



Choose version

v2.6

▼

General

Release notes

Primary Commands

bundle install
bundle update
bundle cache
bundle exec
bundle config
bundle help

Utilities

bundle
bundle add
bundle binstubs
bundle check
bundle clean
bundle console
bundle doctor
bundle env
bundle fund
bundle gem
bundle info
bundle init
bundle inject
bundle issue
bundle licenses
bundle list
bundle lock
bundle open
bundle outdated
bundle platform
bundle plugin
bundle pristine
bundle remove
bundle show
bundle version
bundle viz
gemfile

bundle

`bundle` - Ruby Dependency Management

```
bundle COMMAND [--no-color] [--verbose] [ARGS]
```

Description

Bundler manages an `application's dependencies` through its entire life across many machines systematically and repeatably.

See [the bundler website](#) for information on getting started, and [Gemfile\(5\)](#) for more information on the `Gemfile` format.

Options

`--no-color`

Print all output without color

`--retry`, `-r`

Specify the number of times you wish to attempt network commands

`--verbose`, `-V`

Print out additional logging information

Bundle Commands

We divide `bundle` subcommands into primary commands and utilities:

Primary Commands

`bundle install(1)`

Install the gems specified by the `Gemfile` or `Gemfile.lock`

`bundle update(1)`

Update dependencies to their latest versions

`bundle cache(1)`

Package the `.gem` files required by your application into the `vendor/cache` directory (aliases: `bundle package`, `bundle pack`)

`bundle exec(1)`

Execute a script in the current bundle

`bundle config(1)`

Specify and read configuration options for Bundler

`bundle help(1)`

Display detailed help for each subcommand

Utilities

`bundle add(1)`

Add the named gem to the Gemfile and run `bundle install`

`bundle binstubs(1)`

Generate binstubs for executables in a gem

`bundle check(1)`

Determine whether the requirements for your application are installed and available to Bundler

`bundle show(1)`

Show the source location of a particular gem in the bundle

`bundle outdated(1)`

Show all of the outdated gems in the current bundle

`bundle console(1)` **(deprecated)**

Start an IRB session in the current bundle

`bundle open(1)`

Open an installed gem in the editor

`bundle lock(1)`

Generate a lockfile for your dependencies

`bundle viz(1)` **(deprecated)**

Generate a visual representation of your dependencies

`bundle init(1)`

Generate a simple `Gemfile`, placed in the current directory

`bundle gem(1)`

Create a simple gem, suitable for development with Bundler

`bundle platform(1)`

Display platform compatibility information

`bundle clean(1)`

Clean up unused gems in your Bundler directory

`bundle doctor(1)`

Display warnings about common problems

`bundle remove(1)`

Removes gems from the Gemfile

`bundle plugin(1)`

Manage Bundler plugins

`bundle version(1)`

Prints Bundler version information

Plugins

When running a command that isn't listed in PRIMARY COMMANDS or UTILITIES, Bundler will try to find an executable on your path named `bundler-<command>` and execute it, passing down any extra arguments to it.

Obsolete

These commands are obsolete and should no longer be used:

- `bundle inject(1)`

[Edit this document on GitHub](#) if you caught an error or noticed something was missing.