

---

## Webdrivers



Run Selenium tests more easily with automatic installation and updates for all supported webdrivers.

### Update: Future of this Project

With Google's new Chrome for Testing project, and Selenium's new Selenium Manager feature, what is required of this gem has changed.

If you can update to the latest version of Selenium (4.11+), please do so and stop requiring this gem. Provide feedback or raise issues to Selenium Project

If you cannot upgrade to Selenium 4.11, Webdrivers 5.3.0 will continue to support Ruby 2.6+ and Selenium 4.0 - 4.10

**Webdrivers 6.0** To provide support for Selenium 3 and Ruby < 2.6 a 6.0 version is planned. It requires:

\* Creating a `selenium-manager.gem` based off of <https://github.com/SeleniumHQ/selenium/pull/12429>

\* Re-implementing this gem to wrap `selenium-manager.gem` \* Ensuring compatible with older versions of Selenium & Ruby

If anyone would like to help get Webdrivers 6 working, please let us know.

### Description

`webdrivers` downloads drivers and directs Selenium to use them. Currently supports:

- chromedriver
- geckodriver
- IEDriverServer
- msedgedriver

Works on macOS, Linux, Windows, and Windows Subsystem for Linux (WSL) v1 and v2. And do see the browser and OS specific notes at the bottom.

### Usage

In your Gemfile:

```
1 gem 'webdrivers', '~> 5.0', require: false
```

---

In your project:

```
1 require 'webdrivers'
```

The drivers will now be automatically downloaded or updated when you launch a browser through Selenium.

### Specific Drivers

If you want webdrivers to only manage specific drivers you can specify one or more as follows:

```
1 require 'webdrivers/chromedriver'
2 require 'webdrivers/geckodriver'
3 require 'webdrivers/iedriver'
4 require 'webdrivers/edgedriver'
```

### Download Location

The default download location is `~/ .webdrivers` directory, and this is configurable:

```
1 Webdrivers.install_dir = '/webdrivers/install/dir'
```

Alternatively, you can define the path via the `WD_INSTALL_DIR` environment variable.

### Version Pinning

If you would like to use a specific (older or beta) version, you can specify it for each driver. Otherwise, the latest (stable) driver will be downloaded and passed to Selenium.

```
1 # Chrome
2 Webdrivers::Chromedriver.required_version = '2.46'
3
4 # Firefox
5 Webdrivers::Geckodriver.required_version = '0.23.0'
6
7 # Internet Explorer
8 Webdrivers::IEDriver.required_version = '3.14.0'
9
10 # Edge (Chromium)
11 Webdrivers::Edgedriver.required_version = '76.0.183.0'
```

You can explicitly trigger the update in your code, but this will happen automatically when the driver is initialized:

---

```
1 Webdrivers::Chromedriver.update
```

## Caching Drivers

You can set Webdrivers to only look for updates if the previous check was longer ago than a specified number of seconds.

```
1 Webdrivers.cache_time = 86_400 # Default: 86,400 Seconds (24 hours)
```

Alternatively, you can define this value via the `WD_CACHE_TIME` environment variable. **Only set one to avoid confusion.**

**Special exception for chromedriver and msedgedriver** Cache time will be respected as long as a driver binary exists and the major.minor.build versions of the browser and the driver match. For example, if you update Chrome or Edge to v76.0.123 and its driver is still at v76.0.100, `webdrivers` will ignore the cache time and update the driver to make sure you're using a compatible build version.

## Proxy

If there is a proxy between you and the Internet then you will need to configure the gem to use the proxy. You can do this by calling the `configure` method.

```
1 Webdrivers.configure do |config|
2   config.proxy_addr = 'myproxy_address.com'
3   config.proxy_port = '8080'
4   config.proxy_user = 'username'
5   config.proxy_pass = 'password'
6 end
```

## SSL\_connect errors

If you are getting an error like this (especially common on Windows):

```
SSL_connect returned=1 errno=0 state=SSLv3 read server certificate B:
certificate verify failed
```

Add the following to your Gemfile:

```
1 gem "net_http_ssl_fix"
```

Add the following to your code:

---

```
1 require 'net_http_ssl_fix'
```

Other solutions are documented on the RubyGems website.

## Rake tasks

Each driver has its own set of `rake` tasks (with `Railtie` support) that you can call once before executing the tests. These are especially useful if you're running tests in parallel and want to avoid performing an update check per thread.

If you are using Rails default configuration the `webdrivers` gem will only be loaded in the test group so you will need to specify the test environment when using the tasks:

```
1 RAILS_ENV=test rails webdrivers:chromedriver:update
```

If you are not using Rails, you'll need to load them into your Rakefile like this:

```
1 require 'webdrivers'
2 load 'webdrivers/Rakefile'
```

The full list of available tasks is:

```
1 $ bundle exec rake -T
2 rake webdrivers:chromedriver:remove          # Force remove
   chromedriver
3 rake webdrivers:chromedriver:update[version] # Remove and download
   updated chromedriver if necessary
4 rake webdrivers:chromedriver:version         # Print current
   chromedriver version
5 rake webdrivers:edgedriver:remove            # Force remove
   msedgedriver
6 rake webdrivers:edgedriver:update[version]   # Remove and download
   updated msedgedriver if necessary
7 rake webdrivers:edgedriver:version           # Print current
   msedgedriver version
8 rake webdrivers:geckodriver:remove           # Force remove
   geckodriver
9 rake webdrivers:geckodriver:update[version]  # Remove and download
   updated geckodriver if necessary
10 rake webdrivers:geckodriver:version          # Print current
   geckodriver version
11 rake webdrivers:iedriver:remove              # Force remove
   IEDriverServer
12 rake webdrivers:iedriver:update[version]     # Remove and download
   updated IEDriverServer if necessary
13 rake webdrivers:iedriver:version             # Print current
   IEDriverServer version
```

---

These tasks respect the `WD_INSTALL_DIR`, `WD_CACHE_TIME`, `WD_CHROME_PATH`, and `WD_EDGE_CHROME_PATH` environment variables, which can also be passed through the `rake` command:

```
1 $ bundle exec rake webdrivers:chromedriver:update[2.46] webdrivers:
  geckodriver:update[0.24.0] WD_CACHE_TIME=86_400 WD_INSTALL_DIR='
  my_dir'
2 2019-05-20 19:03:01 INFO Webdrivers Updated to chromedriver 2.46.628388
3 2019-05-20 19:03:04 INFO Webdrivers Updated to geckodriver 0.24.0
```

Please note that these tasks do not use any of the configurations from your project (code) and only respect the `ENV` variables and the version (optional) passed to the `rake` tasks.

## Logging

The logging level can be configured for debugging purpose:

```
1 Webdrivers.logger.level = :DEBUG
```

## Browser & OS Specific Notes

**Chrome/Chromium** The version of `chromedriver` will depend on the version of Chrome you are using it with:

- For versions  $\geq 70$ , the downloaded version of `chromedriver` will match the installed version of Google Chrome. More information [here](#).
- For versions  $\leq 69$ , `chromedriver` version 2.41 will be downloaded.
- For beta versions, you'll have to require the beta version of `chromedriver` using `Webdrivers::Chromedriver.required_version`.

The gem looks for the Chrome/Chromium version that `chromedriver` will use by default. You can override this behavior by providing a path to the browser binary you want to use:

```
1 Selenium::WebDriver::Chrome.path = '/chromium/install/path/to/binary'
```

Alternatively, you can define the path via the `WD_CHROME_PATH` environment variable.

This is also required if Google Chrome is not installed in its default location.

**Chrome on Heroku** Follow the specific instructions [here](#) if you're using `heroku-buildpack-google-chrome`.

---

**Microsoft Edge (Chromium)** Microsoft Edge (Chromium) support was added in v4.1.0. Notes from the Chrome/Chromium section apply to this browser as well.

Please note that `msedgedriver` requires `selenium-webdriver` v4.

**WSLv1 support** While WSLv1 is not designed to run headful applications like Chrome, it can run exes; as such when found to be running in WSL, `webdrivers` will use Chrome on the Windows filesystem.

It's recommended that you install the new PowerShell (PS7) to avoid a known issue with the console font being changed when calling the old PowerShell (PS5).

**WSLv2 support** Webdrivers will detect WSLv2 as running on Linux and use Chrome on the Linux filesystem.

WSLv2 doesn't support connecting to host ports out of the box, so it isn't possible to connect to Chromedriver on Windows without extra configurations, see: <https://github.com/microsoft/WSL/issues/4619>. The simplest way to use Chromedriver with WSLv2 is to run Chrome headless on Linux.

**Chrome and Edge on Apple M1 (arm64)** If you're switching from Intel to M1, you'll have to manually delete the existing Intel (`mac64`) driver before the M1 (`arm64`) build can be downloaded. Otherwise, you'll get an error: `Bad CPU type in executable - ~/.webdrivers/chromedriver (Errno::E086)`

## Wiki

Please see the wiki for solutions to commonly reported issues.

Join us in the `#webdrivers-gem` channel on Slack if you have any questions.

## License

The gem is available as open source under the terms of the MIT License, see LICENSE.txt for full details and copyright.

## Contributing

Bug reports and pull requests are welcome on GitHub. Run `bundle exec rake` and squash the commits in your PRs.