



# Configuring the system

From Gentoo Wiki

< Handbook:AMD64 (/wiki/Special:MyLanguage/Handbook:AMD64) | Installation (/wiki/Special:MyLanguage/Handbook:AMD64/Installation)

Jump to:navigation Jump to:search

## Contents

- 1 Filesystem information
  - 1.1 Filesystem labels and UUIDs
  - 1.2 Partition labels and UUIDs
  - 1.3 About fstab
  - 1.4 Creating the fstab file
    - 1.4.1 DOS/Legacy BIOS systems
    - 1.4.2 UEFI systems
    - 1.4.3 DPS UEFI PARTUUID
- 2 Networking information
  - 2.1 Hostname
    - 2.1.1 Set the hostname (OpenRC or systemd)
    - 2.1.2 systemd
  - 2.2 Network
    - 2.2.1 DHCP via dhcpcd (any init system)
    - 2.2.2 netifrc (OpenRC)
      - 2.2.2.1 Configuring the network
      - 2.2.2.2 Automatically start networking at boot
  - 2.3 The hosts file
- 3 System information
  - 3.1 Root password
  - 3.2 Init and boot configuration
    - 3.2.1 OpenRC
    - 3.2.2 systemd

## Filesystem information

### Filesystem labels and UUIDs

Both MBR (BIOS) and GPT include support for *filesystem* labels and *filesystem* UUIDs. These attributes can be defined in `/etc/fstab` as alternatives for the `mount` command to use when attempting to find and mount block devices. Filesystem labels and UUIDs are identified by the *LABEL* and *UUID* prefix and can be viewed with the `blkid` command:

```
root # blkid
```

**Warning**

If the filesystem inside a partition is wiped, then the filesystem label and the UUID values will be subsequently altered or removed.

For uniqueness, readers who are using MBR-style partition tables are advised to use UUIDs rather than labels to specify mountable volumes in `/etc/fstab`.

**Important**

UUIDs of the filesystem on a LVM volume and its LVM snapshots are identical, therefore using UUIDs to mount LVM volumes should be avoided.

**AMD64 Handbook (/wiki/Handbook:AMD64)**  
**Installation (/wiki/Handbook:AMD64/Full/Installation)**  
About the installation (/wiki/Handbook:AMD64/Installation/About)  
Choosing the media (/wiki/Handbook:AMD64/Installation/Media)  
Configuring the network (/wiki/Handbook:AMD64/Installation/Networking)  
Preparing the disks (/wiki/Handbook:AMD64/Installation/Disks)  
The stage file (/wiki/Handbook:AMD64/Installation/Stage)  
Installing base system (/wiki/Handbook:AMD64/Installation/Base)  
Configuring the kernel (/wiki/Handbook:AMD64/Installation/Kernel)  
Configuring the system  
Installing tools (/wiki/Handbook:AMD64/Installation/Tools)  
Configuring the bootloader (/wiki/Handbook:AMD64/Installation/Bootloader)  
Finalizing (/wiki/Handbook:AMD64/Installation/Finalizing)

**Working with Gentoo (/wiki/Handbook:AMD64/Full/Working)**  
Portage introduction (/wiki/Handbook:AMD64/Working/Portage)  
USE flags (/wiki/Handbook:AMD64/Working/USE)  
Portage features (/wiki/Handbook:AMD64/Working/Features)  
Initscript system (/wiki/Handbook:AMD64/Working/Initscripts)  
Environment variables (/wiki/Handbook:AMD64/Working/EnvVar)  
**Working with Portage (/wiki/Handbook:AMD64/Full/Portage)**

## Partition labels and UUIDs

Systems with GPT disklabel support offer additional 'robust' options to define partitions in **/etc/fstab**. Partition labels and partition UUIDs can be used to identify the block device's individual partition(s), regardless of what filesystem has been chosen for the partition itself. Partition labels and UUIDs are identified by the *PARTLABEL* and/or *PARTUUID* prefixes and can be viewed nicely in the terminal by running the **blkid** command.

Output for an **amd64** EFI system using the Discoverable Partition Specification UUIDs may like the following:

```
root # blkid
```

```
/dev/sr0: BLOCK_SIZE="2048" UUID="2023-08-28-03-54-40-00" LABEL="ISOIMAGE"
TYPE="iso9660" PTTYPE="PMBR"
/dev/loop0: TYPE="squashfs"
/dev/sda2: PARTUUID="0657fd6d-a4ab-43c4-84e5-0933c84b4f4f"
/dev/sda3: PARTUUID="1cdf763a-5b4c-4dbf-99db-a056c504e8b2"
/dev/sda1: PARTUUID="c12a7328-f81f-11d2-ba4b-00a0c93ec93b"
```

While not always true for partition labels, using a UUID to identify a partition in **fstab** provides a guarantee that the bootloader will not be confused when looking for a certain volume, even if the *filesystem* is changed or re-written in the future. Using the older default block device files (**/dev/sd\*N**) for defining the partitions in **fstab** is risky for systems that have SATA block devices regularly added or removed.

The naming for block device files depends on a number of factors, including how and in what order the disks are attached to the system. They also could show up in a different order depending on which of the devices are detected by the kernel first during the early boot process. With this being stated, unless the system administrator intends to constantly fiddle with the disk ordering, using default block device files is a simple and straightforward approach.

## About fstab

Under Linux, all partitions used by the system must be listed in **/etc/fstab** ([/wiki//etc/fstab](#)). This file contains the mount points of those partitions (where they are seen in the file system structure), how they should be mounted and with what special options (automatically or not, whether users can mount them or not, etc.)

## Creating the fstab file

### Note

If the init system being used is systemd, the partition UUIDs conform to the Discoverable Partition Specification as given in [Preparing the disks \(/wiki/Handbook:AMD64/Installation/Disks\)](#), and the system uses UEFI, then creating an fstab can be skipped, since systemd auto-mounts partitions that follow the spec.

The **/etc/fstab** file uses a table-like syntax. Every line consists of six fields, separated by whitespace (space(s), tabs, or a mixture of the two). Each field has its own meaning:

1. The first field shows the block special device or remote filesystem to be mounted. Several kinds of device identifiers are available for block special device nodes, including paths to device files, filesystem labels and UUIDs, and partition labels and UUIDs.
2. The second field shows the mount point at which the partition should be mounted.
3. The third field shows the type of filesystem used by the partition.
4. The fourth field shows the mount options used by **mount** when it wants to mount the partition. As every filesystem has its own mount options, so system admins are encouraged to read the mount man page (**man mount**) for a full listing. Multiple mount options are comma-separated.
5. The fifth field is used by **dump** to determine if the partition needs to be dumped or not. This can generally be left as **0** (zero).
6. The sixth field is used by **fsck** to determine the order in which filesystems should be checked if the system wasn't shut down properly. The root filesystem should have **1** while the rest should have **2** (or **0** if a filesystem check is not necessary).

### Important

The default **/etc/fstab** file provided in Gentoo stage files is *not* a valid **fstab** file but instead a template that can be used to enter in relevant values.

```
root # nano /etc/fstab
```

[Files and directories \(/wiki/Handbook:AMD64/Portage/Files\)](#)  
[Variables \(/wiki/Handbook:AMD64/Portage/Variables\)](#)  
[Mixing software branches \(/wiki/Handbook:AMD64/Portage/Branches\)](#)  
[Additional tools \(/wiki/Handbook:AMD64/Portage/Tools\)](#)  
[Custom package repository \(/wiki/Handbook:AMD64/Portage/CustomTree\)](#)  
[Advanced features \(/wiki/Handbook:AMD64/Portage/Advanced\)](#)  
**OpenRC network configuration (/wiki/Handbook:AMD64/Full/Networking)**  
[Getting started \(/wiki/Handbook:AMD64/Networking/Introduction\)](#)  
[Advanced configuration \(/wiki/Handbook:AMD64/Networking/Advanced\)](#)  
[Modular networking \(/wiki/Handbook:AMD64/Networking/Modular\)](#)  
[Wireless \(/wiki/Handbook:AMD64/Networking/Wireless\)](#)  
[Adding functionality \(/wiki/Handbook:AMD64/Networking/Extending\)](#)  
[Dynamic management \(/wiki/Handbook:AMD64/Networking/Dynamic\)](#)

## DOS/Legacy BIOS systems

Let us take a look at how to write down the options for the **/boot** partition. This is just an example, and should be modified according to the partitioning decisions made earlier in the installation. In the **amd64** partitioning example, **/boot** is usually the **/dev/sda1** partition, with **xfs** recommended for the filesystem. It needs to be checked during boot, so we would write down:

### FILE /etc/fstab An example DOS/Legacy BIOS boot line for /etc/fstab

```
# Adjust for any formatting differences and/or additional partitions created from the "Preparing the
disks" step
/dev/sda1  /boot      xfs      defaults      0 2
```

Some system administrators want the **/boot** partition to not be mounted automatically to improve their system's security. Those people should substitute the **defaults** with **noauto**. This does mean that those users will need to manually mount this partition every time they want to use it.

Add the rules that match the previously decided partitioning scheme and append rules for devices such as CD-ROM drive(s), and of course, if other partitions or drives are used, for those too.

Below is a more elaborate example of an **/etc/fstab** file:

### FILE /etc/fstab A full /etc/fstab example for a DOS/Legacy BIOS system

```
# Adjust for any formatting differences and/or additional partitions created from the "Preparing the
disks" step
/dev/sda1  /boot      xfs      defaults      0 2
/dev/sda2  none       swap     sw            0 0
/dev/sda3  /          xfs      defaults,noatime  0 1
/dev/cdrom /mnt/cdrom auto      noauto,user    0 0
```

## UEFI systems

Below is an example of an **/etc/fstab** file for a system that will boot via UEFI firmware:

### FILE /etc/fstab A full /etc/fstab example for an UEFI system

```
# Adjust for any formatting differences and/or additional partitions created from the "Preparing the
disks" step
/dev/sda1  /efi       vfat     umask=0077    0 2
/dev/sda2  none       swap     sw            0 0
/dev/sda3  /          xfs      defaults,noatime  0 1
/dev/cdrom /mnt/cdrom auto      noauto,user    0 0
```

## DPS UEFI PARTUUID

Below is an example of an **/etc/fstab** file for a disk formatted with a GPT disklabel and Discoverable Partition Specification (DPS) UUIDs set for UEFI firmware:

### FILE /etc/fstab DPS PARTUUID fstab example

```
# Adjust any formatting difference and additional partitions created from the "Preparing the disks"
step.
# This example shows a GPT disklabel with Discoverable Partition Specification (DSP) UUID set:
PARTUUID=c12a7328-f81f-11d2-ba4b-00a0c93ec93b /efi       vfat     umask=0077    0 2
PARTUUID=0657fd6d-a4ab-43c4-84e5-0933c84b4f4f none       swap     sw            0 0
PARTUUID=4f68bce3-e8cd-4db1-96e7-fbcaf984b709 /          xfs      defaults,noatime  0 1
```

When **auto** is used in the third field, it makes the **mount** command guess what the filesystem would be. This is recommended for removable media as they can be created with one of many filesystems. The **user** option in the fourth field makes it possible for non-root users to mount the CD.

To improve performance, most users would want to add the **noatime** mount option, which results in a faster system since access times are not registered (those are not needed generally anyway). This is also recommended for systems with solid state drives (SSDs). Users may wish to consider **lazytime** instead.



Tip

Due to degradation in performance, defining the `discard` mount option in `/etc/fstab` is not recommended. It is generally better to schedule block discards on a periodic basis using a job scheduler such as **cron** or a timer (systemd). See [Periodic fstrim jobs \(/wiki/SSD#Periodic\\_fstrim\\_jobs\)](#) for more information.

Double-check the `/etc/fstab` file, then save and quit to continue.

## Networking information

It is important to note the following sections are provided to help the reader quickly setup their system to partake in a local area network.

For systems running OpenRC, a more detailed reference for network setup is available in the advanced network configuration ([/wiki/Handbook:AMD64/Networking/Introduction](#)) section, which is covered near the end of the handbook. Systems with more specific network needs may need to skip ahead, then return here to continue with the rest of the installation.

For more specific systemd network setup, please review see the networking portion ([/wiki/Systemd#Network](#)) of the `systemd` ([/wiki/Systemd](#)) article.

## Hostname

One of the choices the system administrator has to make is name their PC. This seems to be quite easy, but lots of users are having difficulties finding the appropriate name for the hostname. To speed things up, know that the decision is not final - it can be changed afterwards. In the examples below, the hostname `tux` is used.

### Set the hostname (OpenRC or systemd)

```
root # echo tux > /etc/hostname
```

#### systemd

To set the system hostname for a system currently *running* systemd, the **hostnamectl** utility may be used. During the installation process however, `systemd-firstboot` ([/wiki/Handbook:AMD64/Installation/System#Init\\_and\\_boot\\_configuration\\_systemd](#)) command must be used instead (see later on in handbook).

For setting the hostname to "tux", one would run:

```
root # hostnamectl hostname tux
```

View help by running **hostnamectl --help** or **man 1 hostnamectl**.

## Network

There are *many* options available for configuring network interfaces. This section covers a only a few methods. Choose the one which seems best suited to the setup needed.

### DHCP via dhcpcd (any init system)

Most LAN networks operate a DHCP server. If this is the case, then using the `dhcpcd` program to obtain an IP address is recommended.

To install:

```
root # emerge --ask net-misc/dhcpcd
```

To enable and then start the service on OpenRC systems:

```
root # rc-update add dhcpcd default
```

```
root # rc-service dhcpcd start
```

To enable the service on systemd systems:

```
root # systemctl enable dhcpcd
```

With these steps completed, next time the system boots, `dhcpcd` should obtain an IP address from the DHCP server. See the `Dhcpcd` ([/wiki/Dhcpcd](#)) article for more details.

### netifrc (OpenRC)



#### Tip

This is one particular way of setting up the network using `Netifrc` ([/wiki/Netifrc](#)) on OpenRC. Other methods exist for simpler setups like `Dhcpcd` ([/wiki/Dhcpcd](#)).

### Configuring the network

During the Gentoo Linux installation, networking was already configured. However, that was for the live environment itself and not for the installed environment. Right now, the network configuration is made for the installed Gentoo Linux system.



#### Note

More detailed information about networking, including advanced topics like bonding, bridging, 802.1Q VLANs or wireless networking is covered in the advanced network configuration (</wiki/Handbook:AMD64/Networking/Introduction>) section.

All networking information is gathered in **/etc/conf.d/net**. It uses a straightforward - yet perhaps not intuitive - syntax. Do not fear! Everything is explained below. A fully commented example that covers many different configurations is available in **/usr/share/doc/netifrc-\*/net.example.bz2**.

First install `net-misc/netifrc` (<https://packages.gentoo.org/packages/net-misc/netifrc>):

```
root # emerge --ask --noreplace net-misc/netifrc
```

DHCP is used by default. For DHCP to work, a DHCP client needs to be installed. This is described later in Installing Necessary System Tools.

If the network connection needs to be configured because of specific DHCP options or because DHCP is not used at all, then open **/etc/conf.d/net**:

```
root # nano /etc/conf.d/net
```

Set both `config_eth0` and `routes_eth0` to enter IP address information and routing information:

#### Note

This assumes that the network interface will be called `eth0`. This is, however, very system dependent. It is recommended to assume that the interface is named the same as the interface name when booted from the installation media if the installation media is sufficiently recent. More information can be found in the Network interface naming ([/wiki/Handbook:AMD64/Networking/Advanced#Network\\_interface\\_naming](/wiki/Handbook:AMD64/Networking/Advanced#Network_interface_naming)) section.

#### **/etc/conf.d/net** Static IP definition

```
config_eth0="192.168.0.2 netmask 255.255.255.0 brd 192.168.0.255"
routes_eth0="default via 192.168.0.1"
```

To use DHCP, define `config_eth0`:

#### **/etc/conf.d/net** DHCP definition

```
config_eth0="dhcp"
```

Please read **/usr/share/doc/netifrc-\*/net.example.bz2** for a list of additional configuration options. Be sure to also read up on the DHCP client man page if specific DHCP options need to be set.

If the system has several network interfaces, then repeat the above steps for `config_eth1`, `config_eth2`, etc.

Now save the configuration and exit to continue.

Automatically start networking at boot

To have the network interfaces activated at boot, they need to be added to the default runlevel.

```
root # cd /etc/init.d
```

```
root # ln -s net.lo net.eth0
```

```
root # rc-update add net.eth0 default
```

If the system has several network interfaces, then the appropriate **net.\*** files need to be created just like we did with **net.eth0**.

If, after booting the system, it is discovered the network interface name (which is currently documented as `eth0`) was wrong, then execute the following steps to rectify:

1. Update the **/etc/conf.d/net** file with the correct interface name (like `enp3s0` or `enp5s0`, instead of `eth0`).
2. Create new symbolic link (like **/etc/init.d/net.enp3s0**).
3. Remove the old symbolic link (**rm /etc/init.d/net.eth0**).
4. Add the new one to the default runlevel.
5. Remove the old one using **rc-update del net.eth0 default**.

## The hosts file

An important next step may be to inform this new system about other hosts in its network environment. Network host names can be defined in the **/etc/hosts** file. Adding host names here will enable host name to IP addresses resolution for hosts that are not resolved by the nameserver.

```
root # nano /etc/hosts
```

#### **/etc/hosts** Filling in the networking information

```
# This defines the current system and must be set
127.0.0.1    tux.homenetwork tux localhost

# Optional definition of extra systems on the network
192.168.0.5  jenny.homenetwork jenny
192.168.0.6  benny.homenetwork benny
```

Save and exit the editor to continue.

## System information

### Root password

Set the root password using the **passwd** command.

```
root # passwd
```

Later an additional regular user account will be created for daily operations.

### Init and boot configuration

#### OpenRC

When using OpenRC with Gentoo, it uses **/etc/rc.conf** to configure the services, startup, and shutdown of a system. Open up **/etc/rc.conf** and enjoy all the comments in the file. Review the settings and change where needed.

```
root # nano /etc/rc.conf
```

Next, open **/etc/conf.d/keymaps** to handle keyboard configuration. Edit it to configure and select the right keyboard.

```
root # nano /etc/conf.d/keymaps
```

Take special care with the *keymap* variable. If the wrong keymap is selected, then weird results will come up when typing on the keyboard.

Finally, edit **/etc/conf.d/hwclock** to set the clock options. Edit it according to personal preference.

```
root # nano /etc/conf.d/hwclock
```

If the hardware clock is not using UTC, then it is necessary to set `clock="local"` in the file. Otherwise the system might show clock skew behavior.

#### systemd

First, it is recommended to run **systemd-machine-id-setup** and then **systemd-firstboot** which will prepare various components of the system are set correctly for the first boot into the new systemd environment. The passing the following options will include a prompt for the user to set a locale, timezone, hostname, root password, and root shell values. It will also assign a random machine ID to the installation:

```
root # systemd-machine-id-setup
```

```
root # systemd-firstboot --prompt
```

Next users should run **systemctl** to reset all installed unit files to the preset policy values:

```
root # systemctl preset-all --preset-mode=enable-only
```

It's possible to run the full preset changes but this may reset any services which were already configured during the process:

```
root # systemctl preset-all
```

These two steps will help ensure a smooth transition from the live environment to the installation's first boot.

← [Configuring the kernel \(/wiki/Handbook:AMD64/Installation/Kernel\)](#) [Installing tools \(/wiki/Handbook:AMD64/Installation/Tools\)](#) →

Retrieved from "<https://wiki.gentoo.org/index.php?title=Handbook:AMD64/Installation/System&oldid=212422>  
(<https://wiki.gentoo.org/index.php?title=Handbook:AMD64/Installation/System&oldid=212422>)"

- This page was last edited on 2 January 2015, at 00:03.
- [Privacy policy \(/wiki/Gentoo\\_Wiki:Privacy\\_policy\)](#)
- [About Gentoo Wiki \(/wiki/Gentoo\\_Wiki:About\)](#)
- [Disclaimers \(/wiki/Gentoo\\_Wiki:General\\_disclaimer\)](#)

**© 2001–2024 Gentoo Authors**

Gentoo is a trademark of the Gentoo Foundation, Inc. and of Förderverein Gentoo e.V. The contents of this document, unless otherwise expressly stated, are licensed under the CC-BY-SA-4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>) license. The [Gentoo Name and Logo Usage Guidelines](https://www.gentoo.org/inside-gentoo/foundation/name-logo-guidelines.html) (<https://www.gentoo.org/inside-gentoo/foundation/name-logo-guidelines.html>) apply.

---