

# Power Capping Framework

The power capping framework provides a consistent interface between the kernel and the user space that allows power capping drivers to expose the settings to user space in a uniform way.

## Terminology

The framework exposes power capping devices to user space via sysfs in the form of a tree of objects. The objects at the root level of the tree represent 'control types', which correspond to different methods of power capping. For example, the intel-rapl control type represents the Intel "Running Average Power Limit" (RAPL) technology, whereas the 'idle-injection' control type corresponds to the use of idle injection for controlling power.

Power zones represent different parts of the system, which can be controlled and monitored using the power capping method determined by the control type the given zone belongs to. They each contain attributes for monitoring power, as well as controls represented in the form of power constraints. If the parts of the system represented by different power zones are hierarchical (that is, one bigger part consists of multiple smaller parts that each have their own power controls), those power zones may also be organized in a hierarchy with one parent power zone containing multiple subzones and so on to reflect the power control topology of the system. In that case, it is possible to apply power capping to a set of devices together using the parent power zone and if more fine grained control is required, it can be applied through the subzones.

Example sysfs interface tree:

```

/devices/physical/powercap
a""a"ea"€intel-rapl
a"ea"€a"€intel-rapl:0
a",A A a"ea"€a"€constraint_0_name
a",A A a"ea"€a"€constraint_0_power_limit_uw
a",A A a"ea"€a"€constraint_0_time_window_us
a",A A a"ea"€a"€constraint_1_name
a",A A a"ea"€a"€constraint_1_power_limit_uw
a",A A a"ea"€a"€constraint_1_time_window_us
a",A A a"ea"€a"€device -> ../../intel-rapl
a",A A a"ea"€a"€energy_uj
a",A A a"ea"€a"€intel-rapl:0:0
a",A A a",A A a"ea"€a"€constraint_0_name
a",A A a",A A a"ea"€a"€constraint_0_power_limit_uw
a",A A a",A A a"ea"€a"€constraint_0_time_window_us
a",A A a",A A a"ea"€a"€constraint_1_name
a",A A a",A A a"ea"€a"€constraint_1_power_limit_uw
a",A A a",A A a"ea"€a"€constraint_1_time_window_us
a",A A a",A A a"ea"€a"€device -> ../../intel-rapl:0
a",A A a",A A a"ea"€a"€energy_uj
a",A A a",A A a"ea"€a"€max_energy_range_uj
a",A A a",A A a"ea"€a"€name
a",A A a",A A a"ea"€a"€enabled
a",A A a",A A a"ea"€a"€power
a",A A a",A A a",A A a"ea"€a"€async
a",A A a",A A a",A A []
a",A A a",A A a"ea"€a"€subsystem -> ../../../../class/power_cap
a",A A a",A A a""a"€a"€event
a",A A a"ea"€a"€intel-rapl:0:1
a",A A a",A A a"ea"€a"€constraint_0_name
a",A A a",A A a"ea"€a"€constraint_0_power_limit_uw
a",A A a",A A a"ea"€a"€constraint_0_time_window_us
a",A A a",A A a"ea"€a"€constraint_1_name
a",A A a",A A a"ea"€a"€constraint_1_power_limit_uw
a",A A a",A A a"ea"€a"€constraint_1_time_window_us
a",A A a",A A a"ea"€a"€device -> ../../intel-rapl:0
a",A A a",A A a"ea"€a"€energy_uj
a",A A a",A A a"ea"€a"€max_energy_range_uj
a",A A a",A A a"ea"€a"€name
a",A A a",A A a"ea"€a"€enabled
a",A A a",A A a"ea"€a"€power
a",A A a",A A a",A A a"ea"€a"€async
a",A A a",A A a",A A []
a",A A a",A A a"ea"€a"€subsystem -> ../../../../class/power_cap
a",A A a",A A a""a"€a"€event
a",A A a"ea"€a"€max_energy_range_uj
a",A A a"ea"€a"€max_power_range_uw
a",A A a"ea"€a"€name
a",A A a"ea"€a"€enabled
a",A A a"ea"€a"€power
a",A A a",A A a"ea"€a"€async
a",A A a",A A a",A A []
a",A A a"ea"€a"€subsystem -> ../../../../class/power cap

```



constraint\_j\_\* attributes correspond to the jth constraint (j = 0,1,2).

For example:

```
constraint_0_name
constraint_0_power_limit_uw
constraint_0_time_window_us
constraint_1_name
constraint_1_power_limit_uw
constraint_1_time_window_us
constraint_2_name
constraint_2_power_limit_uw
constraint_2_time_window_us
```

## Power Zone Attributes

### Monitoring attributes

energy\_uj (rw)

Current energy counter in micro joules. Write "0" to reset. If the counter can not be reset, then this attribute is read only.

max\_energy\_range\_uj (ro)

Range of the above energy counter in micro-joules.

power\_uw (ro)

Current power in micro watts.

max\_power\_range\_uw (ro)

Range of the above power value in micro-watts.

name (ro)

Name of this power zone.

It is possible that some domains have both power ranges and energy counter ranges; however, only one is mandatory.

### Constraints

constraint\_X\_power\_limit\_uw (rw)

Power limit in micro watts, which should be applicable for the time window specified by "constraint\_X\_time\_window\_us".

constraint\_X\_time\_window\_us (rw)

Time window in micro seconds.

constraint\_X\_name (ro)

An optional name of the constraint

constraint\_X\_max\_power\_uw(ro)

Maximum allowed power in micro watts.

constraint\_X\_min\_power\_uw(ro)

Minimum allowed power in micro watts.

constraint\_X\_max\_time\_window\_us(ro)

Maximum allowed time window in micro seconds.

constraint\_X\_min\_time\_window\_us(ro)

Minimum allowed time window in micro seconds.

Except power\_limit\_uw and time\_window\_us other fields are optional.

### Common zone and control type attributes

enabled (rw): Enable/Disable controls at zone level or for all zones using a control type.

## Power Cap Client Driver Interface

The API summary:

Call powercap\_register\_control\_type() to register control type object. Call powercap\_register\_zone() to register a power zone (under a given control type), either as a top-level power zone or as a subzone of another power zone registered earlier. The number of constraints in a power zone and the corresponding callbacks have to be defined prior to calling powercap\_register\_zone() to register that zone.

To Free a power zone call powercap\_unregister\_zone(). To free a control type object call powercap\_unregister\_control\_type(). Detailed API can be generated using kernel-doc on include/linux/powercap.h.