

---

# hwd Documentation

*Release 0.3*

**Outernet Inc <apps@outernet.is>**

April 04, 2016



<b>1</b>	<b>Installing</b>	<b>3</b>
<b>2</b>	<b>Basic usage</b>	<b>5</b>
<b>3</b>	<b>API documentation</b>	<b>7</b>
3.1	Base wrapper class . . . . .	7
3.2	Network device wrappers . . . . .	8
3.3	Storage device wrappers . . . . .	9
3.4	udev helpers . . . . .	12
	<b>Python Module Index</b>	<b>13</b>



hwd wraps around [pyudev](#)'s `Device` objects to provide functionality specific to different types of hardware. Currently, hwd provides a base wrapper class that provides common information such as bus, vendor, and model, and classes for wrapping storage and network devices.

hwd is being developed at [Outernet](#) specifically for integration of hardware management functionality into [Librarian](#) that runs on Outernet receives. As such, there are no plans for adding support for devices that are common on personal computers.



---

# Installing

---

hwd is in early development, and is therefore not yet published on a PyPI repository. To install the latest development version from GitHub, use this command:

```
pip install https://github.com/Outernet-Project/hwd/archive/develop.zip
```





---

### Basic usage

---

To use the wrapper classes, you need `pyudev.Device` objects. You can obtain them using `pyudev` API, or using the `devices_by_subsystem()` helper function.

Once you have one or more `Device` objects, you can instantiate the wrapper classes, passing the `Device` object to the constructor.



---

## API documentation

---

### 3.1 Base wrapper class

#### 3.1.1 Basic usage

*Wrapper* class provides access to common basic information about all device types.

Example:

```
>>> devs = list(udev.devices_by_subsystem('net'))
>>> device = wrapper.Wrapper(devs[0])
>>> device.bus
u'pci'
>>> device.model
u'Wireless 7260 (Dual Band Wireless-AC 7260)'
>>> device.name
u'wlp2s0'
>>> device.vendor
u'Intel Corporate'
>>> device.system_path
u'/sys/devices/pci0000:00/0000:00:1c.3/0000:02:00.0/net/wlp2s0'
>>> device.devid
(u'0x8086', u'0x08b1')
```

#### 3.1.2 Module contents

**class** `hwd.wrapper.Wrapper(dev)`

Generic wrapper class that wraps `pyudev.Device` instances.

`dev` is a `pyudev.Device` instance. Device's `sys_name` property is stored as `name` property on the wrapper instance.

**aliases**

Return a list of device aliases (symlinks) that match this device. This list also includes the device node so it can be treated as a list of all filesystem objects that point to this device.

**bus**

Device's bus. If device is not on any bus, this property evaluates to `None`.

**device**

The underlying `pyudev.Device` instance can be accessed by using the `device` property. This property

is cached, so only one lookup is performed to obtain the device object. This cache is invalidated by `refresh()` method.

**dev\_id**

Two-tuple containing device's vendor ID and model ID (hex).

**model**

First non-empty value from the following list:

- model name from model database
- model name as reported by the device driver
- model ID (hex)

If none of the above attributes are available, evaluates to `None`.

**node**

Device node. Not all devices have a node. In case a device has no node, this property evaluates to `None`.

**refresh()**

Clears the `device` cache.

---

**Note:** This method does not cause immediate lookup of the udev context. Lookup is done when the `device` property is accessed.

---

**system\_path**

System path of the device.

**vendor**

First non-empty value from the following list:

- organization name (OUI) from device database
- vendor from vendor database
- vendor as reported by the device driver
- vendor ID (hex)

If none of the above attributes are available, evaluates to `None`.

## 3.2 Network device wrappers

### 3.2.1 Basic usage

### 3.2.2 Module contents

**class** `hwd.network.NetIface(dev)`

Wrapper for `pyudev.Device` objects of 'net' subclass.

**ipv4addr**

IPv4 address.

**ipv4gateway**

IPv4 default gateway.

**ipv4netmask**

IPv4 netmask.

**ipv6addr**

IPv6 address.

**ipv6gateway**

IPv6 default gateway.

**ipv6netmask**

IPv6 netmask.

**is\_connected**

Whether there is carrier.

**mac**

NIC's MAC address.

**type**

NIC type. Not all network devices have this value. For wireless devices this value is 'wlan', and for loopback devices, the value is 'loop'. When the value is missing, 'eth' is returned as most ethernet devices authors have encountered have this value missing.

## 3.3 Storage device wrappers

### 3.3.1 Quick example

Basic usage of storage wrappers uses the disk device type wrapped in *Disk* wrapper, and accessing disk and partition information using properties on that object. All partitions are accessible through the object's *partitions* attribute, so it is not necessary to separately wrap partition objects (though there is nothing wrong with that).

Example:

```
>>> devs = list(udev.devices_by_subsystem('block'))
>>> device = storage.Disk(devs[0])
>>> device.bus
u'ata'
>>> device.size
256060514304
>>> device.sectors
500118192
>>> device.part_table_type
u'gpt'
>>> p1 = device.partitions[0]
>>> p1.offset
4097
>>> p1.size
314573312
>>> p1.format
u'vfat'
```

### 3.3.2 Module contents

**class** hwd.storage.**Disk**(dev)

Wrapper for pyudev.Device objects of 'disk' type.

**is\_read\_only**

Whether disk is read-only. This evaluates to True if disk is read-only.

**is\_removable**

Whether disk is removable. This property evaluates to `True` if disk is removable. Note that this does not mean disk is USB-attached, and does not necessarily match the common notion of removable devices.

If you wish to know whether a device is USB-attached, check whether the value of the `bus` property is `'usb'`.

**part\_table\_type**

Partition table type. Evaluates to either `'dos'` or `'gpt'`.

**partitions**

Iterable containing disk's partition objects. Objects in the iterable are `Partition` instances.

**sectors**

Disk size in sectors. If for some reason, this information is not available, this property evaluates to `-1`.

**size**

Disk capacity in bytes. This value is obtained by multiplying sector size by 512.

**uuid**

Partition table UUID. Note that UUIDs for different partition table types have different fomats.

**class** `hwd.storage.Fstat` (*total, used, free, pct\_used, pct\_free*)

**free**

Alias for field number 2

**pct\_free**

Alias for field number 4

**pct\_used**

Alias for field number 3

**total**

Alias for field number 0

**used**

Alias for field number 1

**class** `hwd.storage.Mountable`

Mixing providing interfaces for mountable storage devices.

**mount\_points**

Iterator of partition's mount points obtained by reading `/proc/mounts`. Returns empty list if `/proc/mounts` is not readable or if there are no mount points.

**size**

Subclasses using this mixin must implement their own `size` property which returns the total capacity in bytes.

**stat**

Return disk usage information for the partition in `Fstat` format. If disk usage information is not available, then `None` is returned. Disk usage information is only available for regular filesystems that are mounted.

**class** `hwd.storage.MtabEntry` (*dev, mdir, fstype, opts, cfreq, cpass*)

**cfreq**

Alias for field number 4

**cpass**

Alias for field number 5

**dev**  
Alias for field number 0

**fstype**  
Alias for field number 2

**mdir**  
Alias for field number 1

**opts**  
Alias for field number 3

**class** `hwd.storage.Partition` (*dev*, *disk=None*)  
Wrapper for `pyudev.Device` objects of 'partition' type.

As with all wrappers, this class takes `dev` as its first argument. The optional `disk` argument can be passed, and is stored as the `disk` property. This is mostly used by [Disk](#) class to maintain a reference to itself.

**format**  
Fiesystem type. This evaluates to any number of supported file system types such as 'ext4' or 'vfat'.

---

**Note:** Extended partitions will have this property evaluate to `None`.

---

**is\_extended**  
Whether partition is extended.

**label**  
Volume label. This property evaluates to `None` if no volume label is not set on a partition.

**number**  
Partition number. This specifies a position of the partition in the partition table. If the value is not known for some reason, this property evaluates to `-1`.

**offset**  
Partition offset in sectors. If this information is not available for some reason, it evaluates to `-1`.

**parent\_class**  
alias of [Disk](#)

**part\_type**  
Partition type ID. This is expressed in hex string. Note that this is not the same as filesystem type which is available through the [part\\_type](#) property.

**scheme**  
Partition entry scheme. This evaluates to either 'dos' or 'gpt'.

**sectors**  
Partition size in sectors. If this information is not available for some reason, it evaluates to `-1`.

**size**  
Partition size in bytes. This value is obtained by multiplying the sector size by 512.

**usage**  
Filesystem usage (purpose). In most cases this should evaluate to 'filesystem'. In some cases (e.g., swap partition), it may evaluate to 'other' or some other value.

**uuid**  
Filesystem UUID. Note that this is not the same as the partition UUID (which is not available through this wrapper, other than directly accessing the underlying `pyudev.Device` object).

**class** `hwd.storage.PartitionBase` (*dev*, *disk=None*)

Base class for all partition devices. This class encapsulates the base functionality for all mountable partitions. It takes *disk* as an optional argument, and allows access to parent device from the partition devices.

**class** `hwd.storage.UbiContainer` (*dev*)

Wrapper for `pyudev.Device` objects of the ‘ubi’ subsystem.

The ‘ubi’ subsystem has two types of devices. The parent devices serve as containers for the actual volumes. This class is meant to be used by containers, rather than volumes.

This class mainly exists to facilitate the *PartitionBase* API, specifically the API that allows access to parent device.

**class** `hwd.storage.UbiVolume` (*dev*, *disk=None*)

Wrapper for `pyudev.Device` objects of the ‘ubi’ subsystem.

The ‘ubi’ subsystem has two types of devices. The parent devices serve as containers for the actual volumes. This class is meant to be used by volumes, rather than their parent devices.

**aliases**

Aliases for UBI volume. This property evaluates to device node itself plus the ‘ubi\${INDEX}:\${LABEL}’ string. The latter is used to identify the device in `/proc/mounts` table, and is not really an alias.

**label**

UBI volume name. For compatibility with *Partition* API, we name this property ‘label’.

**parent\_class**

alias of *UbiContainer*

**sectors**

Simulated number of sectors derived from volume size. This property is provided for compatibility with *Partition* API.

**Warning:** The concept does not really apply to UBI volumes, so it’s best to not rely on the value of this property.

**size**

Volume capacity in bytes. This property evaluates to -1 if size information is not available for any reason.

`hwd.storage.mounts()`

Iterator yielding mount points that appear in `/proc/mounts`. If `/proc/mounts` is not readable or does not exist, this function raises an exception.

## 3.4 udev helpers

`hwd.udev.devices_by_subsystem` (*subsys*, *only=<function <lambda>>*)

Iterator that yields devices that belong to specified subsystem. Returned values are `pyudev.Device` instances.

The *only* argument can be used to further filter devices. It should be a function that takes a `pyudev.Device` object, and returns `True` or `False` to indicate whether device should be returned.

Example:

```
>>> devices_by_subsystem('net')
[Device('/sys/devices/pci0000:00/0000:00:1c.3/0000:02:00.0/net/wlp2s0'),
 Device('/sys/devices/virtual/net/lo')]
```



## h

`hwd.network`, 8  
`hwd.storage`, 9  
`hwd.udev`, [12](#)  
`hwd.wrapper`, 7



## A

aliases (hwd.storage.UbiVolume attribute), 12  
aliases (hwd.wrapper.Wrapper attribute), 7

## B

bus (hwd.wrapper.Wrapper attribute), 7

## C

cfreq (hwd.storage.MtabEntry attribute), 10  
cpass (hwd.storage.MtabEntry attribute), 10

## D

dev (hwd.storage.MtabEntry attribute), 10  
device (hwd.wrapper.Wrapper attribute), 7  
devices\_by\_subsystem() (in module hwd.udev), 12  
devid (hwd.wrapper.Wrapper attribute), 8  
Disk (class in hwd.storage), 9

## F

format (hwd.storage.Partition attribute), 11  
free (hwd.storage.Fstat attribute), 10  
Fstat (class in hwd.storage), 10  
fstype (hwd.storage.MtabEntry attribute), 11

## H

hwd.network (module), 8  
hwd.storage (module), 9  
hwd.udev (module), 12  
hwd.wrapper (module), 7

## I

ipv4addr (hwd.network.NetIface attribute), 8  
ipv4gateway (hwd.network.NetIface attribute), 8  
ipv4netmask (hwd.network.NetIface attribute), 8  
ipv6addr (hwd.network.NetIface attribute), 8  
ipv6gateway (hwd.network.NetIface attribute), 9  
ipv6netmask (hwd.network.NetIface attribute), 9  
is\_connected (hwd.network.NetIface attribute), 9  
is\_extended (hwd.storage.Partition attribute), 11  
is\_read\_only (hwd.storage.Disk attribute), 9

is\_removable (hwd.storage.Disk attribute), 9

## L

label (hwd.storage.Partition attribute), 11  
label (hwd.storage.UbiVolume attribute), 12

## M

mac (hwd.network.NetIface attribute), 9  
mdir (hwd.storage.MtabEntry attribute), 11  
model (hwd.wrapper.Wrapper attribute), 8  
mount\_points (hwd.storage.Mountable attribute), 10  
Mountable (class in hwd.storage), 10  
mounts() (in module hwd.storage), 12  
MtabEntry (class in hwd.storage), 10

## N

NetIface (class in hwd.network), 8  
node (hwd.wrapper.Wrapper attribute), 8  
number (hwd.storage.Partition attribute), 11

## O

offset (hwd.storage.Partition attribute), 11  
opts (hwd.storage.MtabEntry attribute), 11

## P

parent\_class (hwd.storage.Partition attribute), 11  
parent\_class (hwd.storage.UbiVolume attribute), 12  
part\_table\_type (hwd.storage.Disk attribute), 10  
part\_type (hwd.storage.Partition attribute), 11  
Partition (class in hwd.storage), 11  
PartitionBase (class in hwd.storage), 11  
partitions (hwd.storage.Disk attribute), 10  
pct\_free (hwd.storage.Fstat attribute), 10  
pct\_used (hwd.storage.Fstat attribute), 10

## R

refresh() (hwd.wrapper.Wrapper method), 8

## S

scheme (hwd.storage.Partition attribute), 11

sectors (hwd.storage.Disk attribute), [10](#)  
sectors (hwd.storage.Partition attribute), [11](#)  
sectors (hwd.storage.UbiVolume attribute), [12](#)  
size (hwd.storage.Disk attribute), [10](#)  
size (hwd.storage.Mountable attribute), [10](#)  
size (hwd.storage.Partition attribute), [11](#)  
size (hwd.storage.UbiVolume attribute), [12](#)  
stat (hwd.storage.Mountable attribute), [10](#)  
system\_path (hwd.wrapper.Wrapper attribute), [8](#)

## T

total (hwd.storage.Fstat attribute), [10](#)  
type (hwd.network.NetIface attribute), [9](#)

## U

UbiContainer (class in hwd.storage), [12](#)  
UbiVolume (class in hwd.storage), [12](#)  
usage (hwd.storage.Partition attribute), [11](#)  
used (hwd.storage.Fstat attribute), [10](#)  
uuid (hwd.storage.Disk attribute), [10](#)  
uuid (hwd.storage.Partition attribute), [11](#)

## V

vendor (hwd.wrapper.Wrapper attribute), [8](#)

## W

Wrapper (class in hwd.wrapper), [7](#)