*See **Quickstart** to get started immediately. Also note the section on **Not Applicable** patches.*

Neovim was forked from Vim 7.4.160; it is kept up-to-date with relevant Vim patches in order to avoid duplicate work. Run `vim-patch.sh` to see the status of Vim patches:

```
./scripts/vim-patch.sh -l
```

Everyone is welcome to send pull requests for relevant Vim patches, but some types of patches are not applicable.

Quickstart

1. Pull down the Neovim source:

   ```
   git clone https://github.com/neovim/neovim.git
   ```

2. Run `./scripts/vim-patch.sh -l` to see the list of missing Vim patches.
3. Choose a patch from the list (usually the *oldest* one), e.g. `8.0.0123`.
   - Check for open vim-patch PRs.
4. Run `./scripts/vim-patch.sh -p 8.0.0123`
5. Follow the instructions given by the script.

Notes

- It's strongly recommended to work on the *oldest* missing patch, because later patches might depend on the changes.
- Use `git log --grep` or `git log -G` to search the Nvim/Vim source history (even *deleted* code).
  E.g. to find `reset_option_was_set`:

  ```
  git log -p -G reset_option_was_set
  ```

- Pass `git log` options like `--grep` and `-G` to `./scripts/vim-patch.sh -L` to filter unmerged Vim patches
  E.g. to find `+quickfix` patches:

  ```
  ./scripts/vim-patch.sh -L --grep quickfix -- src
  ```

Pull requests

Note: **vim-patch.sh** automates these steps for you. Use it!

- Install gh or git-hub if you want to use `vim-patch.sh` to create PRs automatically
- The pull request *title* should include `vim-patch:8.x.xxxx` (no whitespace)
- The *commit message* should include:

  - A token indicating the Vim patch number, formatted as follows: `vim-patch:8.0.0123` (no whitespace)
  - A URL pointing to the Vim commit:
    * https://github.com/vim/vim/commit/c8020ee825b9d9196b1329c0e097424576fc9b3a
  - The original Vim commit message, including author

Reviewers: hint for reviewing `runtime/` patches

NA ("Not Applicable") patches

Many Vim patches are not applicable to Neovim. If you find NA patches, visit an open "version.c: update" pull request and mention the NA patches in a comment (please edit/update *one* comment, rather than adding a new comment for each patch).

If there are no open `version.c: update` pull requests, include NA patches in a commit message in the following format:

```
vim-patch:<version-or-commit>
vim-patch:<version-or-commit>
...
```

where `<version-or-commit>` is a valid Vim version (like `8.0.0123`) or commit-id (SHA). Each patch is on a separate line.

It is preferred to include NA patches by squashing it in applicable Vim patches, especially if the Vim patches are related. First line of the commit message should be from the applicable Vim patch.

```
./scripts/vim-patch -p <na-patch>
./scripts/vim-patch -p <na-patch>
...
./scripts/vim-patch -P <patch>
git rebase -i master
```

Example: https://github.com/neovim/neovim/commit/00f60c2ce78fc1280e93d5a36bc7b2267d5f4ac6

Types of "Not Applicable" Vim patches:

- Vim9script features, and anything related to `:scriptversion`. (Nvim supports Vimscript version 1 *only*.) Be aware that patches labelled `Vim9:` may still contain applicable fixes to other parts of the codebase, so these patch need to be checked individually.
- Updates to `testdir/Makefile` are usually NA because the Makefile implicitly finds all `test_*.vim` files.
- **Compiler warning fixes**: Neovim strives to have no warnings at all, and has a very different build system from Vim.
  - **Note:** Coverity fixes in Vim *are* relevant to Neovim.
- `*.proto` changes: Neovim autogenerates function prototypes

- **#ifdef tweaking**: For example, Vim decided to enable `FEAT_VISUAL` for all platforms — but Neovim already does that. Adding new `FEAT_` guards also isn't relevant to Neovim.
- **Legacy system support**: Fixes for legacy systems such as Amiga, OS/2 Xenix, Mac OS 9, Windows older than XP SP2, are not needed because they are not supported by Neovim.
  - NA files: src/Make*, src/testdir/Make_*
- `if_*.c` changes: `if_python.c` et. al. were removed.
- **term.c** changes: the Neovim TUI uses libtermkey to read terminal sequences; Vim's `term.c` was removed.
- `job` patches: incompatible API and implementation
  - NA files: src/channel*, src/job*, src/testdir/test_channel*, src/testdir/test_job*
- `:terminal` patches that modify NA files: incompatible API and implementation
  - NA files: src/terminal*, src/testdir/test_terminal*
- `defaults.vim` patches
- Most **GUI-related** changes: Neovim GUIs are implemented external to the core C codebase.
  - NA files: src/gui*, src/gvim*, src/GvimExt/, src/testdir/test_gui*
- `balloon` changes: Neovim does not support balloon feature
  - NA files: src/beval*, src/testdir/test_balloon*
- libvterm changes: Neovim does not vendor libvterm in `src/`.
- Screendump tests from `test_popupwin.vim`, `test_popupwin_textprop.vim`: https://github.com/neovim/neovim/pull/12741#issuecomment-704677141
- json changes: incompatible API https://github.com/neovim/neovim/pull/4131
  - NA files: src/json*, src/testdir/test_json.vim
- `test_restricted.vim` restricted mode is removed in https://github.com/neovim/neovim/pull/11996
- Many tests in `test_prompt_buffer.vim` require incompatible Vim features such as `channel`; they should still be included, but skipped
- non-runtime documentation: Moved to https://neovim.io/doc/, https://github.com/neovim/neovim/wiki#developers
  - NA files: Filelist, README, INSTALL*,
- Anything else might be relevant; err on the side of caution, and post an issue if you aren't sure.

version.c

The list of Vim patches in `src/nvim/version.c` is automatically updated based on the presence of `vim-patch:xxx` tokens in the Neovim git log.

- Don't update `src/nvim/version.c` yourself.
  - `scripts/vim-patch.sh -p` intentionally omits `version.c` to avoid merge conflicts and save time when porting a patch.
- The automation script (`scripts/vimpatch.lua`) only recognizes tokens

like `vim-patch:8.0.1206`, not `vim-patch:<hash>`.

Code differences

The following functions have been removed or deprecated in favor of newer alternatives. See the `memory.c` Doxygen page for more information.

Deprecated or removed	Replacement
`vim_free`	`xfree`
`VIM_CLEAR(&foo)`	`XFREE_CLEAR(foo)`
`malloc alloc lalloc lalloc_id ALLOC_ONE`	`xmalloc`
`calloc lalloc_clear`	`xcalloc`
`realloc vim_realloc`	`xrealloc`
`mch_memmove`	`memmove`
`vim_memset copy_chars copy_spaces`	`memset`
`vim_strncpy strncpy`	`xstrlcpy`
`vim_strcat strncat`	`xstrlcat`
`VIM_ISWHITE`	`ascii_iswhite`
`vim_isalpha`	`mb_isalpha`
`vim_islower vim_isupper`	`mb_islower mb_isupper`
`vim_tolower vim_toupper`	`mb_tolower mb_toupper`
`mb_ptr2len`	`utfc_ptr2len`
`mb_ptr2len_len`	`utfc_ptr2len_len`
`mb_char2len`	`utf_char2len`
`mb_char2bytes`	`utf_char2bytes`
`mb_ptr2cells`	`utf_ptr2cells`
`mb_ptr2cells_len`	`utf_ptr2cells_len`
`mb_char2cells`	`utf_char2cells`
`mb_off2cells`	`utf_off2cells`
`mb_ptr2char`	`utf_ptr2char`
`mb_head_off`	`utf_head_off`
`mb_lefthalve`	`grid_lefthalve`
`mb_fix_col`	`grid_fix_col`
`utf_off2cells`	`grid_off2cells`
`ml_get_curline`	`get_cursor_line_ptr`
`ml_get_cursor`	`get_cursor_pos_ptr`
`screen_char`	`ui_line`
`screen_line`	`grid_put_linebuf`
`screen_*` (most functions)	`grid_*`
`update_prepare`, `update_finish` #9484	removed; use `update_screen` only
`ARRAY_LENGTH`	`ARRAY_SIZE`
`vim_strsave_escape_csi`	`vim_strsave_escape_ks`
`vim_unescape_csi`	`vim_unescape_ks`

Make sure to note the difference between `utf_` and `utfc_` when replacing `mb_` functions. Also indirect call syntax `(*mb_ptr2len)(...)` should be replaced with an ordinary function call `utfc_ptr2len(...)`.

Data type	Format (Vim source)	Portable format (Nvim source)
`long long`	`"%lld"`	`"%" PRId64`
`size_t`	`"%ld"`	`"%zu"`

- See also: https://github.com/neovim/neovim/pull/1729#discussion_r22423779
- Vim's `ga_init2` was renamed to `ga_init` and the original `ga_init` is gone.
- "Old style" Vim tests (`src/testdir/*.in`) should be converted to Lua tests (see #1286 and #1328). See Checklist for migrating legacy tests.
  - However, please *do not* convert "new style" Vim tests (`src/testdir/*.vim`) to Lua. The "new style" Vim tests are faster than the old ones, and converting them takes time and effort better spent elsewhere. Just copy them to `src/nvim/testdir/*.vim`.
- Conditions that check `enc_utf8` or `has_mbyte` are obsolete (only the "true" case is applicable).
  - `enc_utf8` and `has_mbyte` macros were removed in https://github.com/neovim/neovim/pull/13293
- List management has changed in Neovim, see this wiki page.

Documentation differences

The following should be removed from all imported documentation, and not be used in new documentation:

- `{not in Vi}` : we don't care about compatibility with Vi (see `818f7ae`).
- `{Only when compiled with ...}` - the vast majority of features have been made non-optional (see Introduction)