



[Documentation](#) | [Chat](#) | [Twitter](#)

CI **passing** coverage **78%** coverity **passed** clang analysis **11** PVS analysis **20**

in repositories **37** Debian Testing **0.7.2-3** downloads **1.9M**

➤ Neovim **latest/stable v0.7.0**

Neovim is a project that seeks to aggressively refactor Vim in order to:

- Simplify maintenance and encourage [contributions](#)
- Split the work between multiple developers
- Enable [advanced UIs](#) without modifications to the core
- Maximize [extensibility](#)

See the [Introduction](#) wiki page and [Roadmap](#) for more information.

Features

- Modern [GUIs](#)
- [API access](#) from any language including C/C++, C#, Clojure, D, Elixir, Go, Haskell, Java, JavaScript/Node.js, Julia, Lisp, Lua, Perl, Python, Racket, Ruby, Rust
- Embedded, scriptable [terminal emulator](#)
- Asynchronous [job control](#)
- [Shared data \(shada\)](#) among multiple editor instances
- [XDG base directories](#) support
- Compatible with most Vim plugins, including Ruby and Python plugins

See [:help nvim-features](#) for the full list!

Install from package

Pre-built packages for Windows, macOS, and Linux are found on the [Releases](#) page.

[Managed packages](#) are in [Homebrew](#), [Debian](#), [Ubuntu](#), [Fedora](#), [Arch Linux](#), [Void Linux](#), [Gentoo](#), and more!

Install from source

See the [Building Neovim](#) wiki page for details.

The build is CMake-based, but a Makefile is provided as a convenience. After installing the dependencies, run the following command.

```
make CMAKE_BUILD_TYPE=RelWithDebInfo
sudo make install
```

To install to a non-default location:

```
make CMAKE_INSTALL_PREFIX=/full/path/
make install
```

CMake hints for inspecting the build:

- `cmake --build build --target help` lists all build targets.
- `build/CMakeCache.txt` (or `cmake -LAH build/`) contains the resolved values of all CMake variables.
- `build/compile_commands.json` shows the full compiler invocations for each translation unit.

Transitioning from Vim

See [:help nvim-from-vim](#) for instructions.

Project layout

```
| ci/                build automation
| cmake/            build scripts
| runtime/          user plugins/docs
| src/nvim/         application source code (see src/nvim/README.md)
| | api/            API subsystem
| | eval/           VimL subsystem
| | event/          event-loop subsystem
| | generators/     code generation (pre-compilation)
| | lib/            generic data structures
| | lua/            Lua subsystem
| | msgpack_rpc/    RPC subsystem
| | os/             low-level platform code
| | └ tui/          built-in UI
| third-party/      CMake subproject to build dependencies
└ test/            tests (see test/README.md)
```

License

Neovim contributions since [b17d96](#) are licensed under the Apache 2.0 license, except for contributions copied from Vim (identified by the `vim-patch` token). See LICENSE for details.

Vim is Charityware. You can use and copy it as much as you like, but you are encouraged to make a donation for needy children in Uganda. Please see the kcc section of the vim docs or visit the ICCF web site, available at these URLs:

```
http://iccf-holland.org/
http://www.vim.org/iccf/
```

<http://www.iccf.nl/>

You can also sponsor the development of Vim. Vim sponsors can vote for features. The money goes to Uganda anyway.