



# Installing the Gentoo base 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 Chrooting
  - 1.1 Copy DNS info
  - 1.2 Mounting the necessary filesystems
  - 1.3 Entering the new environment
  - 1.4 Preparing for a bootloader
    - 1.4.1 UEFI systems
    - 1.4.2 DOS/Legacy BIOS systems
- 2 Configuring Portage
  - 2.1 Installing a Gentoo ebuild repository snapshot from the web
  - 2.2 Optional: Selecting mirrors
  - 2.3 Optional: Updating the Gentoo ebuild repository
  - 2.4 Reading news items
  - 2.5 Choosing the right profile
    - 2.5.1 No-multilib
  - 2.6 Optional: Adding a binary package host
    - 2.6.1 Repository configuration
    - 2.6.2 Installing binary packages
  - 2.7 Optional: Configuring the USE variable
    - 2.7.1 CPU\_FLAGS\_\*
    - 2.7.2 VIDEO\_CARDS
  - 2.8 Optional: Configure the ACCEPT\_LICENSE variable
  - 2.9 Optional: Updating the @world set
    - 2.9.1 Removing obsolete packages
- 3 Optional: Using systemd as the system and service manager
- 4 Timezone
- 5 Configure locales
  - 5.1 Locale generation
  - 5.2 Locale selection
- 6 References

## Chrooting

### Copy DNS info

One thing still remains to be done before entering the new environment and that is copying over the DNS information in **/etc/resolv.conf**. This needs to be done to ensure that networking still works even after entering the new environment. **/etc/resolv.conf** contains the name servers for the network.

To copy this information, it is recommended to pass the `--dereference` option to the **cp** command. This ensures that, if **/etc/resolv.conf** is a symbolic link, that the link's target file is copied instead of the symbolic link itself. Otherwise in the new environment the symbolic link would point to a non-existing file (as the link's target is most likely not available inside the new environment).

```
root # cp --dereference /etc/resolv.conf /mnt/gentoo/etc/
```

### Mounting the necessary filesystems



#### Tip

If using Gentoo's install media, this step can be replaced with simply: **arch-chroot /mnt/gentoo**.

In a few moments, the Linux root will be changed towards the new location.

#### 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  
Configuring the kernel (/wiki/Handbook:AMD64/Installation/Kernel)

Configuring the system (/wiki/Handbook:AMD64/Installation/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)

Files and directories (/wiki/Handbook:AMD64/Portage/Files)

The filesystems that need to be made available are:

- **/proc/** is a pseudo-filesystem. It looks like regular files, but is generated on-the-fly by the Linux kernel
- **/sys/** is a pseudo-filesystem, like **/proc/** which it was once meant to replace, and is more structured than **/proc/**
- **/dev/** is a regular file system which contains all device. It is partially managed by the Linux device manager (usually **udev**)
- **/run/** is a temporary file system used for files generated at runtime, such as PID files or locks

The **/proc/** location will be mounted on **/mnt/gentoo/proc/** whereas the others are bind-mounted. The latter means that, for instance, **/mnt/gentoo/sys/** will actually *be* **/sys/** (it is just a second entry point to the same filesystem) whereas **/mnt/gentoo/proc/** is a new mount (instance so to speak) of the filesystem.

```
root # mount --types proc /proc /mnt/gentoo/proc
root # mount --rbind /sys /mnt/gentoo/sys
root # mount --make-rslave /mnt/gentoo/sys
root # mount --rbind /dev /mnt/gentoo/dev
root # mount --make-rslave /mnt/gentoo/dev
root # mount --bind /run /mnt/gentoo/run
root # mount --make-slave /mnt/gentoo/run
```

#### Note

The `--make-rslave` operations are needed for systemd support later in the installation.

#### Warning

When using non-Gentoo installation media, this might not be sufficient. Some distributions make **/dev/shm** a symbolic link to **/run/shm/** which, after the chroot, becomes invalid. Making **/dev/shm/** a proper tmpfs mount up front can fix this:

```
root # test -L /dev/shm && rm /dev/shm && mkdir /dev/shm
root # mount --types tmpfs --options nosuid,nodev,noexec shm /dev/shm
Also ensure that mode 1777 is set:
root # chmod 1777 /dev/shm /run/shm
```

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)
<b>OpenRC network configuration (/wiki/Handbook:AMD64/Full/Networking)</b>
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)

## Entering the new environment

Now that all partitions are initialized and the base environment installed, it is time to enter the new installation environment by chrooting into it. This means that the session will change its root (most top-level location that can be accessed) from the current installation environment (installation CD or other installation medium) to the installation system (namely the initialized partitions). Hence the name, *change root* or *chroot*.

This chrooting is done in three steps:

1. The root location is changed from **/** (on the installation medium) to **/mnt/gentoo/** (on the partitions) using chroot or arch-chroot, if available.
2. Some settings (those in **/etc/profile**) are reloaded in memory using the **source** command
3. The primary prompt is changed to help us remember that this session is inside a chroot environment.

```
root # chroot /mnt/gentoo /bin/bash
root # source /etc/profile
root # export PS1="(chroot) ${PS1}"
```

From this point, all actions performed are immediately on the new Gentoo Linux environment.

#### Tip

If the Gentoo installation is interrupted anywhere after this point, it *should* be possible to 'resume' the installation at this step. There is no need to re-partition the disks again! Simply mount the root partition (/wiki/Handbook:AMD64/Installation/Disks#Mounting\_the\_root\_partition) and run the steps above starting with copying the DNS info (/wiki/Handbook:AMD64/Installation/Base#Copy\_DNS\_info) to re-enter the working environment. This is also useful for fixing bootloader issues. More information can be found in the chroot (/wiki/Chroot) article.

## Preparing for a bootloader

Now that the new environment has been entered, it is necessary to prepare the new environment for the bootloader. It will be important to have the correct partition mounted when it is time to install the bootloader.

### UEFI systems

For UEFI systems, **/dev/sda1** was formatted with the FAT32 filesystem and will be used as the EFI System Partition (ESP). Create a new **/efi** directory (if not yet created), and then mount ESP there:

```
root # mount /dev/sda1 /efi
```

## DOS/Legacy BIOS systems

For DOS/Legacy BIOS systems, the bootloader will be installed into the `/boot` directory, therefore mount as follows:

```
root # mount /dev/sda1 /boot
```

## Configuring Portage

### Installing a Gentoo ebuild repository snapshot from the web

Next step is to install a snapshot of the Gentoo ebuild repository. This snapshot contains a collection of files that informs Portage about available software titles (for installation), which profiles the system administrator can select, package or profile specific news items, etc.

The use of **emerge-webrsync** is recommended for those who are behind restrictive firewalls (it uses HTTP/FTP protocols for downloading the snapshot) and saves network bandwidth. Readers who have no network or bandwidth restrictions can happily skip down to the next section.

This will fetch the latest snapshot (which is released on a daily basis) from one of Gentoo's mirrors and install it onto the system:

```
root # emerge-webrsync
```

#### Note

During this operation, **emerge-webrsync** might complain about a missing `/var/db/repos/gentoo/` location. This is to be expected and nothing to worry about - the tool will create the location.

From this point onward, Portage might mention that certain updates are recommended to be executed. This is because system packages installed through the stage file might have newer versions available; Portage is now aware of new packages because of the repository snapshot. Package updates can be safely ignored for now; updates can be delayed until after the Gentoo installation has finished.

### Optional: Selecting mirrors

In order to download source code quickly it is recommended to select a fast, geographically close mirror. Portage will look in the **make.conf** file for the `GENTOO_MIRRORS` variable and use the mirrors listed therein. It is possible to surf to the Gentoo mirror list and search for a mirror (or multiple mirrors) close to the system's physical location (as those are most frequently the fastest ones).

A tool called **mirrorselect** provides a pretty text interface to more quickly query and select suitable mirrors. Just navigate to the mirrors of choice and press `Spacebar` to select one or more mirrors.

```
root # emerge --ask --verbose --oneshot app-portage/mirrorselect
```

```
root # mirrorselect -i -o >> /etc/portage/make.conf
```

Alternatively, a list of active mirrors are available online (<https://www.gentoo.org/downloads/mirrors/>).

### Optional: Updating the Gentoo ebuild repository

It is possible to update the Gentoo ebuild repository to the latest version. The previous **emerge-webrsync** command will have installed a very recent snapshot (usually recent up to 24h) so this step is definitely optional.

Suppose there is a need for the latest package updates (up to 1 hour), then use **emerge --sync**. This command will use the rsync protocol to update the Gentoo ebuild repository (which was fetched earlier on through **emerge-webrsync**) to the latest state.

```
root # emerge --sync
```

On slow terminals, such as certain frame buffers or serial consoles, it is recommended to use the `--quiet` option to speed up the process:

```
root # emerge --sync --quiet
```

### Reading news items

When the Gentoo ebuild repository is synchronized, Portage may output informational messages similar to the following:

```
* IMPORTANT: 2 news items need reading for repository 'gentoo'.
```

```
* Use eselect news to read news items.
```

News items were created to provide a communication medium to push critical messages to users via the Gentoo ebuild repository. To manage them, use **eselect news**. The **eselect** application is a Gentoo-specific utility that allows for a common management interface for system administration. In this case, **eselect** is asked to use its `news` module.

For the `news` module, three operations are most used:

- With `list` an overview of the available news items is displayed.
- With `read` the news items can be read.
- With `purge` news items can be removed once they have been read and will not be reread anymore.

```
root # eselect news list
```

```
root # eselect news read
```

More information about the news reader is available through its manual page:

```
root # man news.eselect
```

## Choosing the right profile

### Tip

Desktop profiles are not exclusively for *desktop environments*. They are also suitable for minimal window managers like i3 or sway.

A *profile* is a building block for any Gentoo system. Not only does it specify default values for *USE*, *CFLAGS*, and other important variables, it also locks the system to a certain range of package versions. These settings are all maintained by Gentoo's Portage developers.

To see what profile the system is currently using, run **eselect** using the `profile` module:

```
root # eselect profile list
```

```
Available profile symlink targets:
```

- [1] default/linux/amd64/23.0 \*
- [2] default/linux/amd64/23.0/desktop
- [3] default/linux/amd64/23.0/desktop/gnome
- [4] default/linux/amd64/23.0/desktop/kde

### Note

The output of the command is just an example and evolves over time.

### Note

To use **systemd**, select a profile which has "systemd" in the name and vice versa, if not

There are also desktop sub-profiles available for some architectures which include software packages commonly necessary for a desktop experience.

### Warning

Profile upgrades are not to be taken lightly. When selecting the initial profile, use the profile corresponding to **the same version** as the one initially used by the stage file (e.g. 23.0). Each new profile version is announced through a news item containing migration instructions; be sure to carefully follow the instructions before switching to a newer profile.

After viewing the available profiles for the amd64 architecture, users can select a different profile for the system:

```
root # eselect profile set 2
```

## No-multilib

In order to select a pure 64-bit environment, with no 32-bit applications or libraries, use a no-multilib profile:

```
root # eselect profile list
```

```
Available profile symlink targets:
```

- [1] default/linux/amd64/23.0 \*
- [2] default/linux/amd64/23.0/desktop
- [3] default/linux/amd64/23.0/desktop/gnome
- [4] default/linux/amd64/23.0/desktop/kde
- [5] default/linux/amd64/23.0/no-multilib

Next select the *no-multilib* profile:

```
root # eselect profile set 5
```

```
root # eselect profile list
```

```
Available profile symlink targets:
```

- [1] default/linux/amd64/23.0
- [2] default/linux/amd64/23.0/desktop
- [3] default/linux/amd64/23.0/desktop/gnome
- [4] default/linux/amd64/23.0/desktop/kde
- [5] default/linux/amd64/23.0/no-multilib \*

### Note

The `developer` sub-profile is specifically for Gentoo Linux development and is not meant to be used by casual users.

## Optional: Adding a binary package host

Since December 2023, Gentoo's Release Engineering team (</wiki/Project:RelEng>) has offered an official binary package host ([/wiki/Binary\\_package\\_quickstart](/wiki/Binary_package_quickstart)) (colloquially shorted to just "binhost") for use by the general community to retrieve and install binary packages (binpkgs).<sup>[1]</sup>

Adding a binary package host allows Portage to install cryptographically signed, compiled packages. In many cases, adding a binary package host will *greatly* decrease the mean time to package installation and adds much benefit when running Gentoo on older, slower, or low power systems.

## Repository configuration

The repository configuration for a binhost is found in Portage's `/etc/portage/binrepos.conf/` directory, which functions similarly to the configuration mentioned in the Gentoo ebuild repository ([/wiki/Handbook:AMD64/Installation/Base#Gentoo\\_ebuild\\_repository](/wiki/Handbook:AMD64/Installation/Base#Gentoo_ebuild_repository)) section.

When defining a binary host, there are two important aspects to consider:

1. The architecture and profile targets within the `sync-uri` value *do* matter and should align to the respective computer architecture (**amd64** in this case) and system profile selected in the Choosing the right profile ([/wiki/Handbook:AMD64/Installation/Base#Choosing\\_the\\_right\\_profile](/wiki/Handbook:AMD64/Installation/Base#Choosing_the_right_profile)) section.
2. Selecting a fast, geographically close mirror will generally shorten retrieval time. Review the **mirrorselect** tool mentioned in the Optional: Selecting mirrors ([/wiki/Handbook:AMD64/Installation/Base#Gentoo\\_ebuild\\_repository](/wiki/Handbook:AMD64/Installation/Base#Gentoo_ebuild_repository)) section or review the online list of mirrors (<https://www.gentoo.org/downloads/mirrors/>) where URL values can be discovered.

**FILE** `/etc/portage/binrepos.conf/gentoobinhost.conf` **CDN-based binary package host example**

```
[binhost]
priority = 9999
sync-uri = https://distfiles.gentoo.org/releases/<arch>/binpackages/<profile>/x86-64/
```

## Installing binary packages

Portage will compile packages from code source by default. It can be instructed to use binary packages in the following ways:

1. The `--getbinpkg` option can be passed when invoking the **emerge** command. This method of for binary package installation is useful to install only a particular binary package.
2. Changing the system's default via Portage's `FEATURES` variable, which is exposed through the `/etc/portage/make.conf` file. Applying this configuration change will cause Portage to query the binary package host for the package(s) to be requested and fall back to compiling locally when no results are found.

For example, to have Portage always install available binary packages:

**FILE** `/etc/portage/make.conf` **Configure Portage to use binary packages by default**

```
# Appending getbinpkg to the list of values within the FEATURES variable
FEATURES="${FEATURES} getbinpkg"
# Require signatures
FEATURES="${FEATURES} binpkg-request-signature"
```

Please also run **getuto** for Portage to set up the necessary keyring for verification:

```
root # getuto
```

Additional Portage features will be discussed in the the next chapter ([/wiki/Handbook:AMD64/Working/Features#Portage\\_features](/wiki/Handbook:AMD64/Working/Features#Portage_features)) of the handbook.

## Optional: Configuring the USE variable

`USE` is one of the most powerful variables Gentoo provides to its users. Several programs can be compiled with or without optional support for certain items. For instance, some programs can be compiled with support for GTK+ or with support for Qt. Others can be compiled with or without SSL support. Some programs can even be compiled with framebuffer support (svglib) instead of X11 support (X-server).

Most distributions compile their packages with support for as much as possible, increasing the size of the programs and startup time, not to mention an enormous amount of dependencies. With Gentoo, users can define what options for which a package should be compiled. This is where `USE` comes into play.

In the `USE` variable users define keywords which are mapped onto compile-options. For instance, `ssl` will compile SSL support in the programs that support it. `-X` will remove X-server support (note the minus sign in front). `gnome gtk -kde -qt5` will compile programs with GNOME (and GTK+) support, and not with KDE (and Qt) support, making the system fully tweaked for GNOME (if the architecture supports it).

The default `USE` settings are placed in the `make.defaults` files of the Gentoo profile used by the system. Gentoo uses a complex inheritance system for system profiles, which will not be covered in depth during the installation process. The easiest way to check the currently active `USE` settings is to run **emerge --info** and select the line that starts with `USE`:

```
root # emerge --info | grep ^USE
```

```
USE="X acl alsa amd64 berkdb bindist bzip2 cli cracklib crypt cxx dri ..."
```

### Note

The above example is truncated, the actual list of `USE` values is much, much larger.

A full description on the available USE flags can be found on the system in `/var/db/repos/gentoo/profiles/use.desc`.

```
root # less /var/db/repos/gentoo/profiles/use.desc
```

Inside the **less** command, scrolling can be done using the `↑` and `↓` keys, and exited by pressing `q`.

As an example we show a USE setting for a KDE-based system with DVD, ALSA, and CD recording support:

```
root # nano /etc/portage/make.conf
```

```
FILE /etc/portage/make.conf Enabling flags for a KDE/Plasma-based system with DVD, ALSA, and CD recording support
```

```
USE="-gtk -gnome qt5 kde dvd alsa cdr"
```

When a *USE* value is defined in `/etc/portage/make.conf` it is *added* to the system's USE flag list. USE flags can be globally *removed* by adding a `-` minus sign in front of the value in the the list. For example, to disable support for X graphical environments, `-X` can be set:

```
FILE /etc/portage/make.conf Ignoring default USE flags
```

```
USE="-X acl alsa"
```

#### Warning

Although possible, setting `-*` (which will disable all USE values except the ones specified in **make.conf**) is *strongly* discouraged and unwise. Ebuild developers choose certain default USE flag values in ebuids in order to prevent conflicts, enhance security, and avoid errors, and other reasons. Disabling *all* USE flags will negate default behavior and may cause major issues.

## CPU\_FLAGS\_\*

Some architectures (including AMD64/X86, ARM, PPC) have a `USE_EXPAND` ([/wiki/USE\\_EXPAND](#)) variable called `CPU_FLAGS_<ARCH>` ([/wiki/CPU\\_FLAGS\\_X86](#)), where `<ARCH>` is replaced with the relevant system architecture name.

#### Important

Do not be confused! **AMD64** and **X86** systems share some common architecture, so the proper variable name for **AMD64** systems is `CPU_FLAGS_X86`.

This is used to configure the build to compile in specific assembly code or other intrinsics, usually hand-written or otherwise extra, and is **not** the same as asking the compiler to output optimized code for a certain CPU feature (e.g. `-march=`).

Users should set this variable in addition to configuring their `COMMON_FLAGS` as desired.

A few steps are needed to set this up:

```
root # emerge --ask --oneshot app-portage/cpuid2cpuflags
```

Inspect the output manually if curious:

```
root # cpuid2cpuflags
```

Then copy the output into **package.use**:

```
root # echo "*/ * $(cpuid2cpuflags)" > /etc/portage/package.use/00cpu-flags
```

## VIDEO\_CARDS

The `VIDEO_CARDS USE_EXPAND` variable should be configured appropriately depending on the available GPU(s). Setting `VIDEO_CARDS` is not required for a console only install.

Below is an example of a properly set `VIDEO_CARDS` variable. Substitute the name of the driver(s) to be used.

```
FILE /etc/portage/make.conf
```

```
VIDEO_CARDS="amdgpu radeonsi"
```

Details for various GPU(s) can be found at the [AMDGPU](#) ([/wiki/AMDGPU](#)), [Intel](#) ([/wiki/Intel](#)), [Nouveau](#) (Open Source) ([/wiki/Nouveau](#)), or [NVIDIA](#) (Proprietary) ([/wiki/NVIDIA](#)) articles.

## Optional: Configure the ACCEPT\_LICENSE variable

Starting with Gentoo Linux Enhancement Proposal 23 (GLEP 23), a mechanism was created to allow system administrators the ability to "regulate the software they install with regards to licenses... Some want a system free of any software that is not OSI-approved; others are simply curious as to what licenses they are implicitly accepting."<sup>[2]</sup> With a motivation to have more granular control over the type of software running on a Gentoo system, the `ACCEPT_LICENSE` variable was born.

During the installation process, Portage considers the value(s) set within the `ACCEPT_LICENSE` variable to determine if the requested package(s) meet the sysadmin's determination of an acceptable license. Here in lies a problem: the Gentoo ebuild repository is filled with *thousands* of ebuids which results in *hundreds* of distinct software licenses (<https://gitweb.gentoo.org/repo/gentoo.git/tree/licenses>)... Does this implicate sysadmin into individually approving each and every new software license? Thankfully no; GLEP 23 also outlines a solution to this problem, a concept called license groups.

For the convenience of system administration, legally-similar software licenses have been bundled together - each according to its like-kind. License group definitions are available for viewing ([https://gitweb.gentoo.org/repo/gentoo.git/tree/profiles/license\\_groups](https://gitweb.gentoo.org/repo/gentoo.git/tree/profiles/license_groups)) and are managed by the Gentoo Licenses project ([/wiki/Project:Licenses](#)). While an individual license is not, license groups are syntactically preceded with an `@` symbol, enabling them to be easily distinguished in the `ACCEPT_LICENSE` variable.



Some common license groups include:

A list of software licenses grouped according to their kinds.

Name	Description
@GPL-COMPATIBLE	GPL compatible licenses approved by the Free Software Foundation <sup>[a_license 1]</sup>
@FSF-APPROVED	Free software licenses approved by the FSF (includes @GPL-COMPATIBLE )
@OSI-APPROVED	Licenses approved by the Open Source Initiative <sup>[a_license 2]</sup>
@MISC-FREE	Misc licenses that are probably free software, i.e. follow the Free Software Definition <sup>[a_license 3]</sup> but are not approved by either FSF or OSI
@FREE-SOFTWARE	Combines @FSF-APPROVED , @OSI-APPROVED , and @MISC-FREE .
@FSF-APPROVED-OTHER	FSF-approved licenses for "free documentation" and "works of practical use besides software and documentation" (including fonts)
@MISC-FREE-DOCS	Misc licenses for free documents and other works (including fonts) that follow the free definition <sup>[a_license 4]</sup> but are NOT listed in @FSF-APPROVED-OTHER .
@FREE-DOCUMENTS	Combines @FSF-APPROVED-OTHER and @MISC-FREE-DOCS .
@FREE	Metaset of all licenses with the freedom to use, share, modify and share modifications. Combines @FREE-SOFTWARE and @FREE-DOCUMENTS .
@BINARY-REDISTRIBUTABLE	Licenses that at least permit free redistribution of the software in binary form. Includes @FREE .
@EULA	License agreements that try to take away your rights. These are more restrictive than "all-rights-reserved" or require explicit approval

1. <https://www.gnu.org/licenses/license-list.html> (<https://www.gnu.org/licenses/license-list.html>)
2. <https://www.opensource.org/licenses> (<https://www.opensource.org/licenses>)
3. <https://www.gnu.org/philosophy/free-sw.html> (<https://www.gnu.org/philosophy/free-sw.html>)
4. <https://freedomdefined.org/> (<https://freedomdefined.org/>)

Currently set system wide acceptable license values can be viewed via:

```
user $ portageq envvar ACCEPT_LICENSE
```

```
@FREE
```

As visible in the output, the default value is to only allow software which has been grouped into the @FREE category to be installed.

Specific licenses or licenses groups for a system can be defined in the following locations:

- System wide within the selected profile - this sets the default value.
- System wide within the **/etc/portage/make.conf** file. System administrators override the profile's default value within this file.
- Per-package within a **/etc/portage/package.license** file.
- Per-package within a **/etc/portage/package.license/** directory of files.

The system wide license default in the profile is overridden within the **/etc/portage/make.conf**:

```
FILE /etc/portage/make.conf Accept licenses with ACCEPT_LICENSE system wide
```

```
# Overrides the profile's ACCEPT_LICENSE default value
ACCEPT_LICENSE="-* @FREE @BINARY-REDISTRIBUTABLE"
```

Optionally system administrators can also define accepted licenses per-package as shown in the following directory of files example. Note that the **package.license** directory will need created if it does not already exist:

```
root # mkdir /etc/portage/package.license
```

Software license details for an individual Gentoo package are stored within the **LICENSE** variable of the associated ebuild. One package may have one or many software licenses, therefore it be necessary to specify multiple acceptable licenses for a single package.

```
FILE /etc/portage/package.license/kernel Accepting licenses on a per-package basis
```

```
app-arch/unrar unrar
sys-kernel/linux-firmware linux-fw-redistributable
sys-firmware/intel-microcode intel-ucode
```

#### Important

The **LICENSE** variable in an ebuild is only a guideline for Gentoo developers and users. It is *not* a legal statement, and there is *no guarantee* that it will reflect reality. It is recommended to not solely rely on a ebuild developer's interpretation of a software package's

license; but check the package itself in depth, including all files that have been installed to the system.

## Optional: Updating the @world set

Updating the system's @world set ([/wiki/World\\_set\\_\(Portage\)](#)) is *optional* and will be unlikely to perform functional changes unless one or more of the following optional steps have been performed:

1. A profile target *different* from the stage file has been selected.
2. Additional USE flags have been set for installed packages.

Readers who are performing an 'install Gentoo speed run' may safely skip @world set updates until *after* their system has rebooted into the new Gentoo environment.

Readers who are performing a slow run can have Portage perform updates for package, profile, and/or USE flag changes at the present time:

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

Readers who added a binary host above ([/wiki/Handbook:AMD64/Installation/Base#Optional: Adding a binary package host](#)) can add **--getbinpkg** (or **-g**) in order to fetch packages from the binary host instead of compiling them:

```
root # emerge --ask --verbose --update --deep --newuse --getbinpkg @world
```

## Removing obsolete packages

It is important to always *depclean* after system upgrades to remove obsolete packages. Review the output carefully with **emerge --depclean --pretend** to see if any of the to-be-cleaned packages should be kept if personally using them. To keep a package which would otherwise be depcleaned, use **emerge --noreplace foo**.

```
root # emerge --ask --pretend --depclean
```

If happy, then proceed with a real depclean:

```
root # emerge --ask --depclean
```

### Tip

If a desktop environment profile target has been selected from a non-desktop stage file, the @world update process could greatly extend the amount of time necessary for the install process. Those in a time crunch can work by this 'rule of thumb': the shorter the profile name, the less specific the system's @world set ([/wiki/World\\_set\\_\(Portage\)](#)). The less specific the @world set, the fewer packages the system will require. E.g.:

- Selecting `default/linux/amd64/23.0` will likely require fewer packages to be updated, whereas
- Selecting `default/linux/amd64/23.0/desktop/gnome/systemd` will likely require more packages to be installed since the profile target has a larger @system ([/wiki/Package\\_sets#.40system](#)) and @profile ([/wiki/Package\\_sets#.40profile](#)) sets: dependencies supporting the GNOME desktop environment.

## Optional: Using systemd as the system and service manager

The remainder of the Gentoo handbook will provide systemd steps alongside OpenRC ([/wiki/OpenRC](#)) (the traditional Gentoo init system) where separate steps or recommendations are necessary. System administrators should also consult the systemd ([/wiki/Systemd](#)) article for more details on managing systemd as the system and service manager.

## Timezone

### Note

This step does not apply to users of the musl libc. Users who do not know what that means should perform this step.

### Warning

Please avoid the `/usr/share/zoneinfo/Etc/GMT*` timezones as their names do not indicate the expected zones. For instance, **GMT-8** is in fact **GMT+8**.

Select the timezone for the system. Look for the available timezones in `/usr/share/zoneinfo/`:

```
root # ls -l /usr/share/zoneinfo
```

```
root # ls -l /usr/share/zoneinfo/Europe/
```

[Expand]

Suppose the timezone of choice is **Europe/Brussels**, to select this timezone, a symlink can be created from this zoneinfo file to `/etc/localtime`: [Expand]

```
root # ln -sf ../usr/share/zoneinfo/Europe/Brussels /etc/localtime
```

### Tip

The target path with `../` at the start is *relative to the link location*, not the current directory.

### Note

An absolute path can be used for the symlink, but a relative link is also created by systemd's **timedatectl** and is more compatible



with alternate *ROOTs*.

## Configure locales

### Note

This step does not apply to users of the `musl` libc. Users who do not know what that means should perform this step.

## Locale generation

Most users will want to use only one or two locales on their system.

Locales specify not only the language that the user should use to interact with the system, but also the rules for sorting strings, displaying dates and times, etc. Locales are *case sensitive* and must be represented exactly as described. A full listing of available locales can be found in the `/usr/share/i18n/SUPPORTED` file.

Supported system locales must be defined in the `/etc/locale.gen` file.

```
root # nano /etc/locale.gen
```

The following locales are an example to get both English (United States) and German (Germany/Deutschland) with the accompanying character formats (like UTF-8).

**FILE** `/etc/locale.gen` Enabling US and DE locales with the appropriate character formats

```
en_US ISO-8859-1
en_US.UTF-8 UTF-8
de_DE ISO-8859-1
de_DE.UTF-8 UTF-8
```

### Warning

Many applications require least one UTF-8 locale to build properly.

The next step is to run the `locale-gen` command. This command generates all locales specified in the `/etc/locale.gen` file.

```
root # locale-gen
```

To verify that the selected locales are now available, run `locale -a`.

On `systemd` installs, `localectl` can be used, e.g. `localectl set-locale ...` or `localectl list-locales`.

## Locale selection

Once done, it is now time to set the system-wide locale settings. Again `eselect` is used, now with the `locale` module.

With `eselect locale list`, the available targets are displayed:

```
root # eselect locale list
```

```
Available targets for the LANG variable:
[1]  C
[2]  C.utf8
[3]  en_US
[4]  en_US.iso88591
[5]  en_US.utf8
[6]  de_DE
[7]  de_DE.iso88591
[8]  de_DE.utf8
[9]  POSIX
[ ]  (free form)
```

With `eselect locale set <NUMBER>` the correct locale can be selected:

```
root # eselect locale set 2
```

Manually, this can still be accomplished through the `/etc/env.d/02locale` file and for `systemd` the `/etc/locale.conf` file:

**FILE** `/etc/env.d/02locale` Manually setting system locale definitions

```
LANG="de_DE.UTF-8"
LC_COLLATE="C.UTF-8"
```

Setting the locale will avoid warnings and errors during kernel and software compilations later in the installation.

Now reload the environment:

```
root # env-update && source /etc/profile && export PS1="(chroot) ${PS1}"
```

For additional guidance through the locale selection process read also the Localization guide (</wiki/Localization/Guide>) and the UTF-8 (</wiki/UTF-8>) guide.

## References

---

1. <https://www.gentoo.org/news/2023/12/29/Gentoo-binary.html> (<https://www.gentoo.org/news/2023/12/29/Gentoo-binary.html>)
2. <https://www.gentoo.org/glep/glep-0023.html#id7> (<https://www.gentoo.org/glep/glep-0023.html#id7>)

← Installing the stage file (/wiki/Handbook:AMD64/Installation/Base) andbook:AMD64)      Configuring the kernel → (/wiki/Handbook:AMD64/Installation/Kernel)

Retrieved from "<https://wiki.gentoo.org/index.php?title=Handbook:AMD64/Installation/Base&oldid=212418> (<https://wiki.gentoo.org/index.php?title=Handbook:AMD64/Installation/Base&oldid=212418>)"

- This page was last edited on 2 January 2015, at 00:03.
- [Privacy policy \(/wiki/Gentoo\\_Wiki:Privacy\\_policy\)](/wiki/Gentoo_Wiki:Privacy_policy)
- [About Gentoo Wiki \(/wiki/Gentoo\\_Wiki:About\)](/wiki/Gentoo_Wiki:About)
- [Disclaimers \(/wiki/Gentoo\\_Wiki:General\\_disclaimer\)](/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 \(https://creativecommons.org/licenses/by-sa/4.0/\)](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.

---