

How to make X listen on port 6000

http://gentoo.linuxhowtos.org/Misc/udev_guide.pdf

[jump to the content](#)

Gentoo.LinuxHowtos.org howtos, tips&tricks and tutorials for gentoo linux

from small one page howto to huge articles all in one place

Other .linuxhowtos.org sites:
toolsntoys.linuxhowtos.org

www.linuxhowtos.org

Last additions:

How to make X listen on port 6000

How to make X listen on port 6000

words:

34

views:

78201

userrating:

May, 25th 2007:

Words

496

why adblockers are bad

April, 26th 2007:

Words

77

Website translation planned

Apr, 10th. 2007:

Words

63

Compile Time Estimator integrated into genlop

image:Druckversion / /data/printer.gif (null)

.

image:pdf icon / /data/pdf.png (null)

You are here:

Gentoo udev Guide

1. What is udev?

The /dev Directory

When Linux-users talk about the hardware on their system in the vicinity of people who believe Linux is some sort of virus or brand of coffee, the use of "slash dev slash foo" will return a strange look for sure. But for the fortunate user (and that includes you) using /dev/hda1 is just a fast way of explaining that we are talking about the primary master IDE, first partition. Or aren't we?

We all know what a device file is. Some even know why device files have special numbers when we take a closer look at them when we issue `ls -l` in /dev. But what we always take for granted is that the primary master IDE disk is referred to as /dev/hda. You might not see it this way, but this is a flaw by design.

Think about hotpluggable devices like USB, IEEE1394, hot-swappable PCI, ... What is the first device? 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 that a printing job is suddenly moved from your supernew laserprinter to your almost-dead matrix printer because your mom decided to pull the plug of the laserprinter which happened to be the first printer?

Enter udev. The goals of the udev project are both interesting and needed:

- Runs in userspace
- Dynamically creates/removes device files
- Provides consistent naming
- Provides a user-space API

To provide these features, udev is developed in three separate projects: namedev, libsysfs and, of course, udev.

namedev

Namedev allows you to define the device naming separately from the udev program. This allows for flexible naming policies and naming schemes developed by separate entities. This device naming subsystem provides a standard interface that udev can use.

Currently only a single naming scheme is provided by namedev; the one provided by LANANA, used by the majority of Linux systems currently and therefore very suitable for the majority of Linux users.

Namedev uses a 5-step procedure to find out the name of a given device. If the device name is found in one of the given steps, that name is used. The steps are:

- label or serial number
- bus device number

How to make X listen on port 6000

http://gentoo.linuxhowtos.org/Misc/udev_guide.pdf

- bus topology
- statically given name
- kernel provided name

The label or serial number step checks if the device has a unique identifier. For instance USB devices have a unique USB serial number; SCSI devices have a unique UUID. If namedev finds a match between this unique number and a given configuration file, the name provided in the configuration file is used.

The bus device number step checks the device bus number. For non-hot-swappable environments this procedure is sufficient to identify a hardware device. For instance PCI bus numbers rarely change in the lifetime of a system. Again, if namedev finds a match between this position and a given configuration file, the name provided in that configuration file is used.

Likewise the bus topology is a rather static way of defining devices as long as the user doesn't switch devices. When the position of the device matches a given setting provided by the user, the accompanying name is used.

The fourth step, statically given name, is a simple string replacement. When the kernel name (the default name) matches a given replacement string, the substitute name will be used.

The final step (kernel provided name) is a catch-all: this one takes the default name provided by the kernel. In the majority of cases this is sufficient as it matches the device naming used on current Linux systems.

libsysfs

udev interacts with the kernel through the sysfs pseudo filesystem. The libsysfs project provides a common API to access the information given by the sysfs filesystem in a generic way. This allows for querying all kinds of hardware without having to make assumptions on the kind of hardware.

udev

Every time the kernel notices an update in the device structure, it calls the /sbin/hotplug program. Hotplug runs the applications linked in the /etc/hotplug.d/default directory where you will also find a symlink to the udev application. Hotplug directs the information given by the kernel to the udev application which performs the necessary actions on the /dev structure (creating or deleting device files).

2. Using udev on Gentoo

Requirements

udev is meant to be used in combination with a 2.6 kernel (like development-sources or gentoo-dev-sources). If you're using such a kernel then you just have to make sure that you have a recent sys-apps/baselayout version. That's all you need.

Code Listing 2.1: Installing udev

```
# emerge udev
```

How to make X listen on port 6000

http://gentoo.linuxhowtos.org/Misc/udev_guide.pdf

udev will install hotplug-base as one of its dependencies. You do not need to install hotplug unless you want your modules automatically loaded when you plug devices in. hotplug also handles the automated bringup of network devices and firmware downloading.

Code Listing 2.2: Installing optional hotplug scripts

```
# emerge hotplug
```

If you want modules loaded for devices that have been plugged in before you boot, use the coldplug package:

Code Listing 2.3: Installing the coldplug package

```
# emerge coldplug
```

Kernelwise, if you're using the default set by genkernel then you're all set. Otherwise be sure to activate the following options:

Code Listing 2.4: Required kernel options

```
General setup --->
[*] Support for hot-pluggable devices

File systems --->
Pseudo filesystems --->
[*] /proc file system support
[*] Virtual memory file system support (former shm fs)
```

You can leave the /dev file system support (OBSOLETE) active if you wish but you have to make sure that "Automatically mount at boot" is disabled:

Code Listing 2.5: Don't automatically mount devfsd

```
File systems --->
Pseudo Filesystems --->
[*] /dev file system support (OBSOLETE)
[ ] Automatically mount at boot
```

Configuration

If you want to use the udev-tweaks Gentoo added to make your life comfortable, then read no more. Gentoo will use udev but keep a static /dev so that you will never have any missing device nodes. The Gentoo init scripts won't run the devfsd daemon and will deactivate devfs when you boot up.

But if you are a die-hard and want to run a udev-only, no-tweaked system as is intended by the udev development (including the difficulties of missing device nodes because udev doesn't support them yet), by all means, read on :)

How to make X listen on port 6000

http://gentoo.linuxhowtos.org/Misc/udev_guide.pdf

We'll deactivate the rules that save the device file nodes: edit the `RC_DEVICE_TARBALL` variable in `/etc/conf.d/rc` and set it to `no`:

Code Listing 2.6: `/etc/conf.d/rc`

```
RC_DEVICE_TARBALL="no"
```

If you have included `devfs` support in your kernel, you can deactivate it in the bootloader configuration: add `gentoo=nodevfs` as a kernel parameter. If you want to use `devfs` and deactivate `udev`, add `gentoo=noudev` as kernel parameter.

3. Known Issues

Missing device node files at boot

If you can't boot successfully because you get an error about `/dev/null` not found, or because the initial console is missing, the problem is that you lack some device files that must be available before `/dev` is mounted and handled by `udev`. This is common on Gentoo machines installed from old media.

If you run `sys-apps/baselayout-1.8.12` or later, this problem is alleviated since the boot process should still manage to complete. However, to get rid of those annoying warnings, you should create the missing device nodes as described below.

To see which devices nodes are present before the `/dev` filesystem is mounted, run the following commands:

Code Listing 3.1: Listing device nodes available at boot

```
# mkdir test
# mount --bind / test
# cd test/dev
# ls
```

The devices needed for a successful boot are `/dev/null` and `/dev/console`. If they didn't show up in the previous test, you have to create them manually. Issue the following commands in the `test/dev/` directory:

Code Listing 3.2: Creating necessary device node files

```
# mknod -m 660 console c 5 1
# mknod -m 660 null c 1 3
```

When you're finished, don't forget to unmount the `test/` directory:

Code Listing 3.3: Unmounting the `test/` directory

```
# cd ../../
# umount test
```

```
# rmdir test
```

udev and nvidia

If you use the proprietary driver from nVidia and the X server fails to start on a udev-only system, then make sure you have:

the nvidia module listed in `/etc/modules.autoload.d/kernel-2.6`

a version of nvidia-kernel equal to or greater than `media-video/nvidia-kernel-1.0.5336-r2`

a version of baselayout equal to or greater than `sys-apps/baselayout-1.8.12`

LVM2 Names Disappear

When you use udev and LVM2 together, you might notice that your created volume groups and logical volumes have disappeared. Well, they haven't, but they are unfortunately named `/dev/dm-#` with # being 0, 1, ...

To fix this, edit `/etc/udev/rules.d/50-udev.rules` and uncomment the following line:

Code Listing 3.4: Uncomment this line from `/etc/udev/rules.d/50-udev.rules`

```
KERNEL="dm-[0-9]*", PROGRAM="/sbin/devmap_name %M %m", NAME="%k", SYMLINK="%c"
```

No Consistent Naming between DevFS and udev

Even though our intention is to have a consistent naming scheme between both dynamical device management solutions, sometimes naming differences do occur. One reported clash is with a HP Smart Array 5i RAID controller (more precisely the cciss kernel module). With udev, the devices are named `/dev/cciss/cXdYpZ` with X, Y and Z regular numbers. With devfs, the devices are `/dev/hostX/targetY/partZ` or symlinked from `/dev/cciss/cXdY`.

If this is the case, don't forget to update your `/etc/fstab` and bootloader configuration files accordingly.

Other issues

If device nodes are not created when a module is loaded from `/etc/modules.autoload.d/kernel-2.6` but they appear when you load the module manually with `modprobe` then you should try upgrading to `sys-apps/baselayout-1.8.12` or later.

Support for the framebuffer devices (`/dev/fb/*`) comes with the kernel starting from version 2.6.6-rc2.

For kernels older than 2.6.4 you have to explicitly include support for the `/dev/pts` filesystem.

How to make X listen on port 6000

http://gentoo.linuxhowtos.org/Misc/udev_guide.pdf

Code Listing 3.5: Enabling the /dev/pts filesystem

```
File systems --->
Pseudo filesystems --->
    [*] /dev/pts file system for Unix98 PTYs
```

4. Resources & Acknowledgements

The udev talk on the Linux Symposium (Ottawa, Ontario Canada - 2003) given by Greg Kroah-Hartman (IBM Corporation) provided a solid understanding on the udev application.

Decibel's UDEV Primer is an in-depth document about udev and Gentoo.

Writing udev rules by fellow Gentoo developer Daniel Drake is an excellent document to learn how to customize your udev installation. current rating:

image:Support us on Content Nation / /images/cn.png (null)

<!--

Other free services

toURL.org

Shorten long

URLs to short

links like

<http://turl.org/2>

turl.org

.

Reverse DNS lookup

Find out which hostname(s)

resolve to a

given IP or other hostnames for the server

www.reversednslookup.org

-->

New Packages

- as

image:rdf newsfeed / /data/rdf.png (null)

- as

image:rss newsfeed / /data/rss.png (null)

- as

image:Atom newsfeed / /data/atom.png (null)

How to make X listen on port 6000

http://gentoo.linuxhowtos.org/Misc/udev_guide.pdf

image:rdf newsfeed //data/rdf.png (null)

|

image:rss newsfeed //data/rss.png (null)

|

image:Atom newsfeed //data/atom.png (null)

- Powered by

image:LeopardCMS //data/leopardcms.png (null)

- Running on

image:Gentoo //data/gentoo.png (null)

-

Copyright 2004-2020 Sascha Nitsch Unternehmensberatung GmbH

image:Valid XHTML1.1 //data/xhtml.png (null)

:

image:Valid CSS //data/css.png (null)

:

image:buttonmaker //data/buttonmaker.png (null)

- Level Triple-A Conformance to Web Content Accessibility Guidelines 1.0 -

- Copyright and legal notices -

Time to create this page: ms

<!--

image:system status display /status/output.jpg (null)

-->