Rake Tasks

David Grayson

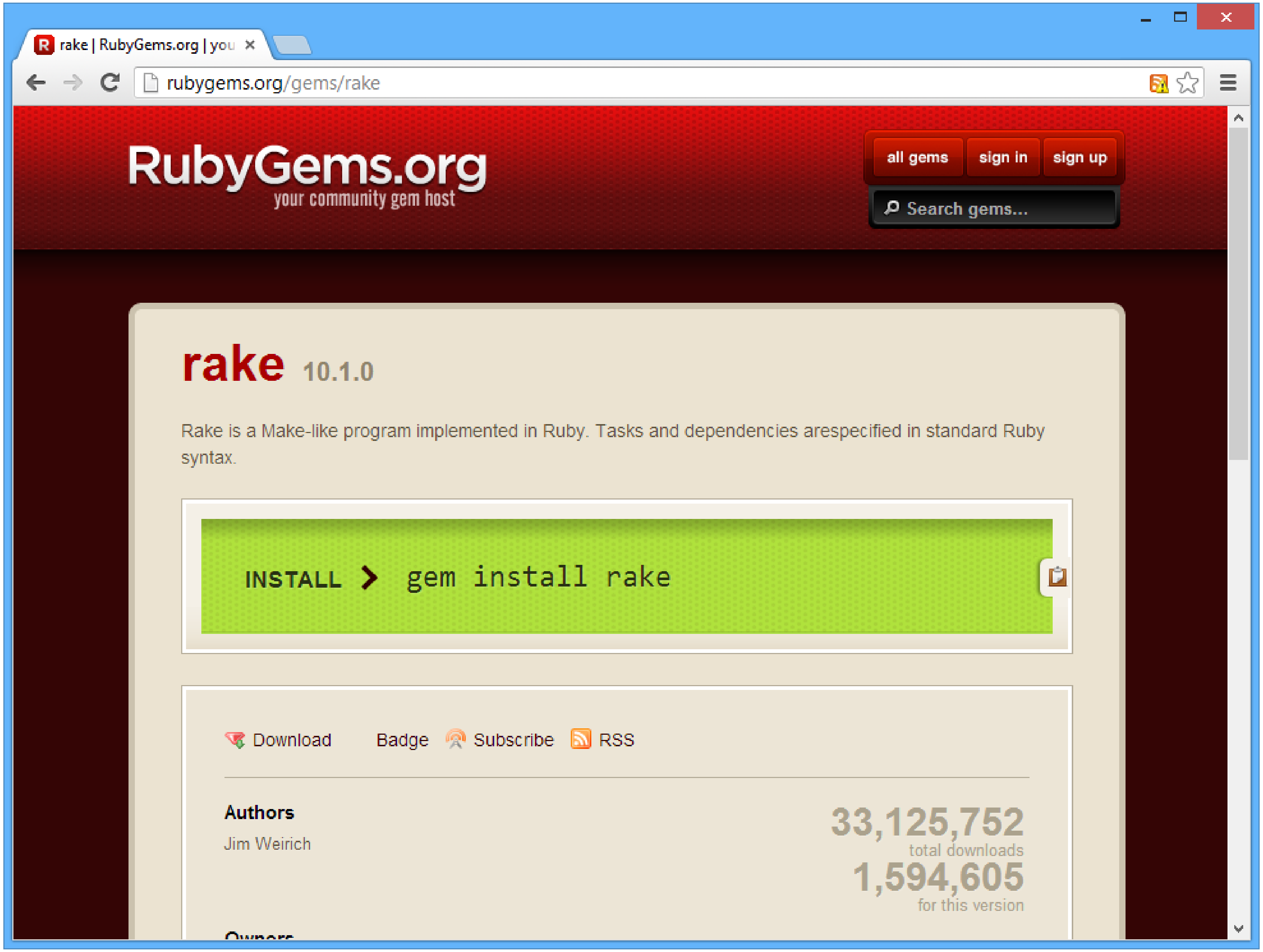
Las Vegas Ruby Meetup

2013-08-14

Two main types of Rails commands:

`rails`	`rake`
`rails new`	`rake db:migrate`
`rails server`	`rake spec`
`rails console`	`rake routes`
`rails generate`	`rake middleware`
	`rake assets:precompile`

Rake is a gem



rake 10.1.0

Rake is a Make-like program implemented in Ruby. Tasks and dependencies arespecified in standard Ruby syntax.

```
INSTALL > gem install rake
```

Authors

Jim Weirich

33,125,752 total downloads

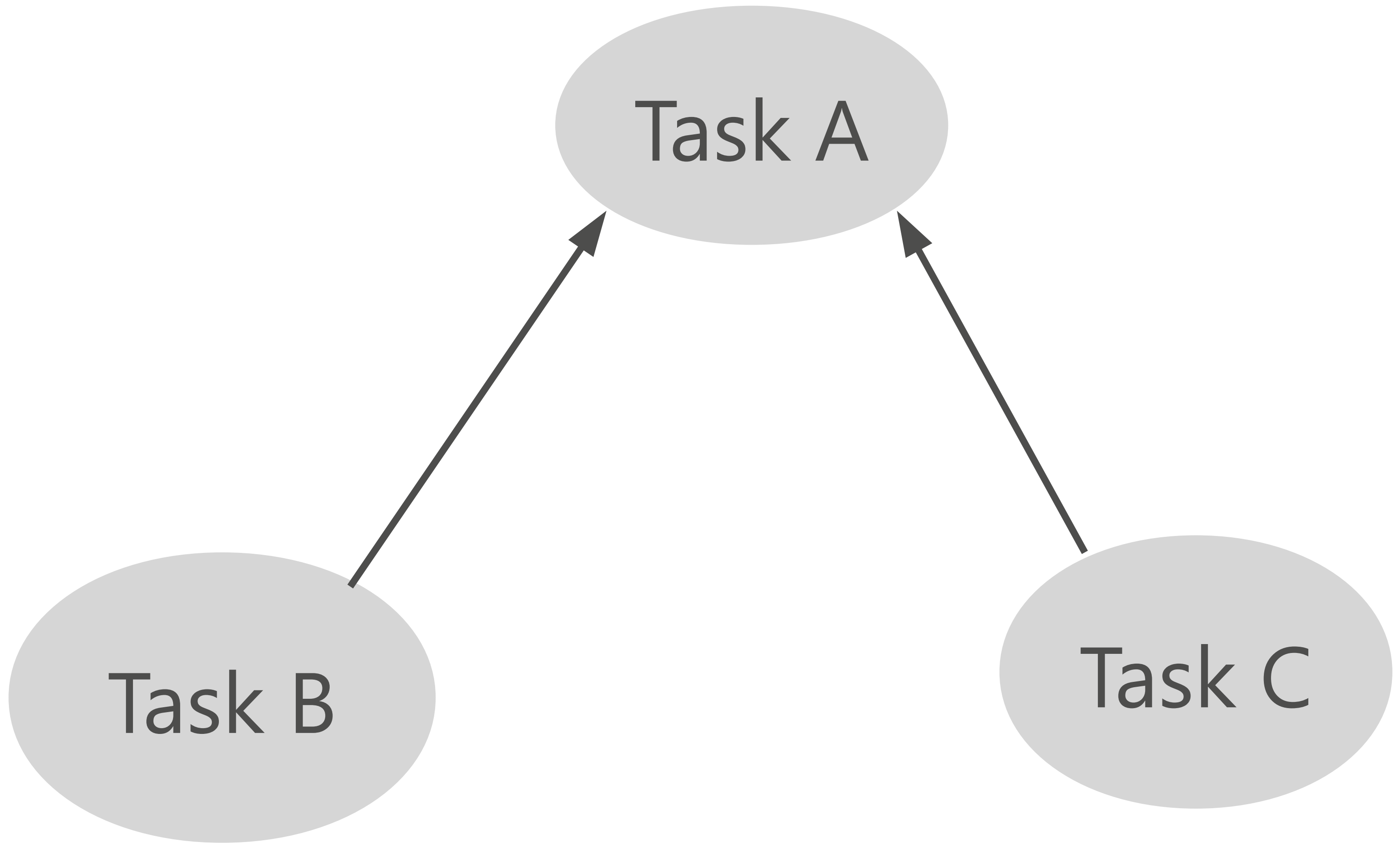
1,594,605 for this version

Rake helps you:

Record how to do standard tasks.

Record dependencies among tasks.

Visualizing dependencies (it's a graph!)



GNU Make

GNU Make: by Stuart Feldman in 1977.

Very popular for compiling programs.

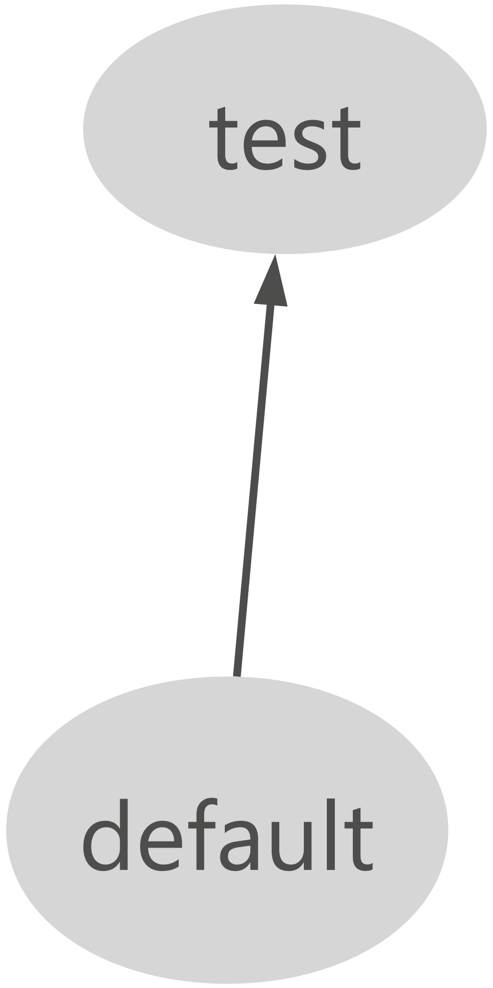
Makefiles have their own funny syntax:

```
define APP_template

APP_RELS := $$(patsubst %.c,%.rel, $$(wildcard apps/$(1)/*.c)) \
            $$(patsubst %.s,%.rel, $$(wildcard apps/$(1)/*.s))
APP_LIBS := $$(DEFAULT_LIBRARIES)
-include apps/$(1)/options.mk
APP_LIBS := $$(foreach lib, $$(APP_LIBS), libraries/lib/$$(lib))

RELs += $$(APP_RELS)
HEXs += apps/$(1)/$(1).hex

apps/$(1)/$(1).hex : $$(APP_RELS) $$(APP_LIBS)
    $$(LINK_COMMAND)
    $$(V)mv -f $$(@:%.hex=%.ihx) $$@

.PHONY : $(1)
$(1) : apps/$(1)/$(1).wxl

.PHONY : load_$(1)
load_$(1) : apps/$(1)/$(1).wxl
    $$(WIXELCMD) write $$< $$(S) -a

.PHONY : open_$(1)
open_$(1) : apps/$(1)/$(1).wxl
    $$(WIXELCONFIG) $$<
endef
```

Rakefile: just a Ruby program

```
task :default => [:test]

task :test do
  ruby "test/unittest.rb"
end
```

```
$ rake
$ rake test
```



Rake file tasks

- Rake looks at file timestamps.
- Actions only get executed if one of the prerequisite files is newer.

Rake file task example

```ruby
file "combo.txt" => ["part1.txt", "part2.txt"] do
  sh "cat part1.txt part2.txt > combo.txt"
end
```

Using Rake in Rails

```
rake db:migrate

rake spec

rake routes

rake middleware

rake assets:precompile
```

Specifying RAILS_ENV

Default is *development*.

```
RAILS_ENV=production rake db:migrate
```

Making your own Rails Rake task

In lib/tasks/foo.rake:

```
namespace :foo do
  desc "Write a nice description here."
  task :bar do
    puts "Foo bar"        # execute arbitrary ruby
    sh "echo 4 > file"    # run a shell command
    ruby "script.rb"      # invoke ruby
  end
end
```

```
$ rake foo:bar
```

http://railscasts.com/episodes/66-custom-rake-tasks

Accessing models in your task

```ruby
desc "Print the number of posts."
task :bar => [:environment] do
  puts Post.count
end
```

Enhancing existing Rake tasks

```
Rake::Task["test:prepare"].enhance ["clear_caches"]
```

```
Rake::Task["spec"].enhance do
  # your actions here
end
```

http://www.dan-manges.com/blog/modifying-rake-tasks

Example custom Rake tasks

- Clearing caches before running the tests
- Auto-generating button images
- Adding triggers to the database

Comments or questions?

Run `rake -T` to see rake tasks:

```
$ rake -T
rake about                  # List versions of all Rails frameworks and the environment
rake assets:clean           # Remove old compiled assets
rake assets:clobber         # Remove compiled assets
rake assets:environment     # Load asset compile environment
rake assets:precompile      # Compile all the assets named in config.assets.precompile
rake db:create              # Create the database from DATABASE_URL or config/database.yml for the current Rails.env ...
rake db:drop                # Drops the database using DATABASE_URL or the current Rails.env
rake db:fixtures:load       # Load fixtures into the current environment's database
rake db:migrate             # Migrate the database (options: VERSION=x, VERBOSE=false, SCOPE=blog)
rake db:migrate:status      # Display status of migrations
rake db:rollback            # Rolls the schema back to the previous version (specify steps w/ STEP=n)
rake db:schema:cache:clear  # Clear a db/schema_cache.dump file
rake db:schema:cache:dump   # Create a db/schema_cache.dump file
rake db:schema:dump         # Create a db/schema.rb file that can be portably used against any DB supported by AR
rake db:schema:load         # Load a schema.rb file into the database
rake db:seed                # Load the seed data from db/seeds.rb
rake db:setup               # Create the database, load the schema, and initialize with the seed data
rake db:structure:dump      # Dump the database structure to db/structure.sql
rake db:version             # Retrieves the current schema version number
rake doc:app                # Generate docs for the app -- also available doc:rails, doc:guides ...
rake log:clear              # Truncates all *.log files in log/ to zero bytes (specify which logs with LOGS=test,development)
rake middleware             # Prints out your Rack middleware stack
rake notes                  # Enumerate all annotations (use notes:optimize, :fixme, :todo for focus)
rake notes:custom           # Enumerate a custom annotation, specify with ANNOTATION=CUSTOM
rake rails:template         # Applies the template supplied by LOCATION=(/path/to/template) or URL
rake rails:update           # Update configs and some other initially generated files ...
rake routes                 # Print out all defined routes in match order, with names
rake secret                 # Generate a cryptographically secure secret key
rake stats                  # Report code statistics (KLOCs, etc) from the application
rake test                   # Runs test:units, test:functionals, test:integration together
rake test:all               # Run tests quickly by merging all types and not resetting db
rake test:all:db            # Run tests quickly, but also reset db
rake time:zones:all         # Displays all time zones, also available: time:zones:us, time:zones:local ...
rake tmp:clear              # Clear session, cache, and socket files from tmp/ ...
rake tmp:create             # Creates tmp directories for sessions, cache, sockets, and pids
```