tpl_exec_static

context
stack
*entry
id
base_priority
max_activate_count
type

tpl_exec_common

*static_desc
*resources
*next_exec
*next_set
activate_count
priority
state

evt_set
evt_wait

tpl_task

*static_isr_desc

tpl_isr

```
void f(void)
{

…

        TerminateTask();
}
```

Task and ISR2 descriptor structures

tpl_exec_common contains the dynamic information shared by both tasks and Interrupt Service Routines 2 (which can call a system service):

static_desc	A pointer to the static descriptor of the task. This one is not dynamic.
priority	current priority of the task.
state	state of the task (SUSPENDED, READY, WAITING, RUNNING).
activate_count	multiple activation count.
resources	list head of the resources gotten by the task.
next_task	used when the task is in READY state. Pointer to the next task in the set (a set is an ordered group of tasks sharing the same priority).
next_set	used when the task in in READY state and for the first task of the set. Pointer to the first task in the next set (set are ordered from the greater priority to the lower priority.

tpl_task contains the dynamic information about task:

evt_set	events of the task.
evt_wait	events the task is waiting for.

tpl_isr contains the dynamic information about ISR2:

static_isr_desc	pointer to a helper function used to search for hardware that launched the interrupt and another pointer to link interruption that are activated with from the same source.

tpl_task_static contains the static information about tasks and ISR2:

id	id of the task. Used by the GetTaskId() service.
base_priority	initial priority of the task.
type	type of the task (BASIC or EXTENDED)
context	architecture dependant context storage. See AD_PPC.ppc for the PowerPC for instance.
stack	architecture dependant structure that define stack(s). See AD_PPC.pdf