

Persistent block device naming

This article describes how to use persistent names for your block devices. This has been made possible by the introduction of udev and has some advantages over bus-based naming. If your machine has more than one SATA, SCSI or IDE disk controller, the order in which their corresponding device nodes are added is arbitrary. This may result in device names like `/dev/sda` and `/dev/sdb` switching around on each boot, culminating in an unbootable system, kernel panic, or a block device disappearing. Persistent naming solves these issues.

Related articles

[fstab](#)

[udev](#)

[LVM](#)

Note:

- Persistent naming has limits that are out-of-scope in this article. For example, while [mkinitcpio](#) may support a method, systemd may impose its own limits (e.g. [FS#42884](#) (<https://bugs.archlinux.org/task/42884>)) on naming it can process during boot.
- If you are using [LVM2](#), this article is not relevant as LVM takes care of this automatically.

Contents

- 1 Persistent naming methods
 - 1.1 by-label
 - 1.2 by-uuid
 - 1.3 by-id and by-path
 - 1.4 by-partlabel
 - 1.5 by-partuuid
 - 1.6 Static device names with Udev
- 2 Using persistent naming
 - 2.1 fstab
 - 2.2 Boot managers

Persistent naming methods

There are four different schemes for persistent naming: **by-label**, **by-uuid**, **by-id** and **by-path**. For those using disks with **GUID Partition Table (GPT)**, two additional schemes can be used **by-partlabel** and **by-partuuid**. You can also use **static device names by using Udev**.

Note: Beware that **Disk cloning** creates two different disks with the same name.

The following sections describes what the different persistent naming methods are and how they are used.

The `lsblk -f` command can be used for viewing graphically the first persistent schemes:

```
$ lsblk -f
```

NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
sda				
-sda1	vfat		CBB6-24F2	/boot
-sda2	ext4	SYSTEM	0a3407de-014b-458b-b5c1-848e92a327a3	/
-sda3	ext4	DATA	b411dc99-f0a0-4c87-9e05-184977be8539	/home
-sda4	swap		f9fe0b69-a280-415d-a03a-a32752370dee	[SWAP]

For those using **GPT**, use the `blkid` command instead. The latter is more convenient for scripts, but more difficult to read.

```
$ blkid
```

```
/dev/sda1: UUID="CBB6-24F2" TYPE="vfat" PARTLABEL="EFI SYSTEM PARTITION" PARTUUID="d0d0d110-0a71-4ed6-936a-304969ea36af"
/dev/sda2: LABEL="SYSTEM" UUID="0a3407de-014b-458b-b5c1-848e92a327a3" TYPE="ext4" PARTLABEL="GNU/LINUX" PARTUUID="98a81274-10f7-40db-872a-03df048df366"
/dev/sda3: LABEL="DATA" UUID="b411dc99-f0a0-4c87-9e05-184977be8539" TYPE="ext4" PARTLABEL="HOME" PARTUUID="7280201c-fc5d-40f2-a9b2-466611d3d49e"
/dev/sda4: UUID="f9fe0b69-a280-415d-a03a-a32752370dee" TYPE="swap" PARTLABEL="SWAP" PARTUUID="039b6c1c-7553-4455-9537-1befbc9fbc5b"
```

by-label

Almost every **filesystem** type can have a label. All your partitions that have one are listed in the `/dev/disk/by-label` directory. This directory is created and destroyed dynamically, depending on whether you have partitions with labels attached.

```
$ ls -l /dev/disk/by-label
```

```
total 0
lrwxrwxrwx 1 root root 10 May 27 23:31 DATA -> ../../sda3
lrwxrwxrwx 1 root root 10 May 27 23:31 SYSTEM -> ../../sda2
```

The labels of your filesystems can be changed. Following are some methods for changing labels on common filesystems:

swap

`swapon -L <label> /dev/XXX` using [util-linux](https://www.archlinux.org/packages/?name=util-linux) (<https://www.archlinux.org/packages/?name=util-linux>)

ext2/3/4

`e2label /dev/XXX <label>` using [e2fsprogs](https://www.archlinux.org/packages/?name=e2fsprogs) (<https://www.archlinux.org/packages/?name=e2fsprogs>)

btrfs

`btrfs filesystem label /dev/XXX <label>` using [btrfs-progs](https://www.archlinux.org/packages/?name=btrfs-progs) (<https://www.archlinux.org/packages/?name=btrfs-progs>)

reiserfs

`reiserfstune -l <label> /dev/XXX` using [reiserfsprogs](https://www.archlinux.org/packages/?name=reiserfsprogs) (<https://www.archlinux.org/packages/?name=reiserfsprogs>)

jfs

`jfs_tune -L <label> /dev/XXX` using [jfsutils](https://www.archlinux.org/packages/?name=jfsutils) (<https://www.archlinux.org/packages/?name=jfsutils>)

xfs

`xfs_admin -L <label> /dev/XXX` using **xfsprogs** (<https://www.archlinux.org/packages/?name=xfsprogs>)

fat/vfat

`fatlabel /dev/XXX <label>` using **dosfstools** (<https://www.archlinux.org/packages/?name=dosfstools>)

`mlabel -i /dev/XXX ::<label>` using **mttools** (<https://www.archlinux.org/packages/?name=mttools>)

exfat

`exfatlabel /dev/XXX <label>` using **exfat-utils** (<https://www.archlinux.org/packages/?name=exfat-utils>)

ntfs

`ntfslabel /dev/XXX <label>` using **ntfs-3g** (<https://www.archlinux.org/packages/?name=ntfs-3g>)

zfs

this filesystem does not support `/dev/disk/by-label`, but **#by-partlabel** may be used

Note:

- Changing the filesystem label of the root partition has to be done from a "live" GNU/Linux distribution because the partition needs to be unmounted first.
- Labels have to be unambiguous to prevent any possible conflicts.
- Labels can be up to 16 characters long.
- Since the label is a property of the filesystem, it is not suitable for addressing a single RAID device persistently.

- When using encrypted containers with **dm-crypt**, the labels of filesystems inside of containers are not available while the container is locked/encrypted.

by-uuid

UUID is a mechanism to give each **filesystem** a unique identifier. These identifiers are generated by filesystem utilities (e.g. `mkfs.*`) when the partition gets formatted and are designed so that collisions are unlikely. All GNU/Linux filesystems (including swap and LUKS headers of raw encrypted devices) support UUID. FAT, exFAT and NTFS filesystems do not support UUID, but are still listed in `/dev/disk/by-uuid/` with a shorter UID (unique identifier):

```
$ ls -l /dev/disk/by-uuid/

total 0
lrwxrwxrwx 1 root root 10 May 27 23:31 0a3407de-014b-458b-b5c1-848e92a327a3 -> ../../sda2
lrwxrwxrwx 1 root root 10 May 27 23:31 b411dc99-f0a0-4c87-9e05-184977be8539 -> ../../sda3
lrwxrwxrwx 1 root root 10 May 27 23:31 CBB6-24F2 -> ../../sda1
lrwxrwxrwx 1 root root 10 May 27 23:31 f9fe0b69-a280-415d-a03a-a32752370dee -> ../../sda4
```

The advantage of using the UUID method is that it is much less likely that name collisions occur than with labels. Further, it is generated automatically on creation of the filesystem. It will, for example, stay unique even if the device is plugged into another system (which may perhaps have a device with the same label).

The disadvantage is that UUIDs make long code lines hard to read and break formatting in many configuration files (e.g. **fstab** or **crypttab**). Also every time a partition is resized or reformatted a new UUID is generated and configs have to get adjusted (manually).

Tip: In case your swap partition does not have an UUID assigned, you will need to reset the swap partition using **mkswap** utility.

by-id and by-path

by-id creates a unique name depending on the hardware serial number, **by-path** depending on the shortest physical path (according to sysfs). Both contain strings to indicate which subsystem they belong to (i.e. **-ide-** for **by-path**, and **-ata-** for **by-id**), so they are linked to the hardware controlling the device. This implies different levels of persistence: the **by-path** will already change when the device is plugged into a different port of the controller, the **by-id** will change when the device is plugged into a port of a hardware controller subject to another subsystem. **[1]** (https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Online_Storage_Reconfiguration_Guide/persistent_naming.html) Thus, both are not suitable to achieve persistent naming tolerant to hardware changes.

However, both provide important information to find a particular device in a large hardware infrastructure. For example, if you do not manually assign persistent labels (**by-label** or **by-partlabel**) and keep a directory with hardware port usage, **by-id** and **by-path** can be

used to find a particular device. [2] (<http://linuxshellaccount.blogspot.in/2008/09/how-to-easily-find-wwns-of-qlogic-hba.html>) [3] (<http://www.linuxquestions.org/questions/linux-server-73/how-to-find-wwn-for-dev-sdc-917269/>)

```
$ ls -l /dev/disk/by-id/
```

```
total 0
lrwxrwxrwx 1 root root 10 May 27 23:31 ata-VBOX_HARDDISK_VBe4474842-dc40379f -> ../../sda
lrwxrwxrwx 1 root root 10 May 27 23:31 ata-VBOX_HARDDISK_VBe4474842-dc40379f-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 May 27 23:31 ata-VBOX_HARDDISK_VBe4474842-dc40379f-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 May 27 23:31 ata-VBOX_HARDDISK_VBe4474842-dc40379f-part3 -> ../../sda3
lrwxrwxrwx 1 root root 10 May 27 23:31 ata-VBOX_HARDDISK_VBe4474842-dc40379f-part4 -> ../../sda4
```

```
$ ls -l /dev/disk/by-path/
```

```
total 0
lrwxrwxrwx 1 root root 10 May 27 23:31 pci-0000:00:1f.2-ata-1 -> ../../sda
lrwxrwxrwx 1 root root 10 May 27 23:31 pci-0000:00:1f.2-ata-1-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 May 27 23:31 pci-0000:00:1f.2-ata-1-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 May 27 23:31 pci-0000:00:1f.2-ata-1-part3 -> ../../sda3
lrwxrwxrwx 1 root root 10 May 27 23:31 pci-0000:00:1f.2-ata-1-part4 -> ../../sda4
```

by-partlabel

Note: This method only concerns disks with **GUID Partition Table (GPT)**.

Partition labels can be defined in the header of the partition entry on GPT disks.

See also [Wikipedia:GUID Partition Table#Partition entries](#).

This method is very similar to the **filesystem labels**, excepted that the dynamic directory is `/dev/disk/by-partlabel`.

```
ls -l /dev/disk/by-partlabel/

total 0
lrwxrwxrwx 1 root root 10 May 27 23:31 EFI\x20SYSTEM\x20PARTITION -> ../../sda1
lrwxrwxrwx 1 root root 10 May 27 23:31 GNU\x2fLINUX -> ../../sda2
lrwxrwxrwx 1 root root 10 May 27 23:31 HOME -> ../../sda3
lrwxrwxrwx 1 root root 10 May 27 23:31 SWAP -> ../../sda4
```

Note:

- GPT partition labels have also to be different to avoid conflicts. To change your partition label, you can use `gdisk` or the ncurses-based version `cgdisk`. Both are available from the **gptfdisk** (<https://www.archlinux.org/packages/?name=gptfdisk>) package. See **Partitioning#Partitioning tools**.
- According to the specification, GPT partition labels can be up to 72 characters long.

by-partuuid

Note: This method only concerns disks with **GUID Partition Table (GPT)**.

Like **GPT partition labels**, GPT partition UUID are defined in the partition entry on GPT disks.

See also [Wikipedia:GUID Partition Table#Partition entries](#).

The dynamic directory is similar to other methods and, like [UUID filesystems](#), using UUIDs is preferred over labels.

```
ls -l /dev/disk/by-partuuid/
```

```
total 0
lrwxrwxrwx 1 root root 10 May 27 23:31 039b6c1c-7553-4455-9537-1befbc9fbc5b -> ../../sda4
lrwxrwxrwx 1 root root 10 May 27 23:31 7280201c-fc5d-40f2-a9b2-466611d3d49e -> ../../sda3
lrwxrwxrwx 1 root root 10 May 27 23:31 98a81274-10f7-40db-872a-03df048df366 -> ../../sda2
lrwxrwxrwx 1 root root 10 May 27 23:31 d0d0d110-0a71-4ed6-936a-304969ea36af -> ../../sda1
```

Static device names with Udev

See [udev#Setting static device names](#).

Using persistent naming

There are various applications that can be configured using persistent naming. Following are some examples of how to configure them.

fstab

See the main article: [fstab#UUIDs](#)

Boot managers

To use persistent names in the **boot manager (boot loader)**, the following prerequisites must be met. On a standard installation following the installation guide both prerequisites are met.

- You are using a **mkinitcpio** initial RAM disk image
- You have udev enabled in `/etc/mkinitcpio.conf`

The location of the root filesystem is given by the parameter `root` on the kernel commandline. The kernel commandline is configured from the bootloader, see **Kernel parameters#Configuration**. To change to persistent device naming, only change the parameters which specify block devices, e.g. `root` and `resume`, while leaving other parameters as is. Various naming schemes are supported:

Non-persistent device naming using the `/dev` path format, in this example `/dev/sdr1` is the root partition.

```
root=/dev/sdr1
```

Persistent device naming using label and the `/dev` path format, in this example `root_myhost` is the label of the root partition.

```
root=/dev/disk/by-label/root_myhost
```

Persistent device naming using label and the `LABEL=` format.

```
root=LABEL=root_myhost
```

Persistent device naming using UUID and the `UUID=` format, in this example `1234-5678` is the UUID of the root partition.

```
root=UUID=1234-5678
```

Persistent device naming using the PARTUUID and the `PARTUUID=` format.

```
root=PARTUUID=1234-5678
```

Persistent device naming using the PARTLABEL and the `PARTLABEL=` format.

```
root=PARTLABEL=root_myhost
```

Retrieved from "https://wiki.archlinux.org/index.php?title=Persistent_block_device_naming&oldid=499900"

- This page was last edited on 28 November 2017, at 10:05.

- Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.