Ruby Tools

Rake, Gem, Bundler

Rake

- It's like make, but for ruby.
- It's essentially a Domain Specific build language
- It's a tool for organizing tasks

Rakefile

- Anything you can do in ruby you can do in a rake file!
- No extension... It's just named Rakefile
- It lives in the root of your project
- To run it you just type 'rake' at the command line.

Rake Tasks

```
task :mac_and_cheese => [:boil_water, :buy_pasta, :buy_cheese] do
    puts "Making mac and cheese!"
end

task :boil_water do
    puts "I'm boiling some water!"
end

task :buy_pasta do
    puts "I'm buying pasta"
end

task :buy_cheese do
    puts "I'm buying cheese"
end
```

Tasks - Breaking it down

```
task :task_name => [:dep_1, :dep_2] do
  puts "Doing all the things!"
end
```

Special Tasks

```
task :default => [:dep_1, :dep_2] do
   puts "Doing all the things!"
end
```

This is the task that runs when you don't specify a task name to run rake.

For our earlier example we'd have to run it with one of the actions since there was no default

Real Example (with rspec)

```
require 'rake'
require 'rspec/core/rake_task'

RSpec::Core::RakeTask.new(:spec) do |task|
  task.rspec_opts = ['-c', '-fd']
end

task :roodi do
  system 'roodi lib/*.rb > metrics/roodi.txt'
end

task :flog do
  system 'flog -adg lib/*.rb> metrics/flog.txt'
end

task :flay do
  system 'flay lib/*.rb > metrics/flay.txt'
end

task :reek do
  system 'reek lib/*.rb > metrics/reek.txt'
end

task :default  => :spec
```

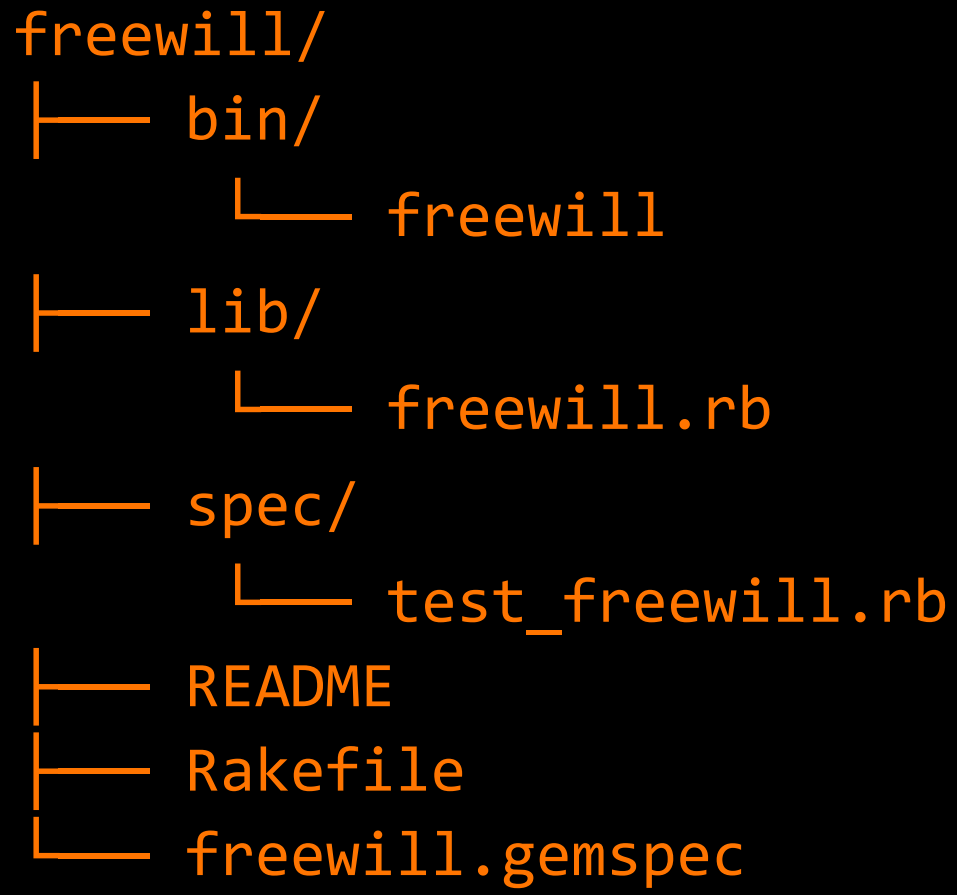
Gem

- Gems are essentially prepackaged ruby code (akin to a java jar)
- They can be libraries or full software packages
- Some examples: nokogiri, bundler, flog, rails, rake

Why do I need them?

- There are many gems that make our lives easier.
- Better use something widely used than rolling your own.

Gem Structure



Requiring Gems

- When you install a gem, you are basically putting the lib directory on your load path
- Any gems that are installed on a system are available to use within a project
- require 'gemname' at the top of your ruby file is you how tell ruby you want to make the gem code available to your code.

Gemspec

Gems have a file in them called 'Gemspec'

```
Gem::Specification.new do |s|
  s.name        = 'example'
  s.version     = '0.1.0'
  s.summary     = "This is an example!"
  s.description = "Longer explanation goes here"
  s.authors     = ["Ruby Coder"]
  s.email       = 'rubycoder@example.com'
  s.files       = ["lib/example.rb"]
  s.homepage    = 'http://rubygems.org/gems/example'
end
```

Bundler

- It is a gem that allows you to manage the dependencies of your project
- You place a file named 'Gemfile' in the root of your project
- To make sure you have all the gems you need you type 'bundle install' at the project root.

Example Gemfile

```
source :rubygems

gem "rails", ">= 3.0"
gem "rack"
gem "clearance", "0.9.0.rc9"

group :development, :test, :cucumber do
 gem "rspec-rails", "~> 2.0.0"
 gem "ruby-debug", :platforms => :mri_18
 gem "ruby-debug19", :platforms => :mri_19
end

group :test, :cucumber do
 gem "cucumber-rails"
 gem "factory_girl_rails"
 gem "capybara"
 gem "database_cleaner"
 gem "shoulda"
 gem "launchy"
 gem "akephalos", :git => "git://github.com/thoughtbot/akephalos.git"
 gem "thin"
```