

Laying Tracks

How you can contribute to the
Ruby on Rails open source project

Josh Susser
Powerset

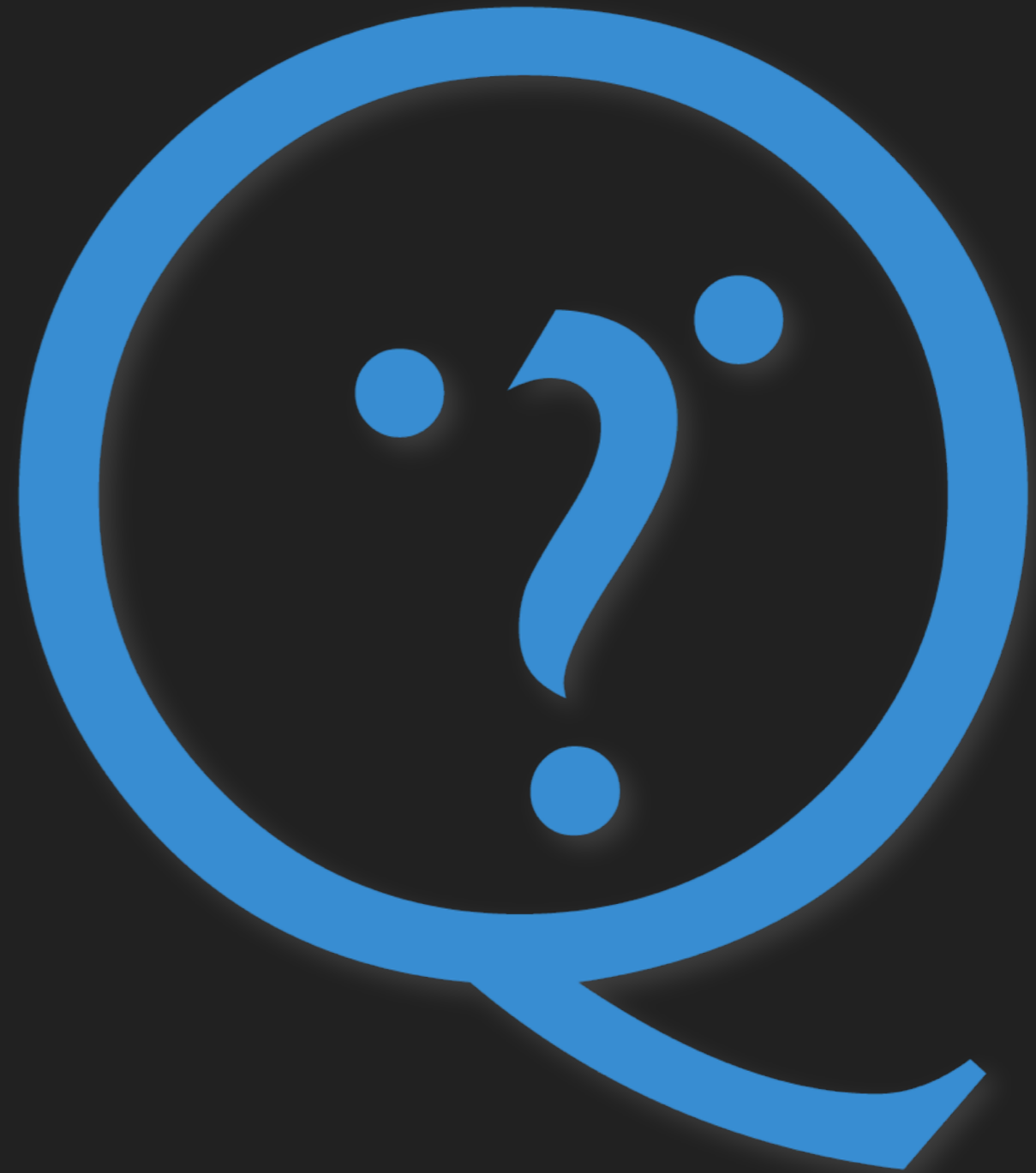
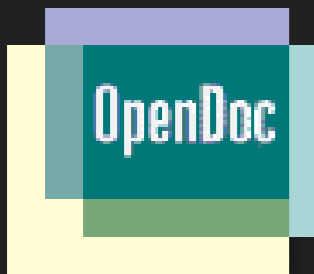
This is the true joy in life, the being used for a purpose recognized by yourself as a mighty one; the being a force of nature instead of a feverish, selfish little clod of ailments and grievances complaining that the world will not devote itself to making you happy.

I am of the opinion that my life belongs to the whole community, and as long as I live it is my privilege to do for it whatever I can.

I want to be thoroughly used up when I die, for the harder I work the more I live. I rejoice in life for its own sake. Life is no “brief candle” for me. It is a sort of splendid torch which I have got hold of for the moment, and I want to make it burn as brightly as possible before handing it on to future generations.

– George Bernard Shaw

Me



blog.hasmanythrough.com

Rails Hackfest 2007 Winners

24 days 263 participants

20 Winners

Who?

- 1 Jeremy McAnally (1953)
- 2 Dan Manges (1771)
- 3 Jarkko Laine (1128)
- 4 Scott Meade (1106)
- 5 Manfred Stienstra (1045)
- 6 Josh Susser (878)
- 7 Zack Chandler (864)
- 8 Ben Scofield (833)
- 9 Ryan Daigle (809)
- 10 Brian Donovan (794)

You

Why?

OSS != free

give
something
back

control your
own destiny

be cool

OSS != free

give

something

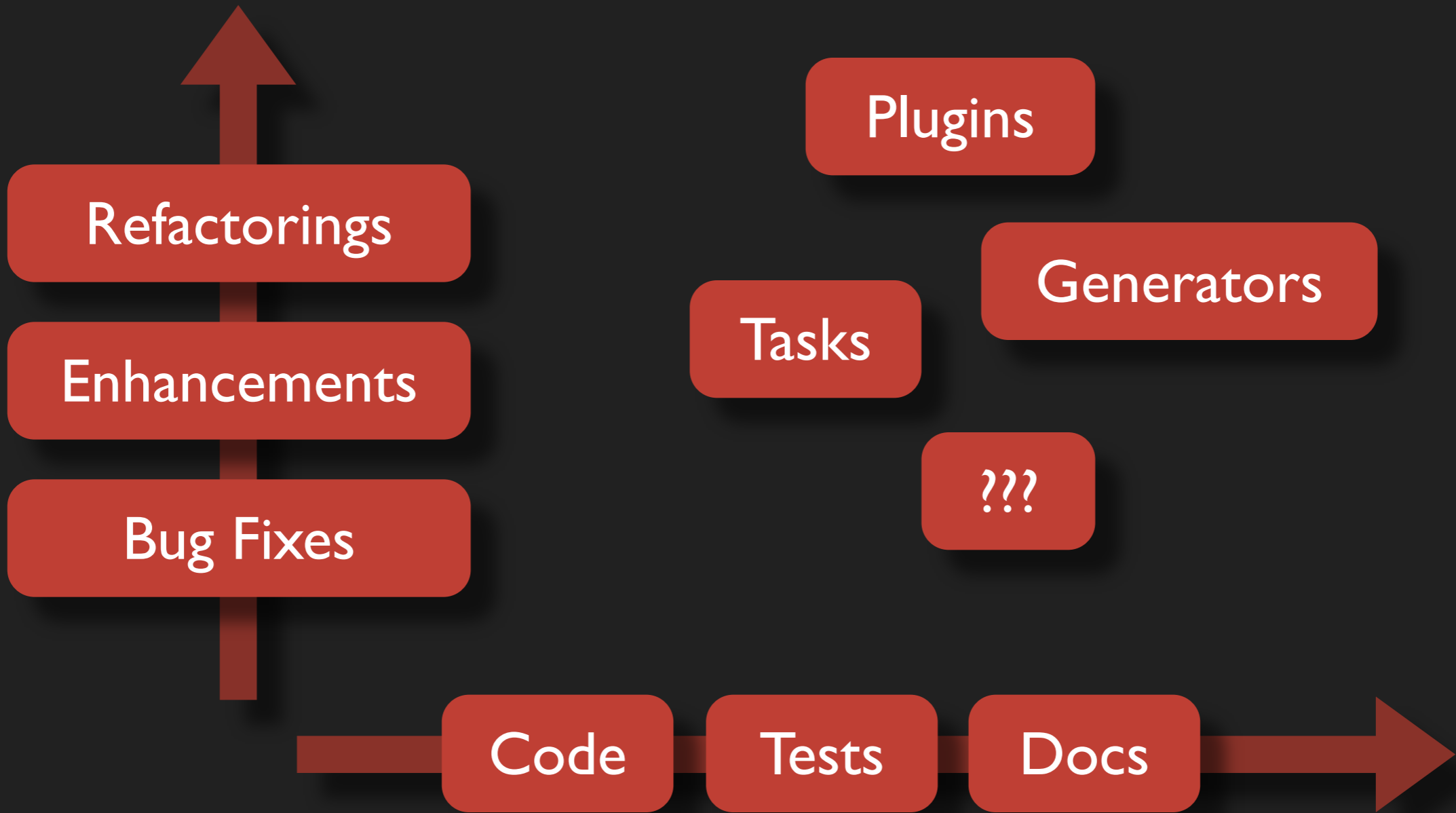
back

enlightened
self-interest

control your own destiny

be cool

What?



Where?

Trac

ticket tracker

dev.rubyonrails.org

Subversion (svn)

source code

dev.rubyonrails.org/svn/rails

email

rails core discussion

rubyonrails-core@googlegroups.com

IRC

discuss contributions

[#rails-contrib](#) on [irc.freenode.net](#)

How?

How

1. Set up Rails development environment
2. Code your little heart out
3. Generate patch file
4. Create ticket
5. Be patient...

Set up dev

1. Create a project just for working on Rails itself

```
$ rails contribute
```

2. Use Subversion to place sources in vendor/rails

```
$ cd vendor
```

```
$ svn co "http://svn.rubyonrails.org/rails/trunk" rails
```

3. Setup test database for ActiveRecord (and sometimes others)

- Read RUNNING_UNIT_TESTS file in module

- See test/connections/<database>/connection.rb

```
$ cd rails/activerecord
```

```
$ rake build_mysql_databases (for mysql, postgresql and frontbase)
```

4. Run tests before any changes as baseline

```
$ cd rails/activerecord
```

```
$ rake test_mysql
```

Implement!

- Start with the tests (TDD)
- Follow the style guidelines
- Don't forget the docs
- Make sure tests pass and docs build

```
$ cd contribute/vendor/rails/activerecord  
$ rake test_mysql
```

```
$ cd contribute  
$ rake doc:rails
```

Generate Patch

- Use Subversion `diff` command

```
$ cd contribute/vendor/rails
$ svn diff > well_named_patch.diff
```

- Examine diff file contents
TextMate has nice coloring

```
Index: activerecord/lib/active_record/associations/association.rb
-----
--- activerecord/lib/active_record/associations/association.rb
+++ activerecord/lib/active_record/associations/association.rb
@@ -1,10 +1,12 @@
 module ActiveRecord
   module Associations
     class AssociationProxy #:nodoc:
-      attr_reader :reflection
+      attr_reader :owner, :reflection
+      alias_method :proxy_owner, :owner
+      alias_method :proxy_reflection, :reflection
       alias_method :proxy_respond_to?, :respond_to?
       alias_method :proxy_extend, :extend
-      instance_methods.each { |m| undef_method m }
+      instance_methods.each { |m| undef_method m }

       def initialize(owner, reflection)
         @owner, @reflection = owner, reflection
```

Create a Ticket

1. Cut a hole in the box
2. Put your diff in the box
3. Profit!

(just kidding)

Create a Ticket

- Trac: **register** (unique name plz), log in
- <http://dev.rubyonrails.org/newticket>
- Summary: start with **[PATCH]**
- **upload** diff file, don't paste
- Describe problem and solution, show usage
- Keywords: doc, test, tiny, ...



Size Matters

- Small changes are easier to get accepted
 - Use “tiny” keyword for really small ones
- Got a big patch?
 - Discuss change on rails-core list first
 - Try out patch as plugin - makes it easy to check stability of change

sine qua non

code

tests

documentation

style

buy-in

Example

My First Patch - The Setup

```
class System < ActiveRecord::Base
  has_many :users, :extend => NameExtension
end
```



```
class System < ActiveRecord::Base
  has_many :users do
    include NameExtension
    include RoleExtension
  end
end
```



```
class System < ActiveRecord::Base
  has_many :users, :extend => [NameExtension, RoleExtension]
end
```

the code

```
def initialize(owner, reflection)
  @owner, @reflection = owner, reflection
-  proxy_extend(reflection.options[:extend]) if reflection.options[:extend]
+  Array(reflection.options[:extend]).each { |ext| proxy_extend(ext) }
  reset
end
```

the whole patch

Index: activerecord/test/associations_extensions_test.rb

```
=====
--- activerecord/test/associations_extensions_test.rb (revision 4201)
+++ activerecord/test/associations_extensions_test.rb (working copy)
@@ -18,6 +18,11 @@
  def test_named_extension_on_habtm
    assert_equal projects(:action_controller), developers(:david).projects_exten
  end
+
+ def test_named_two_extensions_on_habtm
+   assert_equal projects(:action_controller), developers(:david).projects_exten
+   assert_equal projects(:active_record), developers(:david).projects_extended_
+ end

  def test_marshallig_extensions
    david = developers(:david)
```

Index: activerecord/test/fixtures/developer.rb

```
=====
--- activerecord/test/fixtures/developer.rb (revision 4201)
+++ activerecord/test/fixtures/developer.rb (working copy)
@@ -4,6 +4,12 @@
  end
end
```

```
+module DeveloperProjectsAssociationExtension2
+ def find_least_recent
+   find(:first, :order => "id ASC")
+ end
+end
+
class Developer < ActiveRecord::Base
  has_and_belongs_to_many :projects do
    def find_most_recent
@@ -17,6 +23,12 @@
    :association_foreign_key => "project_id",
    :extend => DeveloperProjectsAssociationExtension

+ has_and_belongs_to_many :projects_extended_by_name_twice,
+   :class_name => "Project",
+   :join_table => "developers_projects",
+   :association_foreign_key => "project_id",
```

```
+   :extend => [DeveloperProjectsAssociationExtension, DeveloperProjectsAssoci
+
+   has_and_belongs_to_many :special_projects, :join_table => 'developers_projects

  validates_inclusion_of :salary, :in => 50000..200000
```

Index: activerecord/lib/active_record/associations/association_proxy.rb

```
=====
--- activerecord/lib/active_record/associations/association_proxy.rb (revision 4
+++ activerecord/lib/active_record/associations/association_proxy.rb (working co
@@ -8,7 +8,7 @@
```

```
def initialize(owner, reflection)
  @owner, @reflection = owner, reflection
-   proxy_extend(reflection.options[:extend]) if reflection.options[:extend]
+   Array(reflection.options[:extend]).each { |ext| proxy_extend(ext) }
  reset
end
```

Index: activerecord/lib/active_record/associations.rb

```
=====
--- activerecord/lib/active_record/associations.rb (revision 4201)
+++ activerecord/lib/active_record/associations.rb (working copy)
@@ -215,6 +215,14 @@
  #   has_many :people, :extend => FindOrCreateByNameExtension
  #   end
  #
+ # If you need to use multiple named extension modules, you can specify an ar
+ # In the case of name conflicts between methods in the modules, methods in m
+ # those earlier in the array. Example:
+ #
+ #   class Account < ActiveRecord::Base
+ #     has_many :people, :extend => [FindOrCreateByNameExtension, FindRecentE
+ #   end
+ #
  # === Association Join Models
  #
  # Has Many associations can be configured with the :through option to use an
```

Care and feeding

Feedback cycle

Create ticket

Get an email?

Trac updates

Respond!

Don't fight.
Play nice!

Waiting is

Discuss

rubyonrails-core@googlegroups.com

[#rails-contrib](https://irc.freenode.net/#rails-contrib) on irc.freenode.net

Where to begin

tests

open tickets

write a failing test

write a fix

PDI

“Please Do Investigate”

Let's get real

*Get used to
disappointment*

- Dread Pirate Roberts

popularize

pluginize

Community

rails list

rubyonrails-talk@googlegroups.com

IRC

#rubyonrails

irc.freenode.net

Trac

timeline RSS feed

Blog thyself

personal
relationships

Questions

Again and again some in the crowd wake up. They have no ground in the crowd, and they emerge according to much broader laws. They carry strange customs with them and demand room for bold gestures. The future speaks ruthlessly through them.

— Rainer Maria Rilke

<http://hasmanythrough.com/layingtracks>



Attribution-NonCommercial-ShareAlike 2.5