- blender.org
- code.blender.org

Dev:Doc/Building Blender/Linux /Ubuntu/CMake

Log in
< Dev:Doc | Building Blender | Linux | Ubuntu

Blender 2.6

- Blender 2.6
- Blender 2.5
- **Blender 2.4**

English

- Arabic
- Bulgarian
- Catalan
- Czech
- German
- Danish
- **English**
- Greek
- Esperanto
- Spanish
- Estonian
- Farsi

- Finnish
- French
- Indonesian
- Italian
- Japanese
- Korean
- Lithuanian
- Macedonian
- Mongolian
- Dutch
- Polish
- Portuguese

- Romanian
- Russian
- Serbian

- Swedish
- Thai
- Turkish
- Ukrainian
- Chinese

- Developer Documentation page
- Discussion
- View source
- History

Page

- What links here
- Related changes
- Permanent link

From BlenderWiki

To get the latest Blender successfully running on Linux system, follow a few simple steps.

1. Download Blender source.
2. Install required package dependencies.
3. Compile Blender.

1. Get the source

The first step is to get the latest Blender source code from blender.org's GIT repository.

Copy and paste the following instructions into a terminal window. The following commands create a `blender-git` folder in your home directory by downloading the latest source code commonly referred to as `'master'`. An Internet connection is needed.

```
mkdir ~/blender-git
cd ~/blender-git
git clone https://git.blender.org/blender.git
cd blender
git submodule update --init --recursive
git submodule foreach git checkout master
git submodule foreach git pull --rebase origin master
```

If you want to update your git clone checkout to the latest source do (in ~/blender-git/blender/):

```
git pull --rebase
git submodule foreach git pull --rebase origin master
```

For additional information on using Git with Blender's sources, see: Tools/Git

2. Install/Update the dependencies

Automatic dependencies installation

The preferred way to install dependencies under Linux is now to use the `install_deps.sh` script featured with Blender sources. It currently supports Debian (and derived), Fedora, Suse and Arch distributions. When using the `install_deps.sh` script, you are only required to install the following dependencies:

```
sudo apt-get update; sudo apt-get install git build-essential
```

Then, get the sources and run `install_deps.sh`

```
cd ~/blender-git
./blender/build_files/build_environment/install_deps.sh
```

This script works for Debian/Redhat/SuSE/Arch based distributions, both 32 and 64 bits.

For other distributions, it can:

- Print the list of all main dependencies needed to build Blender (`--show-deps` option).
- Attempt to build main 'big' libraries you cannot easily install from packages (`--build-foo` options, see `--help` of the script for details).

It might be required to re-run install-depsh.sh once in a while, as Blender updates its dependencies. You will typically want to try this when you have build errors after updating the sources.

Some commands in this script requires sudo, so you'll be likely be asked for your password.

When the script finishes installing/building all the packages, it'll print which parameters for CMake and SCons you should use to build Blender.

+ install_deps.sh options

+ Manual Dependency Installation (optional)

3. Compile Blender with CMake

Installing CMake

Install CMake from your package manager.

```
sudo apt-get install cmake cmake-curses-gui
```

Automatic CMake Setup

If you're not interested in manually setting up CMake build directory, configuring, building and installing in separate steps, we provide a convenience makefile in Blender's source directory which sets up CMake for you.

```
cd ~/blender-git/blender
make
```

Once the build finishes you'll get a message like..

```
Blender successfully built, run from: /home/me/blender-git/build_linux/bin/blender
```

Updating your local checkout and rebuilding is as simple as:

```
cd ~/blender-git/blender
make update
make
```

There are some pre-defined build targets:

- **`make`** - some are turned off by default because they can be difficult to correctly configure for newer developers and aren't essential to use & develop Blender in most cases.
- **`make lite`** - the quickest way to get a Blender build up & running, can also help to avoid installing a lot of dependencies if you don't need video-codecs, physics-sim & cycles rendering.
- **`make full`** - this makes a complete build with all options enabled, matching the releases on **blender.org**.

For a full list of the optional targets type...

```
make help
```

Manual CMake Setup

+ Manual CMake Setup (optional)

Optimize Rebuilds

+ Optimize Rebuilds (optional)

See also

- CMake & IDE Configuration

Troubleshooting

+ Expand

Retrieved from "https://wiki.blender.org/index.php/Dev:Doc/Building_Blender/Linux/Ubuntu/CMake"
Category: Script