

History of RubyGems and RDoc

Eric Hodel - drbrain@segment7.net

RubyGems

RubyGems Origins

- Jim Weirich, Rich Kilmer, Chad Fowler
- RubyConf 2003

My First Contribution

- RubyConf 2002
- Rich Kilmer borrowed my notepad

My First Contribution

- RubyConf 2002
- Rich Kilmer borrowed my notepad
- **FALSE!**

My First Contribution

- RubyConf 2001
- Rich Kilmer borrowed my notepad
- **FALSE!**
- Executable uninstall

RubyGems Releases

RubyGems 0.2.0

- `YAML format gems`
- `require_gem`
- `install, uninstall, list, rdoc`

RubyGems 0.3.0

- Library stubs
- Extensions
- Executables bin/your_app

RubyGems 0.7.0

- **Gem::Command refactor**

RubyGems 0.8.0

- tar/gzip format gems
- Custom require
- Platform support

RubyGems 0.8.3

- **YAML index growing pains**

RubyGems 0.8.5

- **YAML index growing pains**

RubyGems 0.8.7

- `gem install --include-dependencies`
- Replaced `'yes | gem install'`

RubyGems 0.8.11

- `gem install -y`
- Signed gems

RubyGems 0.9.0

- YAML index growing pains
 - Incremental update
- gemri

RubyGems 0.9.1

- `ruby -w clean`
- `tabs → spaces`
- `Gem::Installer` refactor
 - `firebrigade`

RubyGem 0.9.5

- Automatic platform gem installs
- Incremental update growing pains
 - Now with Marshal!

RubyGems 1.0.0

- `require_gem` removed

RubyGems 1.1.0

- Incremental update growing pains
 - Persistent connections
 - Latest index

RubyGems 1.2.0

- Modern index format
- Development dependencies

Best Feature

Gem::Command

- OptionParser
- One class per command
- Lazy loading

Gem::Command

- Argument grouping
- ARGV helpers

Gem::Command

- Argument sharing via Module
 - Local/remote options
 - Install/update options
 - Version/platform options

Gem::Command

- Integrated help
- Command completion
 - `gem i` → `gem install`

RubyGems Changes

Gem format

YAML gems

- Ruby self-install header
- YAML gemspec
- YAML manifest
- File data

tar/gzip format gems

- `your_gem-1.0.gem`
 - Files in `data.tar.gz`
 - YAML `gemspec` in `metadata.gz`

Signed Gems

- `your_gem-1.0.gem`
 - `data.tar.gz`
 - `data.tar.gz.sig`
 - `metadata.gz`
 - `metadata.gz.sig`

require + RubyGems

require_gem

- `require_gem 'your_gem'`
- Add gem paths to `$LOAD_PATH`
- Autorequire

Library stubs

- Autorequire → stub
- require 'your_gem/autorequire'
- require_gem 'your_gem'
 - Activate your_gem
 - require 'your_gem/autorequire'

custom_require

- Overrides Kernel#require
 - Real require
 - Activate gem (on failure)
 - Real require (again)

Built-in

- Ruby 1.9
 - `require 'your_gem/some_file'`

Remote Index

YAML

- One Big File
 - Every gemspec
 - Every detail

Incremental Update

- One Big File is too big
- One YAML file per gemspec
 - Fetch what's new

Incremental Update

- Switch to Marshal gemspecs
- Persistent connections

Incremental Update

- One Big File is too big
 - > 128MB to load

Modern Index

- Monolithic cache too big
- Download only what is needed
- Load gemspecs on-demand

Modern Index

- `gem install your_gem`
- `latest_specs.4.8.gz`
 - `[name, version, platform]`
- Download matching gemspec
 - `quick/Marshal.4.8/your_gem-1.0.gem`

Modern Index

- `gem install your_gem --version 1.5.18`
- `specs.4.8.gz`
 - `[name, version, platform]`
- Download matching gemspec

Index Evolution

Original

- `yaml.Z`
- `Cached locally`

Incremental

- `yaml.Z`
- `quick/index.rz`
- `quick/your_gem-1.0.gemspec.rz`

Marshal

- `Marshal.4.8.Z`
- `quick/index.rz`
- `quick/Marshal.4.8/your_gem-1.0.gemspec.rz`

Modern

- `specs.4.8.gz`
- `latest_specs.4.8.gz`
- `quick/Marshal.4.8/your_gem-1.0.gemspec.rz`
- One cache file per gemspec

gem install

Two Installers

- `Gem::Installer`
- `Gem::RemoteInstaller`

Gem::Installer

- Extract .gem file
- Build extensions
- Install executable stubs

Gem::RemoteInstaller

- Ask user about direct dependencies
 - Install dependency
- Install gem

Problems

- Not automatic
 - "Install required dependency?"
- Reinstall dependencies
 - Depth first

Two Installers

- `Gem::Installer`
- `Gem::DependencyInstaller`

Gem::DependencyInstaller

- Resolve dependencies
- Install gems

Platforms

Platform

- CPU, OS, OS version
 - x86-darwin-10
 - universal-darwin-10
 - universal-darwin
- Similar to Gem::Version

Building with Platforms

- Use rake-compiler
- Use Gem::Platform::CURRENT
 - Set required_ruby_version

Installing with Platforms

- Prefer gems matching your platform
- Many 1.8 only gems
- Fallback to RUBY platform
 - Build extension

TODO

gem mirror

- Slow
- Resource intensive
- Use `rubygems-mirror` by James Tucker

rubygems-mirror

- Persistent connections
- Parallel fetching
- Modern index

gem test

- `gem install -t`
- Nobody uses it
- No consistent testing API
- Try `rubygems-test` by Erik Hollensbe

rubygems-test

- Installs development dependencies
- rake test
- In development

Fancy Require

- Add look-up object to `$LOAD_PATH`
- `path_for`
 - File path
 - `true/false`
 - `nil`

Strictness

- Gem::Specification
 - Avoid junk
 - Better gem metadata
- Solution unknown

RDoc

RDoc Origins

**A set of tools to create
documentation in the
source for Ruby**

**"I'm surprised
(and terrified)
they lasted as long
as they did."**

First Contributions

- **Formatter for an IRC bot**
- **Gem ri paths**

RDoc Releases

RDoc 1.0.1

- Ruby 1.8.1

RDoc 2.0

- `TemplatePage` → `ERb`
- `RDoc` namespace
- `ri` cache

RDoc 2.1

- Metaprogrammed methods
- Ancestor lookup in ri
 - ri File#read displays IO#read

RDoc 2.2

- Better ri cache
- Interactive ri

RDoc 2.3

- Darkfish
- RDoc generator speedups
 - Threading
- rdoc/discover

RDoc 2.4

- Removed HTML and XML generators

RDoc 2.5

- New ri data format
- Removed threading

Components

Parsers

- Ruby Parser based on irb
- C parser based on regular expressions

RDoc::CodeObject

- Subclass per Ruby construct
 - File
 - Class/Module
 - Method
 - etc.

Generators

- HTML
- ri
- TAGS

RDoc::Markup

- Formatted plain text
- Block and inline styling
- Embed HTML

ri

- **Command-line documentation tool**
- **Plain text, ANSI, HTML output**

Best Feature

RDoc::CodeObject

- Describes a Ruby project
- Full introspection
- Intermediate data format

RDoc::CodeObject

- Rich interaction
- Easy to create
- Pretty-print

RDoc Changes

RDoc::Markup

Original

- Regexp parser for text blocks
 - Paragraph, list, verbatim
- Inline markup separate
 - `_emphasis_`

New

- Tokenizer
- Recursive descent parser
- Markup tree
- Inline markup still separate

Visitors

- RDoc text
 - Backspace text
 - ANSI text
- HTML
 - HTML + cross-reference links

ri

Original

- ri data directory per gem
- YAML file per method, class
- Many file lookups
- Separate formatters

First Pass

- YAML Object → Hash
- Cache file locations
- Complicated
- Too hard

Second Pass

- Marshal format
 - RDoc::CodeObject
 - Stores RDoc::Markup
- Reuse visitor code
- Index included

Generators

XML, CHM

- Removed from RDoc
- Unused
- Unmaintained

HTML

- TemplatePage
 - JSON-like Hash
 - Very simple, each, if
- RDoc::CodeObject → Hash
- Memory hog

HTML

- ERb
- RDoc::CodeObject → Hash

Darkfish

- ERb
- Walks RDoc::CodeObject tree
 - Memory friendly

Old ri

- `RDoc::CodeObject` → `Ri::Description`
- YAML format
- No index

New ri

- Marshal RDoc::CodeObject
 - RDoc::Markup comments
- Index per directory
 - Classes, methods, ancestors

TODO

Parsers

- Ruby Parser based on irb
- C parser based on regular expressions

Encoding

- Committed in trunk
- Needs battle-testing

Plugin Hooks

- Have
 - Generators
 - Directives (`# :api: public`)
- Need
 - Command line options

What I've Learned

API

Good API

- Clean separation
 - Do one thing
 - Do it well

Good API

- Rich interaction
 - Sort? #<=> and Comparable
 - Enumerate? #each and Enumerable
 - Hash key? #hash and #eql?

Good API

- Refactor aggressively
 - Shave off rough edges
 - Encapsulate state

Getting There

- Test everything
- Build something different
- Refactor

Battle Test

Battle Test

- Tests are good
- Idiot proof it!
- gauntlet

gauntlet

- Download every latest gem
- Convert to tar.gz
- Run a command against the gem

Success Problems

Hard to Tackle

- Few tests
- Many moving parts
- Is it supposed to work this way?

Familiarize Yourself

- Test it
- Learn it
- Improve it
- Replace it

Examine the Problem

- What's really needed?
- Try simple improvements
- Try simplifying
- Replace

Less is More

Plugins

- Less coordination
- Distributed development

Plugins

- More flexibility
- More ideas
- Less forking

Beware Plugins

- When to load
- What to hook
- Beware API lock-in

Questions?

Eric Hodel - drbrain@segment7.net

Links

- <http://github.com/rubygems/rubygems>
- <http://rubyforge.org/projects/rdoc>
- <http://github.com/rubygems/rubygems-mirror>
- <http://github.com/erikh/rubygems-test>