

Jeff Kreeftmeijer

Archive Contact

Search

BE AWESOME: WRITE YOUR .GEMSPEC YOURSELF

2010-10-18

Jeweler is an awesome tool that gives you a generator to quickly set up the main skeleton for building a RubyGem. It has a bunch of rake tasks that make generating, creating, pushing and releasing your gem super easy without having to deal with the scary stuff. It's great for anyone that thinks they don't know or understand how to build a gem.

I'm taking Jeweler as an example here. There are a lot of gem builders out there that do the same thing, I just think Jeweler is the one most people know or use. This is *not* a Jeweler hate campaign, it's about gem builders in general.

Half a year ago, the gems I released were all built and maintained using Jeweler, so I put my gem's information in a `Rakefile` and generated my `.gemspec` — the file used to build the actual gem — from that. I published “**.gitignore your *.gemspec**”, arguing that the `.gemspec` shouldn't be checked into source control, since it was a generated file.

After getting a lot of responses and reading @wycats' **article**, I realized that generating the `.gemspec` file was silly, I completely changed my mind about the whole issue and wrote a follow up titled “**Don't put your *.gemspec in your Rakefile**”. I've been manually maintaining my `.gemspecs` ever since.

A lot of people are still using (tools like) Jeweler, which is fine. I would like to ask you to take a look at a **simple** `.gemspec` file sometime and ask yourself if this would more difficult to write and maintain than a `Rakefile` specifying the

Telerik
Develop
Mobile S
on a CM
Platform

Take advantage of
full spectrum
options with
CMS

ads via Cal

exact same thing. I feel like a lot of people are using Jeweler because everyone did it when they started and haven't looked back since.

Do we *need* this extra layer of abstraction? Doesn't generating .gemspecs make everything more cumbersome? Is the Rakefile the correct place for your gem specification?

The new kid in town

If you read @ryanbigg's great "**Developing a RubyGem using Bundler**" guide, you probably know that **Bundler** can also bootstrap your RubyGem using its `gem` command:

```
$ bundle gem my_gem
```

It'll give you everything you need — like the .gitignore, Rakefile, Gemfile and .gemspec — but it puts your gem specification right where it belongs: directly in your .gemspec. Also, it sets up a Gemfile that automatically gets its dependencies from the .gemspec file, like I explained **before**.

For a more detailed explanation, check out **Ryan's guide**.

Time to move on

While Jeweler and the like were wicked-cool back in the day, I think it's time to move on. Maintaining writing a .gemspec yourself isn't difficult at all and you don't need a tool to generate one for you, look:

```
require File.expand_path("../lib/your_gem/version", __FILE__)

Gem::Specification.new do |gem|
  gem.name      = 'your_gem'
  gem.version   = YourGem::VERSION
  gem.date     = Date.today.to_s

  gem.summary  = "an awesome gem"
  gem.description = "extended description"

  gem.authors  = ['Me Myself', 'One Other']
  gem.email    = 'me@example.com'
  gem.homepage = 'http://github.com/user/your_gem'
```

```
gem.add_dependency('rake')
gem.add_development_dependency('rspec', [">= 2.0.0"])

# ensure the gem is built out of versioned files
gem.files = Dir['Rakefile', '{bin,lib,man,test,spec}/**/*.', 'README*',
'LICENSE*'] & `git ls-files -z`.split("\0")
end
```

If you like to have a tool to set up a gem skeleton for you, I would suggest using Bundler. It's the cleanest gem builder I've come across so far.

What do you think? Do you prefer treating your .gemspec as a generated file? Do you use a gem builder? Let me know in the comments.

Disqus seems to be taking longer than usual. **Reload?**