/rtems/timer/constraint/max*
- *spec:/rtems/userext/constraint/max*
- *spec:/rtems/userext/constraint/object-allocator-pthread-create*
- *spec:/rtems/userext/constraint/object-allocator-system*
- *spec:/rtems/userext/constraint/object-allocator-task-construct*
- *spec:/rtems/userext/constraint/object-allocator-task-create*

5.6.45 spec:/rtems/attr/req/group

spec:/rtems/attr/req/group

The software architecture shall have a component containing the attribute support implementation.

rationale: N/A

identifier: RTEMSImplClassicAttr

This requirement refines *spec:/rtems/req/group*.

5.6.46 spec:/rtems/barrier/req/group

spec:/rtems/barrier/req/group

The software architecture shall have a component containing the Barrier Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicBarrier

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/barrier/req/mem-wait-rel*

- *spec:/rtems/barrier/req/mem-wait-rel-del*
- *spec:/rtems/barrier/req/perf-runtime*

5.6.47 spec:/rtems/clock/req/group

spec:/rtems/clock/req/group

The software architecture shall have a component containing the Clock Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicClock

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/clock/req/mem-get-uptime*
- *spec:/rtems/clock/req/mem-set*
- *spec:/rtems/clock/req/mem-set-get-tod*

5.6.48 spec:/rtems/event/req/group

spec:/rtems/event/req/group

The software architecture shall have a component containing the Event Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicEvent

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/event/req/event-number*
- *spec:/rtems/event/req/mem-snd-rcv*
- *spec:/rtems/event/req/perf-runtime*
- *spec:/rtems/event/req/send-receive*

5.6.49 spec:/rtems/message/req/group

spec:/rtems/message/req/group

The software architecture shall have a component containing the Message Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicMessage

This requirement refines [spec:/rtems/req/group](#).

This requirement is refined by the following requirements:

- [spec:/rtems/message/req/mem-bcst-rcv](#)
- [spec:/rtems/message/req/mem-snd-rcv](#)
- [spec:/rtems/message/req/mem-snd-rcv-del](#)
- [spec:/rtems/message/req/mem-ugt-rcv](#)
- [spec:/rtems/message/req/perf-runtime](#)

5.6.50 spec:/rtems/part/req/group

spec:/rtems/part/req/group

The software architecture shall have a component containing the Partition Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicPartition

This requirement refines [spec:/rtems/req/group](#).

This requirement is refined by the following requirements:

- [spec:/rtems/part/req/mem-get-ret](#)
- [spec:/rtems/part/req/mem-get-ret-del](#)
- [spec:/rtems/part/req/perf-runtime](#)

5.6.51 spec:/rtems/ratemon/req/group

spec:/rtems/ratemon/req/group

The software architecture shall have a component containing the Rate Monotonic Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicRateMonotonic

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/ratemon/req/mem-period*
- *spec:/rtems/ratemon/req/mem-period-del*

5.6.52 spec:/rtems/req/group

spec:/rtems/req/group

The software architecture shall have a component containing the Classic API implementation.

rationale: N/A

identifier: RTEMSImplClassic

This requirement refines *spec:/req/impl*.

This requirement is refined by the following requirements:

- *spec:/rtems/attr/req/group*
- *spec:/rtems/barrier/req/group*
- *spec:/rtems/clock/req/group*
- *spec:/rtems/event/req/group*
- *spec:/rtems/fatal/req/mem-fatal*
- *spec:/rtems/message/req/group*
- *spec:/rtems/part/req/group*
- *spec:/rtems/ratemon/req/group*
- *spec:/rtems/req/ident*
- *spec:/rtems/req/ident-local*
- *spec:/rtems/req/mem-basic*
- *spec:/rtems/req/mem-smp-1*

- *spec:/rtems/req/mem-smp-global-2*
- *spec:/rtems/req/mem-smp-global-4*
- *spec:/rtems/req/mem-smp-part-2*
- *spec:/rtems/req/mem-smp-part-4*
- *spec:/rtems/req/perf-runtime*
- *spec:/rtems/scheduler/req/group*
- *spec:/rtems/scheduler/req/mem-add-cpu*
- *spec:/rtems/scheduler/req/mem-rm-cpu*
- *spec:/rtems/sem/req/group*
- *spec:/rtems/signal/req/group*
- *spec:/rtems/task/req/group*
- *spec:/rtems/timer/req/group*
- *spec:/rtems/userext/req/group*

5.6.53 spec:/rtems/scheduler/req/group

spec:/rtems/scheduler/req/group

The software architecture shall have a component containing the Scheduler Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicScheduler

This requirement refines *spec:/rtems/req/group*.

5.6.54 spec:/rtems/sem/req/group

spec:/rtems/sem/req/group

The software architecture shall have a component containing the Semaphore Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicSemaphore

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/sem/req/mem-obt-rel*

- *spec:/rtems/sem/req/mem-obt-rel-del*
- *spec:/rtems/sem/req/perf-runtime*

5.6.55 spec:/rtems/signal/req/group

spec:/rtems/signal/req/group

The software architecture shall have a component containing the Signal Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicSignal

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/signal/req/mem-catch-snd*
- *spec:/rtems/signal/req/signal-count*

5.6.56 spec:/rtems/task/req/group

spec:/rtems/task/req/group

The software architecture shall have a component containing the Task Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicTask

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/task/req/mem-delete*
- *spec:/rtems/task/req/mem-exit*
- *spec:/rtems/task/req/mem-get-affinity*
- *spec:/rtems/task/req/mem-get-priority*
- *spec:/rtems/task/req/mem-get-scheduler*
- *spec:/rtems/task/req/mem-mode*
- *spec:/rtems/task/req/mem-restart*
- *spec:/rtems/task/req/mem-set-affinity*
- *spec:/rtems/task/req/mem-set-priority*

- *spec:/rtems/task/req/mem-set-scheduler*
- *spec:/rtems/task/req/mem-sus-res*
- *spec:/rtems/task/req/mem-wake-after*
- *spec:/rtems/task/req/mem-wake-when*
- *spec:/rtems/task/req/perf-runtime*

5.6.57 spec:/rtems/timer/req/group

spec:/rtems/timer/req/group

The software architecture shall have a component containing the Timer Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicTimer

This requirement refines *spec:/rtems/req/group*.

This requirement is refined by the following requirements:

- *spec:/rtems/timer/req/mem-after*
- *spec:/rtems/timer/req/mem-cancel*
- *spec:/rtems/timer/req/mem-delete*
- *spec:/rtems/timer/req/mem-reset*
- *spec:/rtems/timer/req/mem-srv-after*
- *spec:/rtems/timer/req/mem-srv-init*
- *spec:/rtems/timer/req/mem-srv-when*
- *spec:/rtems/timer/req/mem-when*

5.6.58 spec:/rtems/userext/req/group

spec:/rtems/userext/req/group

The software architecture shall have a component containing the User Extensions Manager implementation.

rationale: N/A

identifier: RTEMSImplClassicUserExt

This requirement refines [*spec:/rtems/req/group*](#).

This requirement is refined by the following requirements:

- [*spec:/rtems/userext/req/mem-create*](#)
- [*spec:/rtems/userext/req/mem-delete*](#)

5.6.59 [*spec:/score/cpu/if/group*](#)

spec:/score/cpu/if/group

The super core shall have a component containing the implementation specific to the target architecture and target.

rationale: N/A

identifier: RTEMSScoreCPU

This requirement is refined by the following requirements:

- [*spec:/score/cpu/req/fatal-halt*](#)

5.6.60 [*spec:/score/if/group*](#)

spec:/score/if/group

The super core shall be a software architecture component containing the implementation used by multiple APIs.

rationale: N/A

identifier: RTEMSScore

5.6.61 [*spec:/score/interr/if/group*](#)

spec:/score/interr/if/group

The super core shall have a component containing the system termination handling.

rationale: N/A

identifier: RTEMSScoreIntErr

5.6.62 spec:/score mtx/if/group

spec:/score mtx/if/group

The super core shall have a component containing the mutual exclusion object implementation.

rationale: N/A

identifier: RTEMSScoreMutex

This requirement is refined by the following requirements:

- *spec:/score/mtx/req/seize-try*
- *spec:/score/mtx/req/seize-wait*
- *spec:/score/mtx/req/surrender*

5.6.63 spec:/score/object/if/group

spec:/score/object/if/group

The super core shall have a component containing the object services implementation.

rationale: N/A

identifier: RTEMSScoreThreadQueue

This requirement is refined by the following requirements:

- *spec:/score/object/req/allocator-mutex*

5.6.64 spec:/score/sched/req/group

spec:/score/sched/req/group

The super core shall have a component containing the scheduler implementations.

rationale: N/A

identifier: RTEMSScoreScheduler

This requirement is refined by the following requirements:

- *spec:/score/sched/req/yield*
- *spec:/score/sched/smp/req/group*

5.6.65 spec:/score/sched/smp/edf/req/group

spec:/score/sched/smp/edf/req/group

The super core shall have a component containing the EDF SMP /glossary/scheduler implementation.

rationale: N/A

identifier: RTEMSScoreSchedulerSMPEDF

This requirement refines *spec:/score/sched/smp/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/sched/smp/edf/req/set-affinity*
- *spec:/score/sched/smp/edf/req/set-affinity-unsupported-subset*

5.6.66 spec:/score/sched/smp/req/group

spec:/score/sched/smp/req/group

The super core shall have a component containing the SMP /glossary/scheduler implementations.

rationale: N/A

identifier: RTEMSScoreSchedulerSMP

This requirement refines *spec:/score/sched/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/sched/smp/edf/req/group*
- *spec:/score/sched/smp/req/ask-for-help-helping*
- *spec:/score/sched/smp/req/ask-for-help-home*
- *spec:/score/sched/smp/req/ask-for-help-request*
- *spec:/score/sched/smp/req/block-reuse-idle*
- *spec:/score/sched/smp/req/idle-to-scheduled*
- *spec:/score/sched/smp/req/preempt-blocked*

5.6.67 spec:/score/sem/if/group

spec:/score/sem/if/group

The super core shall have a component containing the semaphore object implementation.

rationale: N/A

identifier: RTEMSScoreSemaphore

This requirement is refined by the following requirements:

- *spec:/score/sem/req/seize-try*
- *spec:/score/sem/req/seize-wait*
- *spec:/score/sem/req/surrender*

5.6.68 spec:/score/smp/if/group

spec:/score/smp/if/group

The super core shall have a component containing the SMP-specific implementation.

rationale: N/A

identifier: RTEMSScoreSMP

This requirement is refined by the following requirements:

- *spec:/score/smp/req/fatal-boot-processor-not-assigned-to-scheduler*
- *spec:/score/smp/req/fatal-mandatory-processor-not-present*
- *spec:/score/smp/req/fatal-multitasking-start-on-invalid-processor*
- *spec:/score/smp/req/fatal-multitasking-start-on-unassigned-processor*
- *spec:/score/smp/req/fatal-scheduler-requires-exactly-one-processor*
- *spec:/score/smp/req/fatal-shutdown-response*
- *spec:/score/smp/req/fatal-start-of-mandatory-processor-failed*
- *spec:/score/smp/req/fatal-start-on-not-online-processor*
- *spec:/score/smp/req/fatal-wrong-cpu-state-to-perform-jobs*
- *spec:/score/smp/req/per-cpu-jobs-order*
- *spec:/score/smp/req/processor-online*
- *spec:/score/smp/req/start-of-optional-processor-failed-continue*
- *spec:/score/smp/req/start-of-processor-failed-not-online*

5.6.69 spec:/score/thread/if/group

spec:/score/thread/if/group

The super core shall have a component containing the thread implementation.

rationale: N/A

identifier: RTEMSScoreThreadQueue

This requirement is refined by the following requirements:

- *spec:/score/thread/req/cancel-killer*
- *spec:/score/thread/req/fatal-bad-thread-dispatch-disable-level*
- *spec:/score/thread/req/fatal-bad-thread-dispatch-environment-ipi*
- *spec:/score/thread/req/fatal-thread-exitted*
- *spec:/score/thread/req/pinning*
- *spec:/score/thread/req/suspended-helping*
- *spec:/score/thread/req/thread-dispatch-isr-disabled*
- *spec:/score/thread/req/tls-big-alignment*
- *spec:/score/thread/req/tls-initialized-data*
- *spec:/score/thread/req/tls-max-zero*
- *spec:/score/thread/req/tls-zero-initialized-data*

5.6.70 spec:/score/thread/req/pinning

spec:/score/thread/req/pinning

Pinning of threads to a processor shall be supported.

rationale: The pinning of threads to a processor is an important feature to support dynamically allocated lock-free data structures. It allows efficient and safe access to processor-specific data structures. It is used for example to implement the Epoch Based Reclamation in libbsd. The libbsd is a port of FreeBSD kernel modules to RTEMS.

This requirement refines *spec:/score/thread/if/group*.

This requirement is refined by the following requirements:

- *spec:/score/thread/req/pinning-helping*
- *spec:/score/thread/req/pinning-nested*
- *spec:/score/thread/req/pinning-preemptible*

- *spec:/score/thread/req/pinning-unpin-suspended*

5.6.71 spec:/score/timecounter/req/clock-monotonic

spec:/score/timecounter/req/clock-monotonic

The timecounter component shall provide the CLOCK_MONOTONIC.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/clock-monotonic-initial*
- *spec:/score/timecounter/req/clock-monotonic-initial-frozen*

5.6.72 spec:/score/timecounter/req/clock-realtime

spec:/score/timecounter/req/clock-realtime

The timecounter component shall provide the CLOCK_REALTIME.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/clock-realtime-initial*
- *spec:/score/timecounter/req/clock-realtime-initial-frozen*

5.6.73 spec:/score/timecounter/req/get

spec:/score/timecounter/req/get

The timecounter component shall implement directives to get time values in the highest resolution available to the system.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/get-device*

- *spec:/score/timecounter/req/get-large-delta*
- *spec:/score/timecounter/req/get-non-blocking*

5.6.74 spec:/score/timecounter/req/get-coarse

spec:/score/timecounter/req/get-coarse

The timecounter component shall implement directives to get time values in a coarse resolution.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/get-coarse-no-device*
- *spec:/score/timecounter/req/get-non-blocking*

5.6.75 spec:/score/timecounter/req/get-non-blocking

spec:/score/timecounter/req/get-non-blocking

The directives to get time values shall use a sequence lock to synchronize the reader with exactly one writer.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/get*.

This requirement refines *spec:/score/timecounter/req/get-coarse*.

5.6.76 spec:/score/timecounter/req/group

spec:/score/timecounter/req/group

The super core shall have a component containing the clock implementation.

rationale: N/A

identifier: RTEMSScoreTimecounter

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/clock-monotonic*
- *spec:/score/timecounter/req/clock-realtime*

- *spec:/score/timecounter/req/get*
- *spec:/score/timecounter/req/get-coarse*
- *spec:/score/timecounter/req/install*
- *spec:/score/timecounter/req/ntp*
- *spec:/score/timecounter/req/tick-large-delta*
- *spec:/score/timecounter/req/tick-simple*

5.6.77 spec:/score/timecounter/req/install

spec:/score/timecounter/req/install

The timecounter component shall implement a directive to install a timecounter.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/install-frequency*
- *spec:/score/timecounter/req/install-quality*

5.6.78 spec:/score/timecounter/req/ntp

spec:/score/timecounter/req/ntp

The clock implementation shall optionally use a handler to do time adjustments and second updates through a NTP service.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/ntp-adjustment*
- *spec:/score/timecounter/req/ntp-seconds*
- *spec:/score/timecounter/req/ntp-step*
- *spec:/score/timecounter/req/ntp-step-large*

5.6.79 spec:/score/timecounter/req/tick-simple

spec:/score/timecounter/req/tick-simple

The clock implementation shall provide a simple timecounter tick service for timecounter devices with a period equal to the clock tick interval.

rationale: N/A

This requirement refines *spec:/score/timecounter/req/group*.

This requirement is refined by the following requirements:

- *spec:/score/timecounter/req/tick-simple-delta*
- *spec:/score/timecounter/req/tick-simple-offset*

5.6.80 spec:/score/tq/if/group

spec:/score/tq/if/group

The super core shall have a component containing the thread queue implementation.

rationale: N/A

identifier: RTEMSScoreThreadQueue

This requirement is refined by the following requirements:

- *spec:/score/tq/req/deadlock*
- *spec:/score/tq/req/enqueue-ceiling*
- *spec:/score/tq/req/enqueue-fifo*
- *spec:/score/tq/req/enqueue-mrsp*
- *spec:/score/tq/req/enqueue-priority*
- *spec:/score/tq/req/enqueue-priority-inherit*
- *spec:/score/tq/req/fatal-enqueue-sticky-from-bad-state*
- *spec:/score/tq/req/flush-fifo*
- *spec:/score/tq/req/flush-filter*
- *spec:/score/tq/req/flush-filter-stop*
- *spec:/score/tq/req/flush-priority*
- *spec:/score/tq/req/flush-priority-inherit*
- *spec:/score/tq/req/flush-remove-timer*
- *spec:/score/tq/req/flush-unblock*

- *spec:/score/tq/req/lock*
- *spec:/score/tq/req/priority-change*
- *spec:/score/tq/req/surrender*
- *spec:/score/tq/req/surrender-mrsp*
- *spec:/score/tq/req/surrender-priority-inherit*
- *spec:/score/tq/req/timeout*
- *spec:/score/tq/req/timeout-mrsp*
- *spec:/score/tq/req/timeout-priority-inherit*

5.6.81 spec:/score/tq/req/deadlock

spec:/score/tq/req/deadlock

The thread queue operations shall support deadlock detection.

rationale: N/A

This requirement refines *spec:/score/tq/if/group*.

This requirement is refined by the following requirements:

- *spec:/score/tq/req/deadlock-concurrent*
- *spec:/score/tq/req/enqueue-deadlock*

5.6.82 spec:/score/tq/req/lock

spec:/score/tq/req/lock

The state of each thread queue object shall be protected by an object-specific lock.

rationale: N/A

This requirement refines *spec:/req/fine-grained-locking*.

This requirement refines *spec:/score/tq/if/group*.

5.7 Security and privacy requirements

There are no security or privacy requirements

5.8 Portability requirements

There are no portability requirements

5.9 Software quality requirements

5.9.1 spec:/bsp/req/mem-clock

spec:/bsp/req/mem-clock

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with the clock driver enabled (CONFIGURE_APPLICATION_NEEDS_CLOCK_DRIVER).

rationale: N/A

This requirement refines *spec:/bsp/if/group-clock*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.2 spec:/req/mem-benchmark

spec:/req/mem-benchmark

The system shall provide benchmark programs to show the static memory usage of features.

rationale: N/A

This requirement refines *spec:/req/root*.

This requirement is refined by the following requirements:

- *spec:/rtems/req/mem-basic*

5.9.3 spec:/rtems/barrier/req/mem-wait-rel

spec:/rtems/barrier/req/mem-wait-rel

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_BARRIERS defined to one and calls to rtems_barrier_create, rtems_barrier_wait, and rtems_barrier_release.

rationale:

N/A

This requirement refines *spec:/rtems/barrier/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.4 spec:/rtems/barrier/req/mem-wait-rel-del

spec:/rtems/barrier/req/mem-wait-rel-del

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_BARRIERS defined to one and calls to rtems_barrier_create, rtems_barrier_wait, rtems_barrier_release, and rtems_barrier_delete.

rationale:

N/A

This requirement refines *spec:/rtems/barrier/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.5 spec:/rtems/clock/req/mem-get-uptime

spec:/rtems/clock/req/mem-get-uptime

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_clock_get_uptime.

rationale:

N/A

This requirement refines *spec:/rtems/clock/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.6 spec:/rtems/clock/req/mem-set

spec:/rtems/clock/req/mem-set

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_clock_set.

rationale: N/A

This requirement refines *spec:/rtems/clock/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.7 spec:/rtems/clock/req/mem-set-get-tod

spec:/rtems/clock/req/mem-set-get-tod

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_clock_set and rtems_clock_get_tod.

rationale: N/A

This requirement refines *spec:/rtems/clock/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.8 spec:/rtems/event/req/mem-snd-rcv

spec:/rtems/event/req/mem-snd-rcv

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_event_send and rtems_event_receive.

rationale: N/A

This requirement refines *spec:/rtems/event/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.9 spec:/rtems/fatal/req/mem-fatal

spec:/rtems/fatal/req/mem-fatal

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_fatal.

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.10 spec:/rtems/message/req/mem-bcst-rcv

spec:/rtems/message/req/mem-bcst-rcv

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_message_queue_construct, rtems_message_queue_broadcast, and rtems_message_queue_receive.

rationale: N/A

This requirement refines *spec:/rtems/message/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.11 spec:/rtems/message/req/mem-snd-rcv

spec:/rtems/message/req/mem-snd-rcv

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_message_queue_construct, rtems_message_queue_send, and rtems_message_queue_receive.

rationale: N/A

This requirement refines *spec:/rtems/message/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.12 spec:/rtems/message/req/mem-snd-rcv-del

spec:/rtems/message/req/mem-snd-rcv-del

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_message_queue_construct, rtems_message_queue_send, rtems_message_queue_receive, and rtems_message_queue_delete.

rationale: N/A

This requirement refines *spec:/rtems/message/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.13 spec:/rtems/message/req/mem-ugt-rcv

spec:/rtems/message/req/mem-ugt-rcv

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_message_queue_construct, rtems_message_queue_urgent, and rtems_message_queue_receive.

rationale: N/A

This requirement refines *spec:/rtems/message/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.14 spec:/rtems/part/req/mem-get-ret

spec:/rtems/part/req/mem-get-ret

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PARTITIONS defined to one and calls to rtems_partition_create, rtems_partition_get_buffer, and rtems_partition_return_buffer.

rationale: N/A

This requirement refines *spec:/rtems/part/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.15 spec:/rtems/part/req/mem-get-ret-del

spec:/rtems/part/req/mem-get-ret-del

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PARTITIONS defined to one and calls to rtems_partition_create, rtems_partition_get_buffer, rtems_partition_return_buffer, and rtems_partition_delete.

rationale: N/A

This requirement refines *spec:/rtems/part/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.16 spec:/rtems/ratemon/req/mem-period

spec:/rtems/ratemon/req/mem-period

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PERIODS defined to one and calls to rtems_rate_monotonic_create and rtems_rate_monotonic_period.

rationale: N/A

This requirement refines *spec:/rtems/ratemon/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.17 spec:/rtems/ratemon/req/mem-period-del

spec:/rtems/ratemon/req/mem-period-del

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PERIODS defined to one and calls to rtems_rate_monotonic_create, rtems_rate_monotonic_period, and rtems_rate_monotonic_delete.

rationale: N/A

This requirement refines *spec:/rtems/ratemon/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.18 spec:/rtems/req/mem-basic

spec:/rtems/req/mem-basic

The system shall provide a benchmark program to show the static memory usage of a basic application configuration.

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/req/mem-benchmark*.

This requirement is refined by the following requirements:

- *spec:/bsp/req/mem-clock*
- *spec:/rtems/barrier/req/mem-wait-rel*
- *spec:/rtems/barrier/req/mem-wait-rel-del*
- *spec:/rtems/clock/req/mem-get-uptime*
- *spec:/rtems/clock/req/mem-set*
- *spec:/rtems/clock/req/mem-set-get-tod*
- *spec:/rtems/event/req/mem-snd-rcv*
- *spec:/rtems/fatal/req/mem-fatal*
- *spec:/rtems/message/req/mem-bcst-rcv*
- *spec:/rtems/message/req/mem-snd-rcv*
- *spec:/rtems/message/req/mem-snd-rcv-del*
- *spec:/rtems/message/req/mem-ugt-rcv*
- *spec:/rtems/part/req/mem-get-ret*
- *spec:/rtems/part/req/mem-get-ret-del*
- *spec:/rtems/ratemon/req/mem-period*
- *spec:/rtems/ratemon/req/mem-period-del*
- *spec:/rtems/req/mem-smp-1*
- *spec:/rtems/sem/req/mem-obt-rel*
- *spec:/rtems/sem/req/mem-obt-rel-del*
- *spec:/rtems/signal/req/mem-catch-snd*
- *spec:/rtems/task/req/mem-delete*
- *spec:/rtems/task/req/mem-exit*
- *spec:/rtems/task/req/mem-get-affinity*

- *spec:/rtems/task/req/mem-get-priority*
- *spec:/rtems/task/req/mem-get-scheduler*
- *spec:/rtems/task/req/mem-mode*
- *spec:/rtems/task/req/mem-restart*
- *spec:/rtems/task/req/mem-set-affinity*
- *spec:/rtems/task/req/mem-set-priority*
- *spec:/rtems/task/req/mem-set-scheduler*
- *spec:/rtems/task/req/mem-sus-res*
- *spec:/rtems/task/req/mem-wake-after*
- *spec:/rtems/task/req/mem-wake-when*
- *spec:/rtems/timer/req/mem-after*
- *spec:/rtems/timer/req/mem-cancel*
- *spec:/rtems/timer/req/mem-delete*
- *spec:/rtems/timer/req/mem-reset*
- *spec:/rtems/timer/req/mem-srv-after*
- *spec:/rtems/timer/req/mem-srv-init*
- *spec:/rtems/timer/req/mem-srv-when*
- *spec:/rtems/timer/req/mem-when*
- *spec:/rtems/userext/req/mem-create*
- *spec:/rtems/userext/req/mem-delete*

5.9.19 spec:/rtems/req/mem-smp-1

spec:/rtems/req/mem-smp-1

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PROCESSORS defined to one using the SMP EDF scheduler (CONFIGURE_SCHEDULER_EDF_SMP).

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

This requirement is refined by the following requirements:

- *spec:/rtems/req/mem-smp-global-2*
- *spec:/rtems/req/mem-smp-global-4*

- *spec:/rtems/req/mem-smp-part-2*
- *spec:/rtems/req/mem-smp-part-4*
- *spec:/rtems/scheduler/req/mem-add-cpu*
- *spec:/rtems/scheduler/req/mem-rm-cpu*

5.9.20 *spec:/rtems/req/mem-smp-global-2*

spec:/rtems/req/mem-smp-global-2

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PROCESSORS defined to two using the global SMP EDF scheduler (CONFIGURE_SCHEDULER_EDF_SMP).

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-smp-1*.

5.9.21 *spec:/rtems/req/mem-smp-global-4*

spec:/rtems/req/mem-smp-global-4

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PROCESSORS defined to four using the global SMP EDF scheduler (CONFIGURE_SCHEDULER_EDF_SMP).

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-smp-1*.

5.9.22 *spec:/rtems/req/mem-smp-part-2*

spec:/rtems/req/mem-smp-part-2

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PROCESSORS defined to two using one SMP EDF scheduler for each configured processor (CONFIGURE_SCHEDULER_EDF_SMP).

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-smp-1*.

5.9.23 spec:/rtems/req/mem-smp-part-4

spec:/rtems/req/mem-smp-part-4

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_PROCESSORS defined to four using one SMP EDF scheduler for each configured processor (CONFIGURE_SCHEDULER_EDF_SMP).

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-smp-1*.

5.9.24 spec:/rtems/scheduler/req/mem-add-cpu

spec:/rtems/scheduler/req/mem-add-cpu

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_scheduler_add_processor.

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-smp-1*.

5.9.25 spec:/rtems/scheduler/req/mem-rm-cpu

spec:/rtems/scheduler/req/mem-rm-cpu

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_scheduler_remove_processor.

rationale: N/A

This requirement refines *spec:/rtems/req/group*.

This requirement refines *spec:/rtems/req/mem-smp-1*.

5.9.26 spec:/rtems/sem/req/mem-obt-rel

spec:/rtems/sem/req/mem-obt-rel

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_SEMAPHORES defined to one and calls to rtems_semaphore_create, rtems_semaphore_obtain, and rtems_semaphore_release.

rationale: N/A

This requirement refines *spec:/rtems/sem/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.27 spec:/rtems/sem/req/mem-obt-rel-del

spec:/rtems/sem/req/mem-obt-rel-del

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_SEMAPHORES defined to one and calls to rtems_semaphore_create, rtems_semaphore_obtain, rtems_semaphore_release, and rtems_semaphore_delete.

rationale: N/A

This requirement refines *spec:/rtems/sem/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.28 spec:/rtems/signal/req/mem-catch-snd

spec:/rtems/signal/req/mem-catch-snd

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_signal_catch and rtems_signal_send.

rationale: N/A

This requirement refines *spec:/rtems/signal/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.29 spec:/rtems/task/req/mem-delete

spec:/rtems/task/req/mem-delete

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_delete.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.30 spec:/rtems/task/req/mem-exit

spec:/rtems/task/req/mem-exit

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_exit.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.31 spec:/rtems/task/req/mem-get-affinity

spec:/rtems/task/req/mem-get-affinity

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_get_affinity.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.32 spec:/rtems/task/req/mem-get-priority

spec:/rtems/task/req/mem-get-priority

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_get_priority.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.33 spec:/rtems/task/req/mem-get-scheduler

spec:/rtems/task/req/mem-get-scheduler

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_get_scheduler.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.34 spec:/rtems/task/req/mem-mode

spec:/rtems/task/req/mem-mode

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_mode.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.35 spec:/rtems/task/req/mem-restart

spec:/rtems/task/req/mem-restart

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_restart.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.36 spec:/rtems/task/req/mem-set-affinity

spec:/rtems/task/req/mem-set-affinity

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_set_affinity.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.37 spec:/rtems/task/req/mem-set-priority

spec:/rtems/task/req/mem-set-priority

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_set_priority.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.38 spec:/rtems/task/req/mem-set-scheduler

spec:/rtems/task/req/mem-set-scheduler

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_set_scheduler.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.39 spec:/rtems/task/req/mem-sus-res

spec:/rtems/task/req/mem-sus-res

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_task_suspend and rtems_task_resume.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.40 spec:/rtems/task/req/mem-wake-after

spec:/rtems/task/req/mem-wake-after

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_wake_after.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.41 spec:/rtems/task/req/mem-wake-when

spec:/rtems/task/req/mem-wake-when

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_task_wake_when.

rationale: N/A

This requirement refines *spec:/rtems/task/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.42 spec:/rtems/timer/req/mem-after

spec:/rtems/timer/req/mem-after

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_fire_after.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.43 spec:/rtems/timer/req/mem-cancel

spec:/rtems/timer/req/mem-cancel

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_cancel.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.44 spec:/rtems/timer/req/mem-delete

spec:/rtems/timer/req/mem-delete

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_delete.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.45 spec:/rtems/timer/req/mem-reset

spec:/rtems/timer/req/mem-reset

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_reset.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.46 spec:/rtems/timer/req/mem-srv-after

spec:/rtems/timer/req/mem-srv-after

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_server_fire_after.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.47 spec:/rtems/timer/req/mem-srv-init

spec:/rtems/timer/req/mem-srv-init

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with a call to rtems_timer_initiate_server.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.48 spec:/rtems/timer/req/mem-srv-when

spec:/rtems/timer/req/mem-srv-when

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_server_fire_when.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.49 spec:/rtems/timer/req/mem-when

spec:/rtems/timer/req/mem-when

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with calls to rtems_timer_create and rtems_timer_fire_when.

rationale: N/A

This requirement refines *spec:/rtems/timer/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.50 spec:/rtems/userext/req/mem-create

spec:/rtems/userext/req/mem-create

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_USER_EXTENSIONS defined to one and a call to rtems_extension_create.

rationale: N/A

This requirement refines *spec:/rtems/userext/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.9.51 spec:/rtems/userext/req/mem-delete

spec:/rtems/userext/req/mem-delete

The system shall provide a benchmark program to show the static memory usage of a basic application configuration with CONFIGURE_MAXIMUM_USER_EXTENSIONS defined to one and calls to rtems_extension_create and rtems_extension_delete.

rationale: N/A

This requirement refines *spec:/rtems/userext/req/group*.

This requirement refines *spec:/rtems/req/mem-basic*.

5.10 Software reliability requirements

There are no reliability requirements

5.11 Maintainability requirements

There are no maintainability requirements

5.12 Safety requirements

There are no safety requirements

5.13 Software configuration and delivery requirements

There are no software configuration or delivery requirements

5.14 Data definition and database requirements

There are no data definition or database requirements

5.15 Human factors related requirements

There are no human factors related requirements

5.16 Adaptation and installation requirements

There are no adaptation and installation requirements