

---

## 24.3 Vim

Many distributions of Vim ship with support for Scheme, which will mostly work for Racket. As of [version 7.3.518](#), vim detects files with the extension `.rkt` as having the scheme filetype. [Version 8.2.3368](#) added support for `.rkt` and `.rktl`.

In older versions, you can enable filetype detection of Racket files as Scheme with the following:

```
if has("autocmd")
  autocmd filetypedetect BufReadPost *.rkt,*.rktl,*.rktd set filetype=scheme
endif
```

If your vim supports the `ftdetect` system, in which case it's likely new enough to support Racket already, you can nevertheless put the following in `~/.vim/ftdetect/racket.vim`

(`~/.vimfiles/ftdetect/racket.vim` on MS-Windows; see `:help runtimepath`).

```
" :help ftdetect
" If you want to change the filetype only if one has not been set
autocmd BufRead,BufNewFile *.rkt,*.rktl,*.rktd setfiletype scheme
" If you always want to set this filetype
autocmd BufRead,BufNewFile *.rkt,*.rktl,*.rktd set filetype=scheme
```

---

### 24.3.1 Plugins

Alternatively, you can use a plugin such as

- [wlangstroth/vim-racket](#)
- [benknoble/vim-racket](#)

The major difference between the two is that the

benknoble/vim-racket fork supports more features out of the box and is updated more frequently.

to enable auto-detection, indentation, and syntax highlighting specifically for Racket files.

These plugins work by setting the `filetype` option based on the `#lang` line. For example:

- A file starting with `#lang racket` or `#lang racket/base` has `filetype` equal to `racket`.
- A file starting with `#lang scribble/base` or `#lang scribble/manual` has `filetype` equal to `scribble`.

Depending on which plugin you have, modifiers like `at-exp` may also be ignored, so that `#lang at-exp racket` is still a `filetype` of `racket`.

This approach is more flexible but may lead to more work. Since each `#lang` has its own `filetype`, options, syntax highlighting, and other features need to be configured for each `filetype`. This can be done via the standard `ftplugin` mechanism. See for example `:help ftplugin-overrule` and `:help ftplugin:` place your options for `<lang>` in `"~/.vim/after/ftplugin/<lang>.vim"` (`"$HOME/vimfiles/after/ftplugin/<lang>.vim"` on MS-Windows). Similarly, syntax files follow the standard mechanism documented in `:help syntax`.

Both plugins come with configuration for Racket (and possibly other `#langs`) as `ftplugins`. To enable them, use the `:filetype` command as documented in `:help :filetype`. You likely want to turn on `filetype` plugins (`:help :filetype-plugin-on`) and `filetype` indent plugins (`:help :filetype-indent-on`).

---

## 24.3.2 Indentation

You can enable indentation for Racket by setting both the `lisp` and

autoindent options in Vim. You will want to customize the buffer-local `lispwords` option to control how special forms are indented. See `:help 'lispwords'`. Both plugins mentioned in [Plugins](#) set this option for you.

However, the indentation can be limited and may not be as complete as what you can get in Emacs. You can also use Dorai Sitaram's [scmindent](#) for better indentation of Racket code. The instructions on how to use the indenter are available on the website.

---

### 24.3.3 Highlighting

The [Rainbow Parenthesis](#) script for vim can be useful for more visible parenthesis matching. Syntax highlighting for Scheme is shipped with vim on many platforms, which will work for the most part with Racket. The `vim-racket` script provides good default highlighting settings for you.

---

### 24.3.4 Structured Editing

The [Slimv](#) plugin has a `paredit` mode that works like `paredit` in Emacs. However, the plugin is not aware of Racket. You can either set vim to treat Racket as Scheme files or you can modify the `paredit` script to load on `".rkt"` files.

For a more vim-like set of key-mappings, pair either of

- [guns/vim-sexp](#)
- [benknoble/vim-sexp](#)

The `benknoble/vim-sexp` fork is slightly more modern `vimscript`.

with [tpope/vim-sexp-mappings-for-regular-people](#). The experience

is on par with `paredit`, but more comfortable for the fingers.

---

### 24.3.5 Scribble

Vim support for writing scribble documents is provided by

- [wilbowma/scribble.vim](#)
- [benknoble/scribble.vim](#)

Again, `benknoble/scribble.vim` is updated more frequently and is somewhat more modern.

---

### 24.3.6 Miscellaneous

If you are installing many vim plugins (not necessary specific to Racket), we recommend using a plugin that will make loading other plugins easier. There are many plugin managers.

[Pathogen](#) is one plugin that does this; using it, you can install new plugins by extracting them to subdirectories in the "bundle" folder of your personal Vim files ("`~/vim`" on Unix, "`$HOME/vimfiles`" on MS-Windows).

With newer vim versions, you can use the package system (`:help packages`).

One relatively up-to-date reference on the various managers is [What are the differences between the vim plugin managers?](#). The same site, [Vi & Vim](#) is a great place to get help from vimmers.