

Udisks

udisks (<http://www.freedesktop.org/wiki/Software/udisks/>)

provides a daemon *udisksd*, that implements D-Bus interfaces used to query and manipulate storage devices, and a command-line tool *udisksctl*, used to query and use the daemon.

Contents

- 1 Installation
- 2 Configuration
- 3 Tips and tricks
 - 3.1 Mount helpers
 - 3.1.1 udevadm monitor
 - 3.2 Mount to /media (udisks2)
 - 3.3 Mount loop devices
 - 3.4 Hide selected partitions
 - 3.4.1 Example
 - 3.5 Apply ATA settings (udisks2)
- 4 Troubleshooting

Related articles

udev

Mount

Polkit

**File manager
functionality**

- 4.1 Hidden devices (udisks2)
- 4.2 Devices do not remain unmounted (udisks)
- 4.3 Broken standby timer (udisks2)
- 5 See also

Installation

There are two versions of *udisks* called **udisks** (<https://aur.archlinux.org/packages/udisks/>)^{AUR} and **udisks2** (<https://www.archlinux.org/packages/?name=udisks2>).

Development of **udisks** has ceased in favor of **udisks2**. [1] (<http://davidz25.blogspot.be/2012/03/simpler-faster-better.html>)

udisksd(8) (<https://jlk.fjfi.cvut.cz/arch/manpages/man/udisksd.8>) (for **udisks2**) and **udisks-daemon(8)** (<https://udisks.freedesktop.org/docs/1.0.5/udisks-daemon.8.html>)^[dead link 2017-08-08] (for **udisks**) are started on-demand by **D-Bus**, and should not be enabled explicitly. They can be controlled through the command-line with **udisksctl(1)** (<https://jlk.fjfi.cvut.cz/arch/manpages/man/udisksctl.1>) and **udisks(1)** (<https://udisks.freedesktop.org/docs/1.0.5/udisks.1.html>)^[dead link 2017-08-08], respectively.

Configuration

Actions a user can perform using udisks are restricted with **Polkit**. If the **user session** is not activated or present (for example, when controlling udisks from a **systemd/User** service), adjust Polkit rules accordingly.

See [2] (<https://github.com/coldfix/udiskie/wiki/Permissions>) for common udisks permissions for the `storage` group, and [3] (<https://gist.github.com/gravity/3886114#file-udisks2-allow-mount-internal-js>) for a more restrictive example.

Tips and tricks

Mount helpers

The automatic mounting of devices is easily achieved with udisks wrappers. See also **List of applications#Mount tools**.

- **bashmount** — A bash script to mount and manage removable media as a regular user with *udisks2*.

<https://github.com/jamielinux/bashmount> || **bashmount** (<https://aur.archlinux.org/packages/bashmount/>)^{AUR}

- **udiskie** — *udisks2* automounter with optional notifications, tray icon and support for password protected **LUKS devices**. See the udiskie wiki for **details** (<https://github.com/>

[coldfix/udiskie/wiki/Usage](https://coldfix.github.io/udiskie/wiki/Usage))

<https://pypi.python.org/pypi/udiskie> || **udiskie** (<https://www.archlinux.org/packages/?name=udiskie>)

- **udisksvm** — GUI *udisks2* wrapper written in Python3 and using the Qt5 framework. It uses mouse clicks to mount, unmount removable devices or eject a CD/DVD. See the **README** (<https://github.com/berbae/udisksvm/blob/master/README>) file for details.

<https://github.com/berbae/udisksvm> || **udisksvm** (<https://aur.archlinux.org/packages/udisksvm/>)^{AUR}

- **udevil** — Includes **devmon** (<http://igurublog.wordpress.com/downloads/script-devmon>), which is compatible to *udisks* and *udisks2*.

<https://github.com/IgnorantGuru/udevil> || **udevil** (<https://www.archlinux.org/packages/?name=udevil>)

Note: *devmon* only uses *udisks* or *udisks2* for mounting (in this order) if *udevil* or *pmount* miss the SUID permission. To remove this permission, run `chmod -s /usr/bin/udevil` as root.

udevadm monitor

You may use `udevadm monitor` to monitor block events and mount drives when a new block device is created. Stale mount points are automatically removed by *udisksd*, such that no special action is required on deletion.

```
#!/bin/sh

pathtoname() {
    udevadm info -p /sys/"$1" | awk -v FS== '/DEVNAME/ {print $2}'
}

stdbuf -oL -- udevadm monitor --udev -s block | while read -r -- _ _ event devpath _; do
    if [ "$event" = add ]; then
        devname=$(pathtoname "$devpath")
        udisksctl mount --block-device "$devname" --no-user-interaction
    fi
done
```

Mount to /media (udisks2)

By default, udisks2 mounts removable drives under the ACL controlled directory `/run/media/$USER/`. If you wish to mount to `/media` instead, use this rule:

```
/etc/udev/rules.d/99-udisks2.rules

-----
# UDISKS_FILESYSTEM_SHARED
# ==1: mount filesystem to a shared directory (/media/VolumeName)
# ==0: mount filesystem to a private directory (/run/media/$USER/VolumeName)
# See udisks(8)
ENV{ID_FS_USAGE}=="filesystem|other|crypto", ENV{UDISKS_FILESYSTEM_SHARED}=="1"
```

Since `/media`, unlike `/run`, is not mounted by default as a **tmpfs**, you may also wish to create a **tmpfiles.d** snippet to clean stale mountpoints at every boot:

```
/etc/tmpfiles.d/media.conf
```

```
D /media 0755 root root 0 -
```

Mount loop devices

To easily mount ISO images, use the following command:

```
$ udisksctl loop-setup -r -f image.iso
```

This will create a loop device and show the ISO image ready to mount. Once unmounted, the loop device will be terminated by **udev**.

Note: This mounts a read only image. To mount raw disk images, such as for **QEMU**, remove the `-r` flag, and release the image after use with `udisksctl loop-delete -b /dev/loop0`. Substitute `/dev/loop0` with the name of the loop device.

Hide selected partitions

If you wish to prevent certain partitions or drives appearing on the desktop, you can create a udev rule, for example `/etc/udev/rules.d/10-local.rules` :

```
KERNEL=="sda1", ENV{UDISKS_PRESENTATION_HIDE}="1"  
KERNEL=="sda2", ENV{UDISKS_PRESENTATION_HIDE}="1"
```

shows all partitions with the exception of `sda1` and `sda2` on your desktop. Note that if you are using **udisks2** (<https://www.archlinux.org/packages/?name=udisks2>), the above will not work as `UDISKS_PRESENTATION_HIDE` is no longer supported. Instead, use `UDISKS_IGNORE` as follows:

```
KERNEL=="sda1", ENV{UDISKS_IGNORE}="1"  
KERNEL=="sda2", ENV{UDISKS_IGNORE}="1"
```

Because block device names can change between reboots, it is also possible to use UUIDs (as gathered from executing the `blkid /dev/sdX` command) to hide partitions or whole devices:

Example

```
# blkid /dev/sdX  
/dev/sdX: LABEL="Filesystem Label" UUID="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX" UUID_SUB="YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY" TYPE="btrfs"
```

Then the following line can be used:

```
ENV{ID_FS_UUID}=="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX", ENV{UDISKS_IGNORE}="1"
```

The above line is also useful to hide multi device btrfs filesystems, as all the devices from a single btrfs filesystem will share the same UUID across the devices but will have different SUB_UUID for each individual device.

Apply ATA settings (udisks2)

At start-up and when a drive is connected, `udisksd` will apply configuration stored in the file `/etc/udisks2/IDENTIFIER.conf` where *IDENTIFIER* is the value of the `Drive:Id` property for the drive. Currently ATA settings are supported. See [udisks\(8\)](https://wiki.archlinux.org/manpages/man/udisks.8) (<https://wiki.archlinux.org/manpages/man/udisks.8>) for available options. These settings have essentially the same effect as those of `hdparm`, but they are persistent as long as the `udisks` daemon is autostarted.

For example, to set standby timeout to 240 (20 minutes) for a drive, add the following:

```
/etc/udisks2/DriveId.conf
```

```
[ATA]
StandbyTimeout=240
```

To obtain the `DriveId` for your drive, use

```
$ udevadm info --query=all --name=/dev/sdx | grep ID_SERIAL
```

Alternatively, use a GUI utility to manage the configuration file, such as [gnome-disk-utility](https://www.archlinux.org/packages/?name=gnome-disk-utility) (<https://www.archlinux.org/packages/?name=gnome-disk-utility>).

Troubleshooting

Hidden devices (udisks2)

Udisks2 hides certain devices from the user by default. If this is undesired or otherwise problematic, copy `/usr/lib/udev/rules.d/80-udisks2.rules` to `/etc/udev/rules.d/80-udisks2.rules` and remove the following section in the copy:

```
# -----  
# -----  
# -----  
# Devices which should not be display in the user interface  
[...]
```

Devices do not remain unmounted (udisks)

udisks remounts devices after a given period, or *polls* those devices. This can cause unexpected behaviour, for example when formatting drives, sharing them in a **virtual machine**, power saving, or removing a drive that was not detached with `--detach` before.

To disable polling for a given device, for example a CD/DVD device:

```
# udisks --inhibit-polling /dev/sr0
```

or for all devices:

```
# udisks --inhibit-all-polling
```

See **udisks(8)** (<https://jlk.fjfi.cvut.cz/arch/manpages/man/udisks.8>) for more information.

Broken standby timer (udisks2)

The udisks daemon polls **S.M.A.R.T.** data from drives regularly. Hard drives with a longer standby timeout than the polling interval may fail to enter standby. Drives that are already spun down are usually not effected. There seems no way to disable polling or change the interval as for **udisks2** (<https://www.archlinux.org/packages/?name=udisks2>) by now. See **[4]** (<https://bugs.launchpad.net/ubuntu/+source/udisks2/+bug/1281588>), **[5]** (https://bugs.freedesktop.org/show_bug.cgi?id=26508).

However, Standby timeout applied by udisks2 seems to be unaffected. To set standby timeout via udisks, see **#Apply ATA settings (udisks2)**.

Other possible workarounds could be setting the timeout below the polling interval (10 minutes) or forcing a manual spindown using `hdparm -y /dev/sdx`.

See also

- **gentoo wiki: udisks** (<http://wiki.gentoo.org/wiki/Udisks>)
- **Introduction to udisks** (<http://blog.fpmurphy.com/2011/08/introduction-to-udisks.html?output=pdf>)

Retrieved from "<https://wiki.archlinux.org/index.php?title=Udisks&oldid=505627>"

- This page was last edited on 31 December 2017, at 21:03.
- Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.