

Official repositories

A **[software repository](#)** is a storage location from which software packages are retrieved for installation.

Arch Linux **official repositories** contain essential and popular software, readily accessible via **[pacman](#)**. They are maintained by **[package maintainers](#)**.

Packages in the official repositories are constantly upgraded: when a package is upgraded, its old version is removed from the repository. There are no major Arch releases: each package is upgraded as new versions become available from upstream sources. Each repository is always coherent, i.e. the packages that it hosts always have reciprocally compatible versions.

Contents

[Stable repositories](#)

[core](#)

[extra](#)

[community](#)

[multilib](#)

[Enabling multilib](#)

[Disabling multilib](#)

[Testing repositories](#)

[testing](#)

[community-testing](#)

[multilib-testing](#)

[gnome-unstable](#)

[kde-unstable](#)

[Disabling testing repositories](#)

[Staging repositories](#)

[Historical background](#)

Related articles

[Arch Build System](#)

[Arch User Repository](#)

[makepkg](#)

[Mirrors](#)

[Official repositories web interface](#)

[pacman](#)

[PKGBUILD](#)

[Unofficial user repositories](#)

Stable repositories

core

This repository can be found in `.../core/os/` on your favorite [mirror](#).

core contains packages for:

- booting Arch Linux
- [connecting to the Internet](#)

- [building packages](#)
- management and repair of supported [file systems](#)
- the system setup process (e.g. [openssh](#) (<https://archlinux.org/packages/?name=open+ssh>))

as well as dependencies of the above (not necessarily [makedepends](#)) and the [base](#) (<https://archlinux.org/packages/?name=base>) [meta package](#).

core has fairly strict quality requirements. Developers/users need to signoff on updates before package updates are accepted. For packages with low usage, a reasonable exposure is enough: informing people about update, requesting signoffs, keeping in [testing](#) up to a week depending on the severity of the change, lack of outstanding bug reports, along with the implicit signoff of the package maintainer.

Tip: To create a local repository with packages from *core* (or other repositories) without an internet connection see [Pacman tips#Installing packages from a CD/DVD or USB stick](#)

extra

This repository can be found in `.../extra/os/` on your favorite mirror.

extra contains all packages that do not fit in *core*. Example: Xorg, window managers, web browsers, media players, tools for working with languages such as Python and Ruby, and a lot more.

community

This repository can be found in `.../community/os/` on your favorite mirror.

community contains packages that have been adopted by [Trusted Users](#) from the [Arch User Repository](#). Some of these packages may eventually make the transition to the [core](#) or [extra](#) repositories as the developers consider them crucial to the distribution.

multilib

This repository can be found in `.../multilib/os/` on your favorite mirror.

multilib contains 32-bit software and libraries that can be used to run and build 32-bit applications on 64-bit installs (e.g. [wine](#) (<https://archlinux.org/packages/?name=wine>), [steam](#) (<https://archlinux.org/packages/?name=steam>), etc).

With the multilib repository enabled, the 32-bit compatible libraries are located under `/usr/lib32/`.

Enabling multilib

To enable multilib repository, uncomment the `[multilib]` section in `/etc/pacman.conf`:

```
/etc/pacman.conf
```

```
[multilib]
Include = /etc/pacman.d/mirrorlist
```

Then **upgrade** the system and install the desired multilib packages.

Tip: Run `pacman -Sl multilib` to list all packages in the *multilib* repository. 32-bit library package names begin with `lib32-`.

Disabling multilib

Execute the following command to remove all packages that were installed from *multilib*:

```
# pacman -R $(comm -12 <(pacman -Qq | sort) <(pacman -Slq multilib | sort))
```

If you have conflicts with `gcc-libs` reinstall the **gcc-libs** (<https://archlinux.org/packages/?name=gcc-libs>) package and the **base-devel** (https://archlinux.org/groups/x86_64/base-devel/) group.

Comment out the `[multilib]` section in `/etc/pacman.conf`:

```
/etc/pacman.conf
```

```
#[multilib]
#Include = /etc/pacman.d/mirrorlist
```

Then **upgrade** the system.

Testing repositories

The intended purpose of the *testing* repository is to provide a staging area for packages to be placed prior to acceptance into the main repositories. Package maintainers (and general users) can then access these testing packages to make sure that there are no problems integrating the new package. Once a package has been tested and no errors are found, the package can then be moved to the primary repositories.

Not all packages need to go through this testing process. However, all packages destined for the *core* repository must go to *testing* first. Packages that can affect many packages (such as **perl** (<https://archlinux.org/packages/?name=perl>) or **python** (<https://archlinux.org/packages/?name=python>)) should be tested as well. Testing is also usually used for large collections of packages such as **GNOME** and **KDE**.

Warning:

- Be careful when enabling the *testing* repositories. Your system may break after performing an update. Only experienced users who know how to deal with potential system breakage should use it.
- If you enable *testing*, you must also enable *community-testing*. If you enable any other testing repository listed in the following subsections, you must also enable both *testing* and *community-testing*.

testing

This repository can be found in `.../testing/os/` on your favorite mirror.

testing contains packages that are candidates for the *core* or *extra* repositories.

New packages go into *testing* if:

- They are destined for the *core* repo. Everything in *core* must go through *testing*
- They are expected to break something on update and need to be tested first.

testing is the only repository that can have name collisions with any of the other official repositories. If enabled, it has to be the first repository listed in your `/etc/pacman.conf` file.

Note: *testing* is not for the "newest of the new" package versions. Part of its purpose is to hold package updates that have the potential to break the system, either by being part of the *core* set of packages, or by being critical in other ways. As such, users of *testing* are strongly encouraged to subscribe to the [arch-dev-public mailing list \(https://mailman.archlinux.org/mailman/listinfo/arch-dev-public\)](https://mailman.archlinux.org/mailman/listinfo/arch-dev-public), watch the [testing repository forum \(https://bbs.archlinux.org/viewforum.php?id=49\)](https://bbs.archlinux.org/viewforum.php?id=49), and to [report all bugs](#). You should also consider joining the [Arch Testing Team](#).

community-testing

This repository is similar to the *testing* repository, but for packages that are candidates for the *community* repository.

multilib-testing

This repository is similar to the *testing* repository, but for packages that are candidates for the *multilib* repository.

gnome-unstable

This repository contains testing packages for the next stable or stable release candidate version of the [GNOME](#) desktop environment, before they are moved to the main *testing* repository.

To enable it, add the following lines to `/etc/pacman.conf` :

```
[gnome-unstable]
Include = /etc/pacman.d/mirrorlist
```

The *gnome-unstable* entry should be first in the list of repositories (*i.e.*, above the *testing* entry).

Please report packaging related bugs in our [bug tracker \(https://bugs.archlinux.org/\)](https://bugs.archlinux.org/), while anything else should be reported upstream to [GNOME Gitlab \(https://gitlab.gnome.org\)](https://gitlab.gnome.org).

kde-unstable

This repository contains the latest *beta* or *Release Candidate* of [KDE](#) Plasma and Applications.

To enable it, add the following lines to `/etc/pacman.conf` :

```
[kde-unstable]
Include = /etc/pacman.d/mirrorlist
```

The *kde-unstable* entry should be first in the list of repositories (*i.e.*, above the *testing* entry).

Make sure [you make bug reports](#) if you find any problems.

Disabling testing repositories

If you enabled testing repositories, but later on decided to disable them, you should:

1. Remove (comment out) them from `/etc/pacman.conf`
2. Perform a `pacman -Syuu` to "rollback" your updates from these repositories.

The second item is optional, but keep it in mind if you notice any problems.

Staging repositories

Warning: Do not enable the *staging* repositories for any reason. Your system will unquestionably break after performing an update. This repository is only meant for backend developer use.

This repository contains broken packages and is used solely by developers during rebuilds of many packages at once. In order to rebuild packages that depend on, for example, a new shared library, the shared library itself must first be built and uploaded to the staging repositories to be made available to other developers. As soon as all dependent packages are rebuilt, the group of packages is then moved to testing or to the main repositories, whichever is more appropriate.

See [the announcement of the introduction of the staging repositories \(https://lists.archlinux.org/pipermail/arch-dev-public/2010-August/017579.html\)](https://lists.archlinux.org/pipermail/arch-dev-public/2010-August/017579.html) for more historical details.

Historical background

Most of the repository splits are for historical reasons. Originally, when Arch Linux was used by very few users, there was only one repository known as **official** (now *core*). At the time, *official* basically contained Judd Vinet's preferred applications. It was designed to contain one of each "type" of program — one DE, one major browser, etc.

There were users back then that did not like Judd's selection, so since the [Arch Build System](#) is so easy to use, they created packages of their own. These packages went into a repository called **unofficial**, and were maintained by developers other than Judd. Eventually, the two repositories were both considered equally supported by the developers, so the names *official* and *unofficial* no longer reflected their true purpose. They were subsequently renamed to **current** and **extra** sometime near the release version 0.5.

Shortly after the 2007.8.1 release, *current* was renamed **core** in order to prevent confusion over what exactly it contains. The repositories are now more or less equal in the eyes of the developers and the community, but *core* does have some differences. The main distinction is that packages used for Installation CDs and release snapshots are taken only from *core*. This repository still gives a complete Linux system, though it may not be the Linux system you want.

Some time around 0.5/0.6, there were a lot of packages that the developers did not want to maintain. **Jason Chu** (<https://archlinux.org/people/developer-fellows/#jason>) set up the "Trusted User Repositories", which were unofficial repositories in which trusted users could place packages they had created. There was a **staging** repository where packages could be promoted into the official repositories by one of the Arch Linux developers, but other than this, the developers and trusted users were more or less distinct.

This worked for a while, but not when trusted users got bored with their repositories, and not when untrusted users wanted to share their own packages. This led to the development of the **AUR** (<https://aur.archlinux.org/>). The TUs were conglomerated into a more closely knit group, and they now collectively maintain the **community** repository. The Trusted Users are still a separate group from the Arch Linux developers, and there is not a lot of communication between them. However, popular packages are still promoted from *community* to *extra* on occasion. The **AUR** (<https://aur.archlinux.org/>) also allows untrusted users to submit PKGBUILDs.

After a kernel in *core* **broke many user systems** (<https://archlinux.org/news/please-avoid-kernel-261614-1/>), the "core signoff policy" was introduced. Since then, all package updates for *core* need to go through a **testing** repository first, and only after multiple signoffs from other developers are then allowed to move. Over time, it was noticed that various *core* packages had low usage, and user signoffs or even lack of bug reports became informally accepted as criteria to accept such packages.

In late 2009/the beginning of 2010, with the advent of some new filesystems and the desire to support them during installation, along with the realization that *core* was never clearly defined (just "important packages, handpicked by developers"), the repository received a more accurate description.

Retrieved from "https://wiki.archlinux.org/index.php?title=Official_repositories&oldid=716487"

This page was last edited on 6 February 2022, at 06:38.

Content is available under [GNU Free Documentation License 1.3 or later](#) unless otherwise noted.

- [Privacy policy](#)
- [About ArchWiki](#)
- [Disclaimers](#)