Bundler Ruby dependencies

CONCEPTS

Commit Gemfile and Gemfile.lock. The Gemfile holds your constraints, the lockfile pins the exact versions everyone gets.

Gemfile your constraints
↓ `install / update`
Gemfile.lock exact versions
↓ `install`
Installed gems system or vendor/bundle
↓ `exec / setup`
Your app load path set

PROJECT

`bundle init`
Write a starter Gemfile here

`bundle gem <name>`
Scaffold a new gem

INSTALL & REMOVE

`bundle install`
Install gems, write Gemfile.lock

`bundle add <gem>`
Add a gem to the Gemfile and install

`bundle add <gem> -v "~> 1.2" -g development`
Add with version and group

`bundle remove <gem>`
Remove a gem from the Gemfile

RUN

`bundle exec <cmd>`
Run a command with the bundle's gems

`bundle exec rake <task>`
Run a Rake task in the bundle

UPDATE & OUTDATED

`bundle update <gem>`
Upgrade one gem and what it needs

`bundle update --all`
Upgrade everything

`bundle outdated`
List gems with newer versions

`bundle update --patch`
Prefer patch releases only

INSPECT

`bundle list`
Every gem in the bundle with version

`bundle info <gem>`
Details about one gem

`bundle info <gem> --path`
Where a gem is installed

LOCKFILE

`bundle lock`
Resolve and write the lockfile only

`bundle lock --add-platform <platform>`
Add a platform to the lockfile

CONFIG & DEPLOY

`bundle config set --local <name> <value>`
Set for this project

`bundle config set --local path vendor/bundle`
Install gems into the project

`bundle config set --local deployment true`
Frozen lockfile + vendor/bundle

`bundle config set --local frozen true`
Fail if the lockfile would change

`bundle config set --local without 'development test'`
Skip groups on install

GEMFILE SYNTAX

`source "https://rubygems.org"`
Where gems come from

`gem "<name>"`
Depend on a gem

`gem "<name>", "~> 1.2"`
Pessimistic: >= 1.2, < 2.0

`gem "<name>", "~> 1.2.3"`
Pessimistic: >= 1.2.3, < 1.3

`gem "<name>", group: :test`
Put one gem in a group

`group :development, :test do ... end`
Group several gems