

Contents

1 Abstract	1
2 Introduction	1
3 Releases	2
3.1 Package	2
3.2 Upstream Releases	2
4 Build From Source	2
4.1 Dependencies	2
4.2 Run Make	3
5 Build Package	4
5.1 Install	5
6 Tests	5
7 Links	5
8 History	5
9 Disclaimer of Warranty	5
10 Limitation of Liability	5

1 Abstract

This document provides in-depth instructions for experienced Debian administrators seeking to compile and install a customized version of Neovim, a highly configurable text editor.

2 Introduction

The task of compiling and installing software like Neovim on Debian-based systems can be a complex undertaking when the packaged version is too old, even for seasoned administrators. This guide aims to facilitate this process, providing step-by-step instructions and addressing potential challenges associated with older systems.

3 Releases

3.1 Package

- Neovim is in Debian and a fast development can be seen

Debian	#	Neovim
Bookworm	12	0.7.2-7
Bullseye	11	0.4.4-1
Buster	10	0.3.4-3

3.2 Upstream Releases

The current **source** stable release is neovim 0.9.0 040f145 <https://github.com/neovim/neovim/archive/refs/tags/v0.9.0.tar.gz>

- It is possible to install via tar (nvim-linux64.tar.gz) <https://github.com/neovim/neovim/releases/> The current 2023-05-17 stable release is: <https://github.com/neovim/neovim/releases/download/v0.9.0/nvim-linux64.tar.gz>
- Also at <https://github.com/neovim/neovim/releases> an AppImage package is provided

4 Build From Source

This section describe the build of Neovim from source.

4.1 Dependencies

From the repository's root directory, running make will download and build all the needed dependencies and put the `nvim` executable in `build/bin`. However to be consistent the dependencies also can be installed via package manager.

```
aptitude install ninja-build gettext libtool libtool-bin autoconf automake  
↩ \  
cmake g++ pkg-config unzip curl git
```

The `ninja-build` is a optional build dependency. It is more a development dependency, Neovim will build without, but will build faster with. The build will use `ccache` if available. To disable: `CCACHE_DISABLE=true make`.

The compilation of Neovim can present hurdles, particularly on older systems. A key challenge is that the build process requires `cmake` version 3.10 or above, while Debian 9 Stretch only includes version 3.7.2.

Here are the CMake versions bundled with recent stable Debian releases:

Debian	#	Release	CMake
Stretch	9	June 2017	3.7.2
Buster	10	July 2019	3.13.4
Bullseye	11	August 2021	3.18.4
Bookworm	12	June 2023	3.25.1

For the most accurate and up-to-date information, consider referring to the Debian package tracking system or running `apt-cache policy cmake` on your Debian system. The current version in the stable Debian 11 Bullseye as of 2023-05-16 is `3.18.4-2+deb11u1`.

Some Dependencies Of Neovim Project In Detail

- [LuaJIT](#)
- [Lua](#)
- [Luv](#)
- [gettext](#)
- [libiconv](#)
- [libtermkey](#)
- [libuv](#)
- [libvterm](#)
- [lua-compat](#)
- [tree-sitter](#)
- [unibilium](#)

4.2 Run Make

First, as root

```
mkdir -p /srv/{build,neovim}
chown $USER.$USER /srv/{build,neovim}
```

Then, as `$USER`

```
cd /srv/build
git clone https://github.com/neovim/neovim
cd neovim
# If you want to build the last stable version from the repository
# checkout 'stable'
git checkout stable
# Use gmake on BSD instead of make
make CMAKE_BUILD_TYPE=RelWithDebInfo CMAKE_INSTALL_PREFIX=/srv/neovim
```

The type `RelWithDebInfo` (“Release With Debug Info”) enables optimizations and add debug information so you can get a backtrace in the case of crashing. Do not add a `-j` flag if `ninja` is installed! The build will be in parallel automatically.

On a 4-core CPU from 2017, this process takes less than 1.5 minutes if dependencies need to be downloaded, and significantly less if dependencies are already provided by the package manager. Please note that the build system may encounter issues if run multiple times. If you encounter any issues, try downloading the repository again and running `cmake` on a fresh repository if the suggestion in the next paragraph do not work.

CMake is the main build system that caches in `build/CMakeCache.txt`. If you want to change something, like `CMAKE_BUILD_TYPE` or `CMAKE_INSTALL_PREFIX`, run `rm -rf build` first. This is also required when rebuilding Neovim after a Git commit adds or removes files (this is including from runtime). When you are not sure if somethings changed, run `make distclean`, that is basically a shortcut for `rm -rf build .deps`.

5 Build Package

On Debian you also have the possibility to create a package

```
cd /srv/build/neovim
cd build && cpack -G DEB
```

This will produce 2 files:

- `nvim-linux64.deb`
- `_CPack_Packages/Linux/DEB/nvim-linux64.deb`

To install as root use `dpkg -i nvim-linux64.deb`.

For faster build install: `Ninja`, `ccache`

5.1 Install

```
cd /srv/build/neovim
make install
```

This will give a binary in `/srv/neovim/bin` called `nvim`.

6 Tests

One easy test is to start neovim (`nvim`) and give the command `:checkhealth` to understand what is missing in terms of dependencies and configuration.

7 Links

- <https://neovim.io/>
- <https://github.com/neovim/neovim>
- <https://github.com/neovim/neovim/wiki/Building-Neovim>

8 History

Version	Date	Notes
0.1.0	2023-05-16	Initial release

9 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.