



[Translation\(s\)](#): [English](#) - [Français](#) - [Italiano](#) - [Português \(Brasil\)](#) - [Русский](#) - [Svenska](#) - [简体中文](#)

A Debian repository is a set of Debian binary or source packages organized in a special directory tree and with various infrastructure files - checksums, indices, signatures, descriptions translations, ... - added. Client computers can connect to the repository to download and install the packages using an [Apt](#)-based [PackageManagement](#) tool.

Inhaltsverzeichnis

1. [Anatomy of a repository](#)
2. [Mirrors](#)
3. [Package lists](#)
4. [Working with repositories](#)
 1. [Using a repository](#)
 2. [Set up and maintain a repository](#)

Anatomy of a repository

The following description is mostly for people who browse a repository using a standard web browser and wonder what is where and how everything fits together. A more precise and technical description is in [Format](#).

A Debian repository contains several **releases**. Debian releases are named after characters from the "Toy Story" movies (wheezy, jessie, stretch, ...). The codenames have aliases, so called **suites** (stable, oldstable, testing, unstable). A release is divided in several **components**. In Debian these are named main, contrib, and non-free and indicate the licensing terms of the software they contain. A release also has packages for various **architectures** (amd64, i386, mips, powerpc, s390x, ...) as well as sources and architecture independent packages.

The root directory of a repository has a directory `dists` which in turn has a directory for each release and suite, the latter usually symlinks to the former, but the browser won't show you a difference. Each release subdirectory contains a cryptographically signed `Release` file and a directory for each component. Inside these are directories for the different architectures, named `binary-<arch>` and `sources`. And in these are files `Packages` that are text files


containing the meta data of packages. Hmm, so where are the actual packages?

The packages themselves are below `pool` in the root directory of the repository. Below `pool` there are again directories for all the components, and in these are directories named `0`, ..., `9`, `a`, `b`, ... `z`, `liba`, ... , `libz`. And in these are directories named after the software package they contain, and these directories finally contain the actual packages, i.e the `.deb` files. The name is not necessarily the name of the package itself, the package `bsdutils` e.g resides in the `pool/main/u/util-linux` directory, it is the name of the source that the package is generated from. A single upstream source may generate several binary packages, and all of them will end up in the same subdirectory below `pool`. The additional single letter directories are just a trick to avoid having too many entries in a single directory which is what many systems traditionally have performance problems with.





In the leaf directories below `pool` there usually are several versions of a package, and the information on what releases each version belongs to solely resides in the indices. This way, the same version of a package can belong to several releases but use disk space only once, and that without resorting to hardlinks or symlinks, so mirroring is rather simple and even works with systems that don't have these concepts.

Mirrors

The official Debian package repository is mirrored all over the world.

- The Debian website has an  [official list of Debian mirrors](#)
- There also is a list of [unofficial repositories](#) containing various additional software
- There is a [team in charge of mirrors](#).

Package lists

To browse the list of Debian packages grouped by category, you can look at the  [stable](#),  [testing](#),  [unstable](#) and various other lists, or search on the  [packages homepage](#)

Working with repositories

Working with repositories may mean either of two different things:

1. You can use a repository with the apt family of programs (apt, apt-get, apt-cache, aptitude) to browse or install packages
2. You can set up a repository yourself and add, remove or replace packages in it.

Using a repository

Using a repository is fairly simple: For the official Debian repository open `/etc/apt/sources.list`, insert the line

```
deb http://ftp.debian.org/debian stable main contrib non-free
```

then run

```
apt update
```

(As the root user or using sudo of course). You might also consider to use a URL like:

- `http://ftp.<CC>.debian.org/debian` to use a location closer to you, network-wise.
- `http://deb.debian.org/debian` to let a CDN choose the closest mirror to you (more [here](#))

For the official Debian repositories this simple procedure is sufficient. The [\[SourcesList\]](#) article contains further information about sources.list entries. For other third party repositories, you might want to follow [this excellent guide](#) to reduce the risk of hosing your system from a carelessly maintained repository.

Set up and maintain a repository

There are various reasons for building a repository yourself. You might just have a few packages with local modifications that you want to make available to apt, you may want to run a local mirror with those packages used by several

machines to save bandwidth, or you build packages yourself and want to test them prior to publication.

- A comprehensive list of [tools to set up a repository](#).
- [Feature Comparison](#) of various repository handling software (incomplete)
- Guide how to set up a repository with [reprepro](#)
- How to create a package and a repository with [mini-dinstall](#)

[CategoryPackageManagement](#)