Gentoo Handbook

Gentoo

Why should I pick Gentoo over (insert favorite source distribution here)?

Gentoo has a strong support community, good documentation and for most people very few problems. Its package management system allows safe installation (merging) of new versions - the old one is kept until removed (cleaned). This solves many problems you'd otherwise encounter when installing from source and is a unique feature available only in Gentoo. Also, it's easy to globally enable support for something (e.g. mysql) - edit a single file and issue a single command, and everything that supports mysql will have been recompiled with support for it. Another Gentoo only feature. We think it's better, your mileage may vary, batteries not included.

About the Gentoo Linux Installation

This chapter introduces the installation approach documented in this handbook.

Introduction

Welcome

First of all, welcome to Gentoo! You are about to enter the world of choices and performance. Gentoo is all about choices. When installing Gentoo, this is made clear several times - users can choose how much they want to compile themselves, how to install Gentoo, what system logger to use, etc.

Gentoo is fast, modern meta-distribution with a clean and flexible design. It is built on an ecosystem of free software and does not hide what is beneath the hood from its users.

Portage, the package maintenance system wich Gentoo uses, is written in Python, meaning the user can easily view and modify the source code.

Gentoo's packaging system uses source code (altrough support for pre-compiled packages is included too) and configuring Gentoo happens through regular text files. In other words, openness everywhere.

It is very important that everyone understands that choices are what makes Gentoo run. We try not to force users into anything they do not like. If anyone belives otherwise, please bug report it.

How the installation is structured

The Gentoo installation can be seen as a 10-step procedure, corresponding to the next set of chapters. Each step results in a certain state:

[1] The user is in a working environment ready to install Gentoo.

[2] The Internet connection is ready to install Gentoo.

[3] The hard disk are initializad to host the Gentoo installation.

[4] The installation environment is prepared and the user is ready to chroot into the new environment.

[5] Core packages, which are the same on all Gentoo installations, are installed.

[6] The Linux kernel is installed.

[7] The user will have configured most of the Gentoo system configuration files.

[8] The necessary system tools are installed.

[9] The proper boot loader has benn installed an configured.

[10] The freshly installed gentoo Linux enviroment is ready to be explored.

Whenever a certain choice is presented the handbook will try to explain the pros and cons of each choice. Although the next then continues with a default choice(identified by "Default: "in the title). Do not think that the default is what Gentoo recommends. *It is however what Gentoo belives most users will use.*

Sometimes an optional step can be followed. Such steps are marked as "Optional: " and are therefore not needed to install Gentoo. However, some optional steps are dependent on a previously made a decision is made, and right before the optional step is described.

Installation options for Gentoo

Gentoo can be installed in many different ways. It can be downloaded and installed from official Gentoo installation media such as our CDs and DVDs. The installation media can be installed on USB stick or accessed via netbooted environment. Alternatively, Gentoo can be installed from non-official media such as an alredy installed distribution or a non-Gentoo bootable disk (such as Knoppix).

This document covers the installation using official Gentoo Installation media or, in certain cases, netbooting.

Note: *For help on the other installation approaches, including using non-Gentoo CDs, please read our Alternative installation guide.*

We also provide a Gentoo Installation tips and tricks document that might be useful to read as well.

Troubles

If a problem is found in the installation (or in the installation documentation), please visit our bug tracking system and check if the bug is known. If not, please create a bug report for it so we can take care of it. Do not be afraid of the developers who are assigned to the bugs - they (generally) don’t eat people.

Note though that, although this document is architecture-specific, it might contain references to other architectures as well. This is due to the fact that large parts of the Gentoo Handbook use installation source text that is shared for all architectures (to avoid duplication of efforts and starvation of development resources). We will try to keep this to a minimum to avoid confusion.

Speaking of wich, if there are any additional questions regarding Gentoo, chech out the Frequently Asked Questions article. There are also FAQs on the Gentoo Forums.

1. Choosing the right installation medium

It is possible to install Gentoo in many ways. This chapter explains how to install Gentoo using the minimal installation CD.

Hardware requirements

Before we start, we first list what hardware requirements are needed to sucessfully install Gentoo on an amd64 box.

AMD64 livedisk hardware requirements:

```
CPU:         Any AMD64 CPU or EM64T CPU (Core i3, i5, andi7 are EM64T)

Memory:      256MB

Disk space:      2.5GB (excluding swap space)

Swap space:      At least 256MB
```

The AMD64 project page is a good place to be for more information about Gentoo's amd64 support.

Gentoo Linux installation media.

Minimal installation CD

The Gentoo minimal installation CD is a bootable image which contains a self-sustainded Gentoo environment. It allows the user to boot Linux from the CD or other installation media. During the boot process the hardware is detected and the appropriate drivers are loaded.

The image is maintained by Gentoo developers and allows anyone to install Gentoo if an active Internet connnection is available.

The Minimal Installation CD is called **install-amd64-minimal-<release>.iso**

The occasional Gentoo LiveDVD

Occasionally, a special DVD is crafted by the Gentoo Ten project which can be used to install Gentoo. The instructions further down this chapter target the Minimal Installation CD so might be a bit different. However, the LiveDVD (or any other bootable Linux environment) supports getting a root prompt by just invoking sudo su - or sudo -i in a terminal.

What are stages then?

A stage3 tarball is an archive containing a profile specific minimal Gentoo environment. Stage3 tarball are suitable to continue the Gentoo installation using the instructions in this hand book. Previously, the handbook described the installation using one of three stage tarballs. While Gentoo still offers stage1 and stage2 tarballs, this oficial installation method uses the stage3 tarball. Those interested in performing a Gentoo installation using a stage less than three should read the Gentoo FAQ on **[How do in install Gentoo using a stage1 or stage2 tarball]**

Stage3 tarballs can be downloaded from **releases/amd64/autobuilds/** on any of the official Gentoo mirrors. Stage files update frequently and are not included in official installation images

Downloading

Obtain the media

The default installation media that Gentoo Linux uses are the *minimal installation* CDs, wich host a bootable, very small Gentoo Linux environment. This environment contains all the right tools to install Gentoo. The CD images themselves can be downloaded from the downloads page (recommended) or by manually browsing to the ISO location on one of the many available mirrors.

If downloading from a mirror, the minimal installation CDs can be found as follows:

Go to the `releases/` directory.

Select the directory for the relevant target architecture (such as amd64/).

Select the `autobuilds/` directory.

For amd64 and x86 architectures select either the `current-install-amd64-minimal/` or `current-install-x86-minimal/` directory (respectively). For all other architectures navigate to the `current-iso/` directory.

Note: *Some target architecture such as arm, mips, and s390 will not have minimal install CDs. At this time the Gentoo Release Engineering project does not support building .iso files for these targets.*

Inside this location, the installation media file is the file with the .iso suffix. For instance, take a look at the following listing:

[CODE] Example list of downloadable files at releases/amd64/autobuilds/current-iso/

```
[DIR]   hardened/                                          05-Dec-2014
01:42    -

[    ]  install-amd64-minimal-20141204.iso                 04-Dec-2014
21:04  208M

[    ]  install-amd64-minimal-20141204.iso.CONTENTS        04-Dec-2014
21:04  3.0K

[    ]  install-amd64-minimal-20141204.iso.DIGESTS         04-Dec-2014
21:04  740

[TXT]   install-amd64-minimal-20141204.iso.DIGESTS.asc     05-Dec-2014
01:42  1.6K

[    ]  stage3-amd64-20141204.tar.bz2                      04-Dec-2014
21:04  198M

[    ]  stage3-amd64-20141204.tar.bz2.CONTENTS             04-Dec-2014
21:04  4.6M

[    ]  stage3-amd64-20141204.tar.bz2.DIGESTS              04-Dec-2014
21:04  720

[TXT]   stage3-amd64-20141204.tar.bz2.DIGESTS.asc          05-Dec-2014
01:42  1.5K
```

In the above example, the `install-amd64-minimal-20141204.iso` file is the minimal installation CD itself. But as can be seen, other related files exist as well:

- `.CONTENTS` file wich is a text file listing all files available on the installation media. This file can be useful to verify if particular firmware or drivers are available on the installation media before downloading it.
- `.DIGESTS` file which contains the hash of the ISO file itself, in various hashing formats/algorithms. This file can be used to verify if the downloaded ISO file is corrupt or not.
- `.DIGESTS.asc` file which not only contains the hash of the ISO file (like the `.DIGESTS` file), but also cryptographic sigature of that file. This can be used to both verify if the downloaded ISO file is corrupt or not, as well as verify that the downloaded is indeed provided by the Gentoo Release Engineering team and has not been tampered with.

Ignore the other files available at this location for now - those will come back when the installation has proceeded further. Download the `.iso` file and, if verification of the download if wanted, download the `.DIGESTS.asc` file for the `.iso` file as well. The `.CONTENTS` file does not need to be downloaded as the installation instructions will not refer to this file anymore, and the `.DIGESTS` file should contain the same information as the `.DIGESTS.asc` file, except that the latter also contains a signature on top of it.

Verifying the downloaded files

Note: *This is an optional step and not necessary to install Gentoo Linux. However, it is recommended as it ensures that the downloaded file is not corrupt and has indeed been provided by the Gentoo insfrastructure team.*

Through the `DIGESTS` and `DIGESTS.asc` files, the validity of the ISO can be confirmed using the right set of tools. This verification is usually done in two steps:

First, the cryptographic signature is validated to make sure that the installation file is provided by the Gentoo Release Engineering team.

If the cryptographic signature validates, then the checksum is verified to make sure that the downloaded file itself is not corrupted.

Microsoft Windows based verification

On a Microsoft Windows system, chances are low that the right set of tools to verify checksums and cryptographic signatures are in place.

To verify the cryptographic signature, tools such as the **GPG4Win programm** can be used. After installation, the public keys of the Gentoo Release Engineering team need to be imported. The list of keys is available on the signatures page. Once imported, the user can then verify the signature of the `DIGESTS.asc` file.

Important: *This does not verify the .DIGESTS file is correct, only that the .DIGESTS.asc file is. That also implies that the checksum should be verified against the values in the .DIGESTS.asc file, which is why the instructions above only refer to downloading the .DIGESTS.asc file.*

The checksum itself can be verified using the Hashcalc applicaton, although many others exist as well. Most of the time, these tools will show the user the calculated checksum, and the user is requested to verify this checksum with the value that is inside the `DIGESTS.asc` file.

Linux based verification

On a Linux system, the most common method for verifying the cryptographic signature is to use the "app-crypt/gnupg" software. With this package installed, following commands can be used to verify the cryptographic signature of the `DIGESTS.asc` file.

```
user $gpg --keyserver hkps://hkps.pool.sks-keyservers.net --recv-keys 0xBB572E0E2D182910

gpg: requesting key 0xBB572E0E2D182910 from hkp server pool.sks-key-servers.net

newgpg: key 0xBB572E0E2D182910: Gentoo Linux Release Engineering (Auto-mated Weekly Release Key) <releng@gentoo.org>1

gpg: 3 marginal(s) needed, 1 complete(s) needed, classic trust model

gpg: depth: 0  valid:   3  signed:  20  trust: 0-, 0q, 0n, 0m, 0f, 3u

gpg: depth: 1  valid:  20  signed:  12  trust: 9-, 0q, 0n, 9m, 2f, 0u

gpg: next trustdb check due at 2018-09-15

gpg: Total number processed: 1

gpg:         new signatures: 1
```

Alternatively you can use instead the KWD to download the key:

```
user    $wget    -O-    https://gentoo.org/.well-known/openpgpkey/hu/wtk-tzo4gyuhzu8a4z5fdj3fgmr1u6tob?l=releng | gpg --import

--2019-04-19     20:46:32--      https://gentoo.org/.well-known/openpgp-key/hu/wtktzo4gyuhzu8a4z5fdj3fgmr1u6tob?l=releng

Resolving gentoo.org (gentoo.org)... 89.16.167.134

Connecting to gentoo.org (gentoo.org)|89.16.167.134|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 35444 (35K) [application/octet-stream]

Saving to: ’STDOUT’

     0K  .........  .........  .........  ....                  100% 11.9M=0.003s

2019-04-19 20:46:32 (11.9 MB/s) - written to stdout [35444/35444]

gpg: key 9E6438C817072058: 84 signatures not checked due to missing keys

gpg: /tmp/test2/trustdb.gpg: trustdb created

gpg: key 9E6438C817072058: public key Gentoo Linux Release Engineering (Gentoo Linux Release Signing Key) <releng@gentoo.org>imported"

gpg: key BB572E0E2D182910: 12 signatures not checked due to missing keys

gpg: key BB572E0E2D182910: 1 bad signature
```

```
gpg: key BB572E0E2D182910: public key Gentoo Linux Release Engineering
(Automated Weekly Release Key) <releng@gentoo.org>imported"

gpg: Total number processed: 2

gpg:               imported: 2

gpg: no ultimately trusted keys found
```

Next verify the cryptogrsphic signature of the `DIGESTS.asc` file:

```
user $gpg --verify install-amd64-minimal-20141204.iso.DIGESTS.asc

gpg: Signature made Fri 05 Dec 2014 02:42:44 AM CET

gpg:                using RSA key 0xBB572E0E2D182910

gpg: Good signature from Gentoo Linux Release Engineering (Automated
Weekly Release Key) <releng@gentoo.org> [unknown]

gpg: WARNING: This key is not certified with a trusted signature!

gpg:          There is no indication that the signature belongs to the
owner.

Primary key fingerprint: 13EB BDBE DE7A 1277 5DFD  B1BA BB57 2E0E 2D18
2910
```

To be absolutely ceratain that everything is valid, verify the fingerprint shown with the finegerprint on the **Gentoo signatures page.**

With the cryptographic signature validated, next verify the checksum to make sure the downloaded ISO file is not corrupted. The `.DIGESTS.asc` file contains multiple hashing algorithms, so one of the methods to validate the right one is to first look at the checksum registered in the `DIGESTS.asc` file. For instance, to get the SHA512 checksum:

```
user $grep -A 1 -i sha512 install-amd64-minimal-20141204.iso.DIGESTS.asc

# SHA512 HASH

364d32c4f  8420605f8a  9fa3a0fc55  864d5b0d1a  f11aa62b7a  4d4699a427
e5144b2d918   225dfb7c5de   c8d3f0fe2cdd   b7cc306da6f0   cef4f01abec3
3eec74f3024  install-amd64-minimal-20141204.iso

--

# SHA512 HASH

0719a8954d c7432750de 2e3076c8b8 43a2c79f5e6 0defe43fcca8c 32ab26681dfb
9898b102e2 11174a895ff 4c8c41ddd9e9 a00ad6434d3 6c68d74bd02 f19b57f  in-
stall-amd64-minimal-20141204.iso.CONTENTS
```

On the above output, two SHA512 checksums are shown - one for the `install-amd64-minimal-20141204.iso` file and one for its accompanying `.CONTENTS` file. Only the first checksum is of interest, as it needs to be compared with the calculated SHA512 checksum which can be generated as follows: `user $sha512sum install-amd64-minimal-20141204.iso 364d32c4f8 420605f8a 9fa3a0fc55 864d5b0d1a f11aa62b7a4 d4699a427e 5144b2d91822`

```
5dfb7c5dec8d 3f0fe2cddb7 cc306da6f0cef 4f01abec33 eec74f3024  install-
amd64-minimal-20141204.iso
```

As both checksums match, the file is not corrupted and the installation can continue.

Burning a disk

Of course, with just an ISO file downloaded, the Gentoo Linux installation cannot be started. The ISO file needs to be burned on a CD to boot from, and is such a way that it's *content* is burned on the CD, not just the file itself. Below a few common methods are described - a more elaborate set of intructions can be found in our FAQ on burning an ISO file.

Burning with Linux

On Linux, the ISO file can be burned on a CD using the **cdrecord** utility, which is included in the "app-cdr/cdrtools" package.

For instance, to burn the ISO file on the CD in the `/dev/sr0` device (this is the first CD device on the system - substitute with the right device file if neccesary):

```
user $cdrecord dev=/dev/sr0 install-amd64-minimal-20141204.iso
```

Users that prefer a graphical user interface can use K3B, part of the "kde-apps/k3b" package. In K3B, go to `Tools` and use `Burn CD Image.` Then follow the instructions provided by K3B.

Booting

Booting the installation media

Once the installation media is ready, it is time to boot it. Insert the media in the system, reboot, and enter the motherboard's firmware user interface. This is usually performed by pressing a keyboard key such as `[DEL], [F1] [F10],` or `[ESC]` during the Power-On Self-test (POST) process.

The 'trigger' key varies depending on the system and motherboard. If it is not obvious use an internet search engine and do some research using the motherboard's model name as the search keyword. Results should be easy to determine. Once inside the motherboard's firmware menu, changr the boot order so that the external bootable media (CD/DVD disks or USB drives) are tried *before* the internal disk devices. Without this change, the system will most likely reboot to the terminal disk device, ignoring the externall bot media. **Important** *When installing Gentoo with the purpose of using the UEFI interface instead of BIOS, it is recommended to boot with UEFI immediateky. If not, then it might be neccessary to create a bootable UEFI USB stick (or other medium) once before finalizing the Gentoo Linux Installation.*

IF not yet done, ensure that the installation media is inserted or plugged into the system, and reboot. A boot prompt should be shown. At this screen, `[Enter]` will begin the boot process with the default boot options, To boot the installation media with custom boot options, specify a kernel followed by boot options and then hit `[Enter].` At the boot prompt, users get the option of displaying the available kernels (`[F1]`) and boot options (`[F2]`). If no choice is made within 15 seconds (either displaying information or using a kernel) then the installation media will fall back to booting from disk. This allows installations to reboot and try out their installed enviroment without the need to remove the CD from the tray (something well appreciated for remote installations).

Specifying a kernel was mentioned. On the Minimal installation media, only two predefined kernel boot options are provided. The default option is called **gentoo.** The other being the *-nofb* variant; this disables kernel framebuffer support.

The next section displays a short overview of the available kernels and their descriptions:

Kernel choices

gentoo

Default kernel with support for K8 CPUs (including NUMA support) and EM64T CPUs.

gentoo-nofb

Same as *gentoo* but without framebuffer support

IMPORTANT *The framebuffer's original function is as a video RAM cache to allow more flexibility to (older) video cards. Many newer cards come with framebuffers on board, which are often already compatible with many operating systems. Enabling framebuffer support in the Linux kernel will often cause graphical artifacts or black screen displays. For most newer cards, this option should not be selected when using the LiveDVD.*

memtest86

Test the local RAM for errors

Alongside the kernell, boot options help in turning the boot process further.

Hardware options

acpi=on

This loads support for ACPI and also causes the acpid daemon to be started by the CD on boot. This is only needed if the system requires ACPI to function properly. This is not required for Hyperthreading support.

acpi=off (Not recommended) (See also "doapm")

Completely disables ACPI. This useful on some older systems and is also a requirement for using APM. This will disable any Hyperthreading support of your processor.

console=X

This sets up serial console access for the CD. The first option is the device, usually ttyS0 on x86, followed by any connection options, which are comma separated. The default options are 9600,n,8,1.

dmraid=X

This allows for passing options to the device-mapper RAID subsystem. Options should be encapsulated in quotes.

doapm (old machines)

This loads APM driver support. This also requires that `"acpi=off"`

dopcmcia

This loads support for most SCSI controllers. This is also a requirement for booting most USB devices, as they use the SCSI subsystem of the kernel.

Note *The Personal Computer Memory Card International Association (PCMCIA) was a group of computer hardware manufacturers, operating under that name from 1989 to 2009/2010. Starting with the eponymous PCMCIA card in 1990, it created various standards for peripheral interfaces designed for laptop computers. The PCMCIA association was dissolved in 2009 and all of its activities have since been managed by the USB Implementer's Forum, according to the PCMCIA website.*

doscsi

This loads support for most SCSI controllers. This is also a requirement for booting most USB devices, as they use the SCSI subsystem of the kernel.

Note *The ancestral SCSI standard, X3.131-1986, generally referred to as SCSI-1, was published by the X3T9 technical committee of the American National Standards Institute (ANSI) in 1986. SCSI-2 was published in August 1990 as X3.T9.2/86-109, with further revisions in 1994 and subsequent adoption of a multitude of interfaces. Further refinements have resulted in improvements in performance and support for ever-increasing storage data capacity*

sda=stroke

This allows the user to partition the whole hard disk even when the BIOS is unable to handle large disks. This option is only used on machines with an older BIOS. Replace sda with the device that requires this option.

ide=nodma

This forces the disabling of DMA in the kernel and is required by some IDE chipsets and also by some CDROM drives, If the system is having trouble reading from IDE CDROM, try this option. This also disables the default hdparm settings from being executed.

noapic

This disables the Advanced Programmable Interrupt Controller that is present on newer motherboards. It has been known to cause some problems on older hardware.

nodetect

This disables all of the autodetection done by the CD, including device autodetection and DHCP probing. This is useful for doing debugging of failing CD or driver.

nodhcp

This disables DHCP probing on detected network cards. This is useful on networks with only static addresses.

nodmraid

Disables support for device-mapper RAID, such as that used for on-board IDE/SATA RAID controllers.

nofirewire

This disables the loading of Firewire modules. This sould only be necessary if your Firewire hardware is causing a problem with booting the CD.

Note: *What exactly is FireWire? It is a serial bus similar in principle to USB, but runs at speeds of up to 800 Mbit/s and is not centered around a PC (i.e. there may be none or multiple PCs on the same bus). It has a mode of transmission which guarantees bandwidth which makes it ideal for digital video cameras and similar devices.*

nogpm

This disables gpm console mouse support

nohotplug

This disables the loading of the hotplug and coldplug init scripts at boot. This is useful for doing debugging of a failing CD or driver.

nokeymap

This disables the keymap selection used to select non-US keyboard layouts.

nolapic

This disables the local APIC on Uniprocessor kernels.

nosata

This disables the loading of Serial ATA modules. This is used if the system is having problems with the SATA subsystem.

nosmp

This disables SMP, or Symmetric Multiprocessing, on SMP-enabled kernels. This is useful for debugging SMP-related issues with certain drivers and motherboards.

nosound

This disables sound support and volume setting. This is useful for systems where sound support causes problems.

nousb

This disables the autoloading of USB modules. This is useful for debugging USB issues.

slowusb

his adds some extra pauses into the boot process for slow USB CDROMs, like in the IBM BladeCenter.

Logical volume/device management

dolvm

This enables support for Linux's LVM.

Other options

debug

Enables debugging code. Thismight get messy, as it displays a lot of data to the screen.

docache

This caches the entire runtime portion of the CD into RAM, which allows the user to `/mnt/cdrom` and mount another CDROM. This option requires that there is at least twice as much available RAM as the size of the CD.

doload=X

This causes the initial ramdisk to load any module listed, as well as depencies. Replace X with the module name. Multiple modules can be specified by comma-separated list.

dosshd

Starts sshd on boot, which is useful for unattended installs.

passwd=foo

Sets whatever follows the equals as the root password, which is required for *dosshd* since the root password is by default scrambled.

noload=X

This causes the initial ramdisk to skip the loading of a specific module that may be causing a problem. Syntax matches that of doload.

nonfs

Disables the starting of portmap/nfsmount on boot

nox

This causes an X-enabled Lice CD to not automatically start X, but rather, to drop to the command line instead.

scandelay

This causes the CD to pause for 10 seconds during certain portions the boot process to allow for devices that are slow to initialize to be ready for use.

scandelay=X

This allow the user to specify a given delay, in seconds, to be added to certain portions of the boot process to allow foe devices that are slow to initialize to be ready for use. Replace X with the number of seconds to pause.

Note: *The bootable media check for* **no*** *options before* **do*** *options, so that options can be overrodden in the exact order specified.*

Now boot the media, select a kernel (if the default **gentoo** kernel does not suffice) and boot options. As an example, we boot **gentoo** kernel, with `dopcmia` as a kernel parameter:

```
boot: gentoo dopcmcia
```

`[Alt]` + `[F1]` to switch to verbose mode and follow the prompt.

If no selection is made in 10 seconds the default (US keryboard) will be accepted and boot process will continue. Once the boot process completes, the user is automatically logged in to the "Live" Gentoo Linux environment as the *root* user, the super user. A root prompt is displayed on the current console, and one can switch to other consoles by pressing `[Alt]` + `[F2]` , `[Alt]` + `[F3]` and `[Alt]` + `[F4].`

Get back to the one started on by pressing `[Alt]` + `[F1]`

Extra hardware configuration

When the Installation medium boots, it tries to detect all hardware devices and loads the appropriate kernel modules to support the hardware. In the vast majority of cases, it does a very good job. However, in some cases it may not auto-load the kernel modules needed by the system. If the PCI auto-detection missed some of the system’s hardware, the appropiate kernel modules have to be loaded manually.

In the next example the 8139too module (which supports certain kinds of network interfaces) is loaded: `root # modprobe 8139too`

Optional: User accounts

If other people need acess to the installation environment, or there is need to run commands as a non-root user on the installation medium (such as to chat using **irssi** without root privileges for security reasons), then an additional user account needs to be created and the root password set to a strong password.

To change the root password, use the **passwd** utility:

```
root # passwd

New password: (Enter the new password)

Re-enter password: (Re-enter the password)
```

To create user account, first enter their credentials, followed by the account’s password. The **useradd** and **passwd** commands are used for these tasks.

In the next example, a user called *john* is created:

```
root # useradd -m -G users john
```

```
passwd john

New password: (Enter john's password)

Re-enter password: (Re-enter john's password)
```

To switch from the (current) *root* user to newly created user account, use the **su** command:

```
root # su - john
```

Optional: Viewing documentation while installing

TTYs

To view the Gentoo handbook during the installation, first create a user account as described above. Then press `[Alt]` + `[F2]` to go a new terminal.

During the installation, the **links** command can be used to browse the Gentoo handbook - of course only from the moment that the internet connection is working.

```
user $links https://wiki.gentoo.org/wiki/Handbook:AMD64
```

To go back to the original, press `[Alt]` + `[F1]`

GNU Screen

The Screen utility is installed by default on official Gentoo installation media. It may be more efficient for the seasoned Linux enthusiast to use **screen** to view installation intructions via split panes rather than the multiple TTY method mentioned above.

Optional: Starting the SSH daemon

To allow other users to access the system during the installation (perhaps to support during an installation, or even do it remotely), a user account needs to be created (as was documented earlier on) and ther SSh daemon needs to be started.

To fire up the SSH daemon on an OpenRC init, execute the following command:

```
root # rc-service sshd start
```

Note: *If users log on the system, they will see a message that the host key for this system needs to be confirment (through what is called a fingerprint). This behavior is typical and can be expected for initial connections to an SSH server, However, later when the system is set up and someone logs on to the newly created system, the SSH client will warn that the host keys has been changed. This is because the user now logs on to - for SSH - a different server (namely the freshly installed Gentoo system rather than the live environment that the installation is currently using). Follow the instructions given on the screen then to replace the host key on the client system.*

To be able to use sshd, the network needs to function properly. Continue with the chapter on Configuring the network.

2. Configuring the network

To be able to download the latest source code, networking will need to be configured.

Automatic network detection

Maybe it just works?

If the system is plugged into an Ethernet network with a DHCP server, it is very likely that the networking configuration has alredy been set up automatically. If so, then the many included network-aware commands on the installation CD such as **ssh, scp, ping, irssi, wget,** and **links,** Among others, will work immediately.

Determine interface names

ifconfig command

If networking has been configured, the **ifconfig** command should list one or more network interfaces (besides `lo`). In the example below `eth0` shows up:

```
root #ifconfig

eth0      Link encap:Ethernet  HWaddr 00:50:BA:8F:61:7A

          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0

          inet6 addr: fe80::50:ba8f:617a/10 Scope:Link

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:1498792 errors:0 dropped:0 overruns:0 frame:0

          TX packets:1284980 errors:0 dropped:0 overruns:0 carrier:0

          collisions:1984 txqueuelen:100

          RX bytes:485691215 (463.1 Mb)  TX bytes:123951388 (118.2 Mb)

          Interrupt:11 Base address:0xe800
```

As a result of the shift towards *predictable network interface names* , the interdace name on the system can be quite different from the old `eth0` naming convention. Recent installation media might show regulatr network interfaces names likes `eno0` , `ens1` , or `enp5s0.` Look for the interface in the **ifconfig** output that has an IP address related to the local network.

Tip: *If no interfaces are displayed when the stanrd ifconfig command is used, try using the same command with the* `-a` *option. This option forces the utility to show al network interfaces detected by the system wheter they be in an up or down state. If* **ifconfig -a** *produces no results then the hardware is faulty or the driver for the interface has not been loaded into the kernel. Both situations reach beyond the scope of this Handbook. Contact the #gentoo irc channel for support.*

ip command

As an alternative to **ifconfig** , the **ip** command can be used o determine interface names.

The following example shows the output of **ip addr** (of another system so the information shown is different from the previous example):

```
root #ip addr

2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000

    link/ether e8:40:f2:ac:25:7a brd ff:ff:ff:ff:ff:ff

    inet 10.0.20.77/22 brd 10.0.23.255 scope global eno1

       valid_lft forever preferred_lft forever

    inet6 fe80::ea40:f2ff:feac:257a/64 scope link

       valid_lft forever preferred_lft forever
```

The output above may be a bit more complicated to read than alternative. The interface name in the adove example directly follows the number; it is `eno1.`

In the remainder of this document, The handbook will assume that the operating network interface is called `eth0`

Optional: Configure any proxies

If the internet is accessed through a proxy, then it is necessary to set up proxy information during the installation. It is very easy to define a proxy: just define a variable which contains the proxy server information.

In most cases, it sufficient to define the variables using the server hostname. As an example, we assume the proxy is called proxy.gentoo.org and the port is 8080.

To set up an HTTP proxy (for HTTP and HTTPS traffic):

```
root #export http_proxy='http://proxy.gentoo.org:8080''
```

To set up an FTP proxy

```
root #export ftp_proxy=''ftp://proxy.gentoo.org:8080''
```

To set up an RSYNC proxy:

```
root #export RSYNC_PROXY=''proxy.gentoo.org:8080''
```

If the proxy requires a username and password, use the following syntax for the variable

[CODE] Adding username/password to the proxy variable

```
http://username:password@proxy.gentoo.org:8080
```

Testing the network

Try pinging your ISP's DNS server (found in `/etc/resolv.conf`) and a web site of choice. This ensures that the network is functioning properly and that the network packets are reaching the ner, DNS name resolution is working correctly, etc.

```
root # ping www.gentoo.org
```

If this all works, then the remainder of this chapter can be skipped to jump right to the next step of the installation instructions [3][Preparing the disks]

Automatic network configuration

If the network doesn't work inmediately, some installation media allow the user to use **net-setup** (for regular or wireless networks), **pppoe-setup** (for ADSL users) or **pptp** (for PPTP users).

If the installation medium does not contain any of these tools, continue with the network*[*Manual

- Regular Ethernet users should continue with [Default: Using net-setup]
- ADSL users should continue with [Alternative: Using PPP]
- PPTP users should continue with [Alernative: Using PPTP]

Deafult: Using net-setup

The simplest way to set up networking if it didn't get configured automatically is to run the **net-setup** script:

```
root # net-setup eth0
```

net-setup will ask some questions about network environment. When all is done, the network connection should work. Test the network connection as stated before. If the test are positive, congratulations! Skip the rest of this section and continue with **[3] [Preparing the disks]**

If the network still doesn't work, continue with [Manual network configuration]

Alternative: Using PPP

Assuming PPPoE is needed to connect the internet, the installation CD (any version) has made things easier by including ppp. Use the provided .B "pppoe-setup" script to configure the connection. During the setup the Ethernet device that is connected to your ADSL modem, the username and password, IPs of the DNS servers and if a basic firewall is needed or not will be asked.

```
root # pppoe-setup
```

```
root # ppoe-start
```

If something goes wrong, double-check that the username and password are correct by looking at `etc/ppp/pap-secrets` or `/etc/ppp/chap-secrets` and make sure to use the right Ethernet device. If the Ethernet device does not exist, the appropriate network modules need to be loaded. IN that case continue with [Manual network configuration] as it it will explain how to load the appropriate network modules there.

If everything worked, continue with **[3][Preparing the disks].**

Alternative: Using PPTP

If PPTP support is needed, use **pptpclient** which is provided by the installationCDs. But first make sure that the configuration is correct. Edit `/etc/ppp/pap-secrets` or `/etc/ppp/chap-secrets` so it contains the correct username/password combination:

```
root # nano -w /etc/pp/chap-secrets
```

Then adjust /etc/ppp/options.pptp if necessary:

```
root # nano -w /etc/pp/options.pptp
```

When all that is done, run **pptp** (along with the options that couldn't be set in options.pptp) to connect the server:

```
root # pptp <server ipv4 address>
```

Now continue with **[3][Preparing the disks].**

Manual network configuration

Loading the appropiate network modules

When the installation CD boots, it tries to detect all the hardware devices and loads the appropriate kernel modules (drivres) to support the hardware. In the vast majority of cases, it does a very good job, However, in some cases it maynot auto-load the kernel modules needed.

If `net-setup` or `ppoe-setup` failed, then it is possible that the network card wasn't found immediately. Thid means users may have to load appropriate kernel modules manually.

To find out what kernel modules are provided for networking, use the **ls** command:

```
root # ls /lib/modules/‘uname -r‘/kernel/drivers/net
```

If a driver is found for the network device, use **modprobe** to load the kernel module. For instance, to load the pcnet32 module:

```
root #ifconfig eth0

eth0      Link encap:Ethernet  HWaddr FE:FD:00:00:00:00

          BROADCAST NOARP MULTICAST  MTU:1500  Metric:1

          RX packets:0 errors:0 dropped:0 overruns:0 frame:0

          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:0

          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

If however the following error is shown, the network card is not detected:

```
root #ifconfig eth0

eth0: error fetching interface information: Device not found
```

The available network interface names on the system can be listed trough the `/sys` file system:

```
root #ls /sys/class/net

dummy0  eth0  lo  sit0  tap0  wlan0
```

In the above example, 6 interfaces are found. The `eth0` one is most likely the (wired) Ethernet adapter whereas `wlan0` is the wireless one.

Assuming that the network card is now detected, retry **net-setup** or **ppoe-setup** again (which sould work now), but for the hardcore people we explain how to configure the network manually as well.

Using DHCP

DHCP (Dynamic Host Configuration Protocol) makes it possible to automatically receive networking information (IP addres, netmask, broadcast address, gateway, nameservers etc.).

This only works if a DHCP server is in network (or if the ISP provider provides a DCHP service). To have a network interface receive this information automatically, use **dhcpcd:**

```
root # dhcpcd eth0
```

Some network administrators require that the hostname and domainname provided by the DHCP server is used by the system. In that case, use:

```
root # dhcpcd -HD eth0
```

If this works (try pinging some Internet server, like Google’s 8.8.8.8 or Cloudflare’s 1.1.1.1), the everything is set and ready to continue. Skip the rest of this section and continue with [Preparing the disks]