



udev

From Gentoo Wiki



Not to be confused with eudev (/wiki/Eudev).

udev (user /dev) is systemd's device manager for the Linux kernel. It manages device nodes in /dev and handles all user space actions when adding or removing devices.

Contents

- 1 What is udev?
 - 1.1 The /dev directory
- 2 Installation
 - 2.1 Kernel
 - 2.2 USE flags
 - 2.3 Emerge
- 3 Configuration
 - 3.1 Service
 - 3.2 Keep classic 'eth0' naming
- 4 Advanced Configuration
 - 4.1 Rules
 - 4.2 Persistent device names
- 5 Usage
- 6 Troubleshooting
 - 6.1 Log monitor messages
 - 6.2 Debug mode
 - 6.3 Missing device files /dev/null and /dev/console
 - 6.4 NIC assigned eth0, but is moved to eth1
- 7 See also
- 8 External resources

Resources

 Home (https://www.freedesktop.org/wiki/Software/systemd/)
 Wikipedia (https://en.wikipedia.org/wiki/Udev)
 GitWeb (https://cgit.freedesktop.org/systemd/systemd/tree/src/udev)
 Project (/wiki/Project:Udev) (/wiki/)

What is udev?

The /dev directory

Most Linux users understand that `/dev/sda1` is just a fast way of referring to the first partition on the first disk that the kernel found. That's pretty easy, right?

But consider hotpluggable devices like USB, IEEE1394, hot-swappable PCI, etc. What is the first device for each of these? And for how long? What will the other devices be named when the first one disappears? How will that affect ongoing transactions? Wouldn't it be fun if a printing job were suddenly moved from a high-end laser printer to an almost-dead matrix printer just because someone's mom decided to pull the plug on the laser printer (which just happened to be the first printer)?

Enter the device manager. A modern device manager (including **udev** and **eudev** (</wiki/Eudev>)) must:

- Run in userspace;
- Dynamically create and remove device files (/wiki/Device_file);
- Provide consistent device naming;
- Provide a userspace application program interface (API).

Every time a change happens within the device structure, the kernel emits a *uevent* which gets picked up by the device manager. The device manager then follows the rules declared in the `/etc/udev/rules.d`, `/run/udev/rules.d` and `/lib/udev/rules.d` directories. Based on the information contained within the *uevent*, it finds the rule or rules it needs to trigger and performs the required actions. These actions may involve the creation or deletion of device files, and may also trigger the loading of particular firmware files into kernel memory.

Installation

❗ Important

When updating **udev**, check the **udev upgrade guide** (/wiki/Udev/Upgrade_Guide) for information that can prevent unbootable systems.

Kernel

udev requires the following kernel options:

KERNEL

```
General setup --->
[*] Configure standard kernel features (expert users) --->
```

```

    [ ] Enable deprecated sysfs features to support old userspace tools
    [*] Enable signalfd() system call
Enable the block layer --->
    [*] Block layer SG support v4
Networking support --->
    Networking options --->
        <*> Unix domain sockets
Device Drivers --->
    Generic Driver Options --->
        () path to uevent helper
        [*] Maintain a devtmpfs filesystem to mount at /dev
    < > ATA/ATAPI/MFM/RLL support (DEPRECATED) --->
File systems --->
    [*] Inotify support for userspace
Pseudo filesystems --->
    [*] /proc file system support
    [*] sysfs file system support

```

USE flags

Portage knows the `udev` global USE flag for enabling support for udev in other packages. Adding this USE flag value to the USE flag list (default in all Linux profiles (`/wiki/Profile_(Portage)`)) will pull in the `sys-fs/udev` (<https://packages.gentoo.org/packages/sys-fs/udev>) package automatically:

FILE `/etc/portage/make.conf`

```
USE="udev"
```

USE flags for sys-fs/udev (<https://packages.gentoo.org/packages/sys-fs/udev>) Linux dynamic and persistent device naming ...

ac1 (https://packages.gentoo.org/useflags/ac1)	Add support for Access Control Lists	global
kmod (https://packages.gentoo.org/useflags/kmod)	Enable kernel module loading/unloading support using sys-apps/kmod	local
selinux (https://packages.gentoo.org/useflags/selinux)	!!internal use only!! Security Enhanced Linux support, this must be set by the selinux profile or breakage will occur	global

More information about USE flags

Data provided by the Gentoo Package Database (<https://packages.gentoo.org>) · Last update: 2018-02-04 00:56

Emerge

After setting USE flags update the system so the changes take effect:

```
root # emerge --ask --changed-use --deep @world
```

Configuration

Service

To start udev at boot time, add it to the sysinit runlevel. This can be done by issuing the following command with root privileges:

```
root # rc-update add udev sysinit
```

Keep classic 'eth0' naming

Network device names `eth0`, `wlan0`, etc. as provided by the kernel could be changed on boot (see `dmesg`) by the `/lib/udev/rules.d/80-net-name-slot.rules` udev rule.

To keep the classic naming this rule can be overwritten with an equally named empty file in the `/etc/udev/rules.d` directory:

```
root # touch /etc/udev/rules.d/80-net-name-slot.rules
```

Advanced Configuration

Rules

udev provides a set of rules that match against exported values of uevents (events sent by the kernel) and properties of the discovered device. A matching rule will possibly name and create a device node and run configured programs to setup and configure the device.

The rule definitions are stored in two locations:

1. `/lib/udev/rules.d/` - Rules in this directory are installed by certain packages, they generally should not be changed by users;
2. `/etc/udev/rules.d/` - This folder is for end-user specified rules. Any new rules should be added in this directory;

In these directories, multiple rule files (with suffix `.rules`) are traversed in alphanumerical order. Inside the rules files, udev will find expressions that might match a uevent together with the state to match (is the uevent because a device is added or removed) and the command to execute.

The event matching is based on information such as:

- The *SUBSYSTEM* of the uevent (for which type of device is the uevent fired);
- The *ACTION* that is taken (add, change, or remove);
- One or more attributes (through *ATTR* or *ATTRS*), such as the device class, vendor or other device information;
- The kernel-provided name (through *KERNEL*), such as `sd*` (for SCSI/SATA disks) or `input*` (for input devices such as mice and keyboards);

- One or more environment settings (through *ENV*), used to send information between multiple rules.

Based on this information, the rule can then state if:

1. Some information needs to be shared with later events (through environment variables)
2. Links need to be created in */dev*
3. Commands need to be executed

Udev does this for every rule that matches (it does not stop after the first match) to allow a flexible device management approach.

Persistent device names

The kernel detects devices asynchronously, udev mirrors the kernel's sysfs (</wiki/Sysfs>) filesystem and so the devices are named and numbered in order of detection. So by default udev provides no persistent device names. However there are mechanisms for some device classes to provide these:

- Udev creates for storage devices additional symlinks based on the device's ID, label, UUID (/wiki/Removable_media#UUIDs_and_labels) and path. See the */dev/disk/by-** directory. So instead of using e.g. the device file */dev/sda* use the file */dev/disk/by-label/SOME_LABEL*.
- The same for input devices in the */dev/input* directory.
- Using custom rules enables users to create their own device files.

Usage

Some useful commands are:

- Show all messages about a given device file:

```
root # udevadm info --query=all --name=/dev/DEVICE_FILE
```

- Monitor udev activities:

```
root # udevadm monitor
```

- Assign a persistent name to an Ethernet device:


FILE `/etc/udev/rules.d/ethernet.rules`

```
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="01:23:45:67:89:ab", NAME="eth6789ab"
```

See the *udevadm* man page (/wiki/Man_page) for more information.

Troubleshooting

troubleshooting

-  Gentoo bugtracker: known bugs (https://bugs.gentoo.org/buglist.cgi?quicksearch=udev&order=bug_id%20DESC)

Log monitor messages

To log all message when **udevadm monitor** is ran, modify the following configuration file:

FILE `/etc/conf.d/udev`

```
udev_monitor="YES"
```

It will create the new log file located at `/run/udev/udevmonitor.log`

Debug mode

Enabling debug mode will output more log messages:

FILE `/etc/conf.d/udev`

```
udev_debug="YES"
```

Set the logging priority:

FILE `/etc/udev/udev.conf`

```
udev_log="debug"
```

The log file `/run/udevdebug.log` will be created but no messages will be logged to it. The most recent versions of udev will log all messages to **dmesg**.

Missing device files `/dev/null` and `/dev/console`

Some udev versions need the `/dev/null` and `/dev/console` files in order to work properly, but can not create them on their own. To manually create these files for udev run the following commands with root privileges:

```
root # mkdir test
root # mount --bind / test
root # cd test/dev
root # mknod -m 660 console c 5 1
root # mknod -m 660 null c 1 3
root # cd ../../
```

```
root # umount test
root # rmdir test
```

NIC assigned eth0, but is moved to eth1

Those having dual network cards on their motherboards may run into a situation where **ifconfig** may show no eth0 or eth1. **dmesg** may show their NIC detected as eth0, and later moved to eth1. Performing a **ifconfig -a** will also show the NIC as eth1. This is caused by using the kernel assigned names in the first place. Users should write custom rules like `/etc/udev/rules.d/70-my-network.rules` to use free names like `lan0` or `wireless0` or use predictable interface names (which have been enabled by default since udev version 197).

Remember to also remove old files from old versions of udev:

```
root # rm /etc/udev/rules.d/70-persistent-net.rules /etc/systemd/network/99-default.link
```

Also make sure not to pass `net.ifnames=0` on the kernel commandline. This setting would disable the predictable interface names feature of udev altogether.

See also

- [eudev \(/wiki/Eudev\)](/wiki/Eudev) — Gentoo's maintained fork of **udev**, systemd's device file (/wiki/Device_file) manager for the Linux kernel.

External resources

✱ Bug 575718 - Request for council decision regarding virtual/udev default provider (https://bugs.gentoo.org/show_bug.cgi?id=575718) - Default virtual/udev (<https://packages.gentoo.org/packages/virtual/udev>) provider changed to sys-fs/eudev (<https://packages.gentoo.org/packages/sys-fs/eudev>)

Retrieved from "<http://wiki.gentoo.org/index.php?title=Udev&oldid=706876> (<http://wiki.gentoo.org/index.php?title=Udev&oldid=706876>)"

Categories (</wiki/Special:Categories>):

Pages with syntax highlighting errors (/index.php?title=Category:Pages_with_syntax_highlighting_errors&action=edit&redlink=1)

| Documents containing Metadata (/wiki/Category:Documents_containing_Metadata) | Core system (/wiki/Category:Core_system)

| Daemons (</wiki/Category:Daemons>)

- This page was last modified on 30 January 2018, at 07:31.

© 2001–2018 Gentoo Foundation, Inc.

Gentoo is a trademark of the Gentoo Foundation, Inc. The contents of this document, unless otherwise expressly stated, are licensed under the CC-BY-SA-3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>) license. The Gentoo Name and Logo Usage Guidelines (<https://www.gentoo.org/inside-gentoo/foundation/name-logo-guidelines.html>) apply.