

## VOSTreeInfo Class Reference

### Public Member Functions

- def [\\_\\_init\\_\\_](#) (self, [dlfile](#))
- def [get\\_adk\\_version](#) (self)  
Retrieve ADK version. [More...](#)
- def [get\\_os\\_version](#) (self)  
Retrieve OS version. [More...](#)
- def [has\\_os](#) (self)  
Check if contains an OS. [More...](#)
- def [has\\_blacklists](#) (self)  
Check if contains blacklists. [More...](#)
- def [get\\_blacklists](#) (self)  
Returns dictionary such as dict[device] = set(blacklisted bundles,...) [More...](#)
- def [get\\_full\\_device\\_list](#) (self)  
Return sorted list of devices. [More...](#)
- def [get\\_bundle\\_device\\_list](#) (self, bundle)  
Returns sorted list of all products on which this bundle can be installed. [More...](#)

### Data Fields

[dlfile](#)

[blacklists](#)

[full\\_device\\_list](#)

### Constructor & Destructor Documentation

[?\\_\\_init\\_\\_\(\)](#)

```
def __init__( self,
              dlfile
            )
```

## Member Function Documentation

### [? get\\_adk\\_version\(\)](#)

```
def get_adk_version ( self )
```

Retrieve ADK version.

Returns

adk version string or empty string if not found

### [? get\\_blacklists\(\)](#)

```
def get_blacklists ( self )
```

Returns dictionary such as dict[device] = set(blacklisted bundles,...)

Returns

return blacklists dictionary

### [? get\\_bundle\\_device\\_list\(\)](#)

```
def get_bundle_device_list ( self,
                            bundle
                          )
```

Returns sorted list of all products on which this bundle can be installed.

This takes into account blacklisting and targeting :param bundle: bundle node

Returns

sorted list of supported devices for this bundle

### [? get\\_full\\_device\\_list\(\)](#)

```
def get_full_device_list ( self )
```

Return sorted list of devices.

Dlfile must contain an OS. Looks for devices in blacklists, then by llayer bundle names

Returns  
sorted list of devices

### **? get\_os\_version()**

```
def get_os_version ( self )
```

Retrieve OS version.

Returns  
adk version string or empty string if not found

### **? has\_blacklists()**

```
def has_blacklists ( self )
```

Check if contains blacklists.

Returns  
True if contains blacklists, False if not

### **? has\_os()**

```
def has_os ( self )
```

Check if contains an OS.

Returns  
True if contains OS, False if not

## **Field Documentation**

### **? blacklists**

blacklists

## **? dfile**

dfile

## **? full\_device\_list**

full\_device\_list

---

The documentation for this class was generated from the following file:

- packman/packmanlib/[treeinfo.py](#)