# APM

November 4, 2002

## 1 Module `Apm` : APM interface

This library provides an interface with the APM driver.

### 1.1 APM status

```
type  apm_support =
  | Supported
```

APM support of the right version exists in the kernel.

```
  | Unsupported
```

No APM support in the kernel.

```
  | WrongVersion
```

APM support of a wrong version exists in the kernel.

```
val exists : unit -> apm_support
```

`exists ()` returns a flag indicating the status of the support of APM in the kernel.

### 1.2 Getting APM info

```
type  ac_line_status =
  | OffLine
```

AC off-line

```
  | OnLine
```

AC on-line

```
  | OnBackupPower
```

On backup power

The AC line status is reported using a value of type `ac_line_status`.

```
type  battery_status =
  | High
```

Battery status high

`| Low`

Battery status low

`| Critical`

Battery status critical

`| Charging`

Battery charging

`| NoBattery`

No battery.

The battery status is reported using a value of type `battery_status`.

`type  info = {`

`driver_version : string ;`

The kernel driver version.

`version_major : int ;`

The major version number of the APM BIOS.

`version_minor : int ;`

The minor version number of the APM BIOS.

`support_32_bits : bool ;`

Indicates wether the 32-bit APM interface is supported.

`power_management_disabled : bool ;`

Indicates wether the APM BIOS Power Management is currently disabled.

`power_management_disengaged : bool ;`

Indicates wether the APM BIOS Power Management is currently disengaged.

`ac_line_status : ac_line_status ;`

Gets the current AC line status.

`battery_status : battery_status ;`

Gets the current battery status.

`battery_percentage : int option ;`

The percentage of charging of the battery, if available.

`battery_time : int option ;`

The time in seconds, if available.

`}`

Information about APM are reported using a record of type `info`. It provides generic information about the driver and current information about the AC line and battery status.

`val read : unit -> info`

`read ()` returns a record of type `info` giving the current information from the APM driver. If APM is not supported by the kernel, an exception `Apm.Failure`[1.4] is raised.

## 1.3 Sending messages

`val suspend : unit -> unit`

`suspend ()` tries to set the power state to suspend. If it fails, an exception `Apm.Failure`[1.4] is raised.

`val standby : unit -> unit`

`standby ()` tries to set the power state to standby. If it fails, an exception `Apm.Failure`[1.4] is raised.

## 1.4 Errors report

`exception Failure of string * string`

Functions of this library raise an exception `Failure` with an appropriate message when they encouter an error.

`val handle_apm_error : ('a -> 'b) -> 'a -> 'b`

`handle_apm_error f x` applies `f` to `x` and returns the result. If the exception `Failure` is raised, it prints a message describing the error and exits with code 2.