Rake Ruby task runner

CONCEPTS

A Rakefile is plain Ruby. A task runs its prerequisites first, once per run, then its block. Use do...end, not braces, so prerequisites bind correctly.

Rakefile Ruby DSL
↓ load
Task graph tasks + prereqs
↓ resolve
Prerequisites each runs once
↓ then
Task block Ruby code

Rake resolves prerequisites before running a task

Only tasks with a desc appear in rake -T. Use rake -T -A or rake -P to see the rest. Quote task arguments in zsh: rake 'build[x]'.

RUN

rake
Run the default task

rake *<task>* *<task>*
Run several tasks in order

rake *<ns>*:*<task>*
Run a namespaced task

rake -n *<task>*
Dry run: print tasks without executing

rake -j *<N>*
Run up to N tasks concurrently

rake -f *<file>*
Use another Rakefile

rake *<NAME>*=*<value>*
Set an ENV variable for the run

bundle exec rake *<task>*
Run with the bundled gem versions

INSPECT

rake -T
List tasks that have a desc

INSPECT (cont.)

rake -T *<pattern>*
List tasks matching pattern

rake -T -A
List all tasks, including undocumented

rake -P
List tasks with prerequisites

DEBUG AND OUTPUT

rake -t
Trace invoke and execute, with backtraces

rake --backtrace
Show full backtraces on errors

DEFINE TASKS

task :*<name>* **do ... end**
Define a task

task *<name>*: **%w[***<a>* *<b>***]**
Task with prerequisites

task default: :*<name>*
Set the default task

desc '*<text>***'**
Describe the next task for -T

namespace :*<ns>* **do ... end**
Group tasks under a prefix

task :*<name>***, [:***<a>***, :***<b>***] do |t, args|**
Task with named arguments

FILES AND HELPERS

file '*<out>***' ⇒ %w[***<src>***] do |t|**
Rebuild when a source is newer

sh '*<cmd>***'**
Run a shell command, fail the task on error

rake *<task>***[***<a>***,***<b>***]**
Pass arguments on the command line

rake 'name[a b,c]'
Quote arguments that contain spaces