Redmine - Defect #7613

Generated test instances may share the same attribute value object

2011-02-12 13:30 - Etienne Massip

Status:	Closed	**Start date:**	2011-02-12
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Code cleanup/refactoring	**Estimated time:**	0.00 hour
Target version:	1.2.2		
Resolution:	Fixed	**Affected version:**	

Description

Actually, this test code will fail :

```
p1 = Project.generate!
p2 = Project.generate!

assert_not_equal p1.identifier, p2.identifier
```

That's because ObjectDaddy Project identifier generator actually returns the same object each time.

Thus, in source:trunk/test/exemplars/project_exemplar.rb :

```
  # Project#next_identifier is defined on Redmine
  def self.next_identifier_from_object_daddy
    @last_identifier ||= 'project-0000'
    @last_identifier.succ!
    @last_identifier
  end
```

Should be :

```
  # Project#next_identifier is defined on Redmine
  def self.next_identifier_from_object_daddy(last_identifier)
    last_identifier ||= 'project-0000'
    last_identifier.succ
  end
```

I'll post a patch soon.

Associated revisions

Revision 7458 - 2011-09-22 22:51 - Etienne Massip

Cleanup test exemplars and fix subsequent #generate calls generating same values (#7613).

Revision 7482 - 2011-09-23 19:23 - Etienne Massip

Merged r7458 from trunk (#7613).

History

#1 - 2011-02-12 14:08 - Etienne Massip

- File object_daddy_exemplars.patch added

- Status changed from New to Resolved

Here comes the patch.

All tests pass.

This issue was blocking me writting tests for #7456.

#2 - 2011-02-12 14:11 - Etienne Massip

- Subject changed from Generated test instances may share the same attribute value objects to Generated test instances may share the same attribute value object

#3 - 2011-02-12 15:03 - Jean-Baptiste Barth

Sorry I don't really understand how your patched version works. Does ObjectDaddy pass the last_<item> as an argument implictly ?

But I see the problem in actual version. Why not a @@last_identifier instead of @last_identifier ? I think the problem is here since it's a class method, an instance variable doesn't make much sense. And it should solve your first problem.

What do you think ?

#4 - 2011-02-12 17:00 - Etienne Massip

Jean-Baptiste Barth wrote:

> But I see the problem in actual version. Why not a @@last_identifier instead of @last_identifier ? I think the problem is here since it's a class method, an instance variable doesn't make much sense. And it should solve your first problem.

Nope, I've been a bit too much sybilline in my description :

The problem is that the generator returns the same object instance each time (@last_name), and that this instance is set by ObjectDaddy as the attribute value of the newly spawned Project instance.

That is to say :

```
p1.identifier.object_id == p2.identifier.object_id == @last_name.object_id
```

So, doing a @last_name.succ! when generating p2 actually also change the p1.identifier value, which is not the desired effect.

If @last_name.succ! was replaced by @last_name = @last_name.succ, the test would pass, but using a class variable would not change anything.

> Does ObjectDaddy pass the last_<item> as an argument implictly ?

Absolutly, it passes the previous value if generator method/block arity is 1 ; using this, IMHO, is the most elegant way to fix this issue.

#5 - 2011-02-13 02:21 - Jean-Baptiste Barth

Okay, I'll have a deeper look at it, thanks for the infos.

#6 - 2011-02-14 15:49 - Etienne Massip

- Status changed from Resolved to New

#7 - 2011-02-21 13:51 - Jean-Philippe Lang

I confirm the problem in current implementation but according to object_daddy documentation, there's a cleaner way to declare generators. The following patch fixes the problem for project generators:

```
Index: test/exemplars/project_exemplar.rb
===================================================================
--- test/exemplars/project_exemplar.rb    (revision 4892)
+++ test/exemplars/project_exemplar.rb    (working copy)
@@ -1,22 +1,9 @@
 class Project < ActiveRecord::Base
-  generator_for :name, :method => :next_name
-  generator_for :identifier, :method => :next_identifier_from_object_daddy
+  generator_for :name, :start => 'Project 0'
+  generator_for :identifier, :start => 'project-0000'
   generator_for :enabled_modules, :method => :all_modules
   generator_for :trackers, :method => :next_tracker

-  def self.next_name
-    @last_name ||= 'Project 0'
-    @last_name.succ!
-    @last_name
-  end
-
-  # Project#next_identifier is defined on Redmine
-  def self.next_identifier_from_object_daddy
-    @last_identifier ||= 'project-0000'
```

```
-    @last_identifier.succ!
-    @last_identifier
-  end
-
   def self.all_modules
     [].tap do |modules|
       Redmine::AccessControl.available_project_modules.each do |name|
```

#8 - 2011-02-21 14:02 - Etienne Massip

Yes, saw that too.

I was not at ease with so much change and I thought Eric was not ignorant of this way to do when he committed the exemplars in the first place, and that he chose to write full generator methods on purpose.

But really, no idea why he didn't do that.

#9 - 2011-02-21 15:05 - Etienne Massip

Also, there is possibility that the Issue.generate_for_project!() method in source:trunk/test/object_daddy_helpers.rb#L30 could have been written to supersede this issue, as it is called many times in a row in gantt_test.rb and that generate!() already deals with a block argument.

#10 - 2011-02-21 18:23 - Jean-Philippe Lang

I don't think so. I think the problem hasn't been noticed before.
Anyway, I don't see any reason not to clean up these generators.

#11 - 2011-02-21 18:25 - Etienne Massip

I do agree, the more clean, the better !

#12 - 2011-02-26 17:43 - Etienne Massip

- Target version set to Candidate for next minor release

#13 - 2011-09-17 14:03 - Etienne Massip

- Category changed from Core Plugins to Code cleanup/refactoring

#14 - 2011-09-23 08:47 - Etienne Massip

- Target version changed from Candidate for next minor release to 1.2.2

Still needs merging.

#15 - 2011-09-23 08:54 - Etienne Massip

- Status changed from New to Resolved

#16 - 2011-09-26 20:06 - Etienne Massip

- Resolution set to Fixed

#17 - 2011-11-11 12:29 - Jean-Philippe Lang

- Status changed from Resolved to Closed

Files

object_daddy_exemplars.patch	16.3 KB	2011-02-12	Etienne Massip