

Ruby trunk - Feature #4796

Coverage should be restartable

05/29/2011 12:41 PM - xaviershay (Xavier Shay)

Status:	Closed
Priority:	Normal
Assignee:	nagachika (Tomoyuki Chikanaga)
Target version:	1.9.3
Description	
I would like a way to be able to make the following test pass:	
<pre>require "test/unit" require "coverage" require 'tmpdir' class TestCoverage < Test::Unit::TestCase def test_restarting_coverage Dir.mktmpdir { tmp Dir.chdir(tmp) { File.open("test.rb", "w") do f f.puts <<-EOS def coverage_test_method puts :ok end EOS end } Coverage.start require tmp + '/test.rb' Coverage.result Coverage.start coverage_test_method assert_equal 1, Coverage.result.size } end end</pre>	
<p>The problem is that Coverage.start doesn't track any files loaded before it is called. This is probably desired behaviour so that stdlib files are not tracked, but it limits the usefulness of Coverage. Specifically, I am trying to collate coverage reports from workers in multiple processes. Also I want to associate coverages with specific tests (this test executed this code, etc...). This is very difficult without being able to restart.</p>	
<p>What would be involved in doing this? If you point me in the right direction I can perhaps have a go.</p>	

Associated revisions

Revision ea8b9163 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32404 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 32404 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

Revision 32404 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

Revision 32404 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

Revision 32404 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

Revision 32404 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

Revision 32404 - 07/04/2011 03:55 PM - nagachika (Tomoyuki Chikanaga)

- ext/coverage/coverage.c: resurrect r32071 + add GC guard for rb_coverages. [ruby-core:37352] [Bug #4927] [ruby-core:36539] [Feature #4796]
- test/coverage/test_coverage.rb resurrect r32071.

History

#1 - 05/29/2011 01:51 PM - mame (Yusuke Endoh)

- Category set to lib
- Status changed from Open to Assigned
- Assignee set to mame (Yusuke Endoh)
- Target version set to 1.9.3

Hello.

As a maintainer of ext/coverage, I like to approve this feature unless there is objection. But I'd like to confirm some points.

2011/5/29 Xavier Shay xavier-list@rhn.net:

The problem is that Coverage.start doesn't track any files loaded before it is called. This is probably desired behaviour so that stdlib files are not tracked, but it limits the usefulness of Coverage.

First of all, it is difficult to track files loaded before start' is first called. This is because the VM compiler inserts instructions to measure coverage after start' is first called. Since the mechanism have considerable overhead, it cannot be enabled by default.

You are talking about second (or later) calls to Coverage.start, right?

Specifically, I am trying to collate coverage reports from workers in multiple processes.

I cannot understand why the feature is needed for this use case. Could you elaborate this?

Also I want to associate coverages with specific tests (this test executed this code, etc...). This is very difficult without being able to restart.

I can understand this.

Here is a patch.

```
diff --git a/ext/coverage/coverage.c b/ext/coverage/coverage.c
index 29ac709..3a26aaa 100644
--- a/ext/coverage/coverage.c
+++ b/ext/coverage/coverage.c
@@ -11,6 +11,8 @@
#include "ruby.h"
#include "vm_core.h"
```

```
+static VALUE rb_coverages = Qundef;
+
+/*
```

- call-seq:
- Coverage.start => nil @@ -21,19 +23,25 @@ static VALUE rb_coverage_start(VALUE klass) { if (!RTEST(rb_get_coverages())) {
 - VALUE coverages = rb_hash_new();
 - RBASIC(coverages)->klass = 0;
 - rb_set_coverages(coverages);
 - if (rb_coverages == Qundef) {
 - rb_coverages = rb_hash_new();
 - RBASIC(rb_coverages)->klass = 0;
 - }
 - rb_set_coverages(rb_coverages); } return Qnil; }

```
static int
-coverage_result_i(st_data_t key, st_data_t val, st_data_t dummy)
+coverage_result_i(st_data_t key, st_data_t val, st_data_t h)
{
```

- VALUE path = (VALUE)key; VALUE coverage = (VALUE)val;
- RBASIC(coverage)->klass = rb_cArray;
- VALUE coverages = (VALUE)h;
- coverage = rb_ary_dup(coverage);
- rb_ary_clear((VALUE)val); rb_ary_freeze(coverage);
- rb_hash_aset(coverages, path, coverage); return ST_CONTINUE; }

```

@@ -48,14 +56,14 @@ static VALUE
rb_coverage_result(VALUE klass)
{
VALUE coverages = rb_get_coverages();

  • VALUE ncoverages = rb_hash_new(); if (!RTEST(coverages)) { rb_raise(rb_eRuntimeError, "coverage measurement is not enabled"); }
  • RBASIC(coverages)->klass = rb_cHash;
  • st_foreach(RHASH_TBL(coverages), coverage_result_i, 0);
  • rb_hash_freeze(coverages);
  • st_foreach(RHASH_TBL(coverages), coverage_result_i, ncoverages);
  • rb_hash_freeze(ncoverages); rb_reset_coverages();
  • return coverages;
  • return ncoverages; }

/* Coverage provides coverage measurement feature for Ruby.

--
Yusuke Endoh mame@tsg.ne.jp

```

#2 - 05/29/2011 02:37 PM - xaviershay (Xavier Shay)

You are talking about second (or later) calls to Coverage.start, right?
Yes, that would be fine.

I cannot understand why the feature is needed for this use case.
Could you elaborate this?

It's not required, it just would of made my life easier. I was trying to integrate with the parallel and SimpleCov gems, and ended up with a work around which involves conditionally hooking at_exit and a file lock, which is a bit of a mess.

Here is a patch.
I will try it out and report back. Thanks!

#3 - 06/05/2011 03:15 PM - xaviershay (Xavier Shay)

I tried this patch and it works as expected. Here is a patch for the test case, slightly cleaned up:

```

diff --git a/test/coverage/test_coverage.rb b/test/coverage/test_coverage.rb
index ace49d3..56966b1 100644
--- a/test/coverage/test_coverage.rb
+++ b/test/coverage/test_coverage.rb
@@ -1,5 +1,6 @@
require "test/unit"
require "coverage"
+require "tmpdir"

class TestCoverage < Test::Unit::TestCase
def test_result_without_start
@@ -14,4 +15,29 @@ class TestCoverage < Test::Unit::TestCase
assert_kind_of(Array, val)
end
end
+

```

- def test_restarting_coverage
- loaded_features = \$.dup +
- Dir.mktmpdir {|tmp|
- Dir.chdir(tmp) {
- File.open("test.rb", "w") do |f|
- f.puts <<-EOS
- def coverage_test_method
- :ok
- end
- EOS
- end +
- Coverage.start
- require tmp + '/test.rb'
- Coverage.result
- Coverage.start
- coverage_test_method

- assert_equal 1, Coverage.result.size
- }
- }
- ensure
- \$.replace loaded_features
- end end

#4 - 06/14/2011 01:14 AM - mame (Yusuke Endoh)

- Status changed from Assigned to Closed

Sorry for late action. I've committed my implementation and your test patch. Thank you!

--
Yusuke Endoh mame@tsg.ne.jp

#5 - 06/26/2011 02:58 PM - mame (Yusuke Endoh)

- Status changed from Closed to Open
- Assignee deleted (mame (Yusuke Endoh))
- Priority changed from Normal to 3
- Target version changed from 1.9.3 to 2.0.0

Unfortunately this patch seems to cause SEGV on os x. [#4927](#)
I cannot fix it because I have no os x. So I can't help but revert it.

Anyone who wants this feature, please fix it by yourself or please give me a Mac :-)

--
Yusuke Endoh mame@tsg.ne.jp

#6 - 07/05/2011 12:47 AM - nagachika (Tomoyuki Chikanaga)

- Status changed from Open to Assigned
- Assignee set to nagachika (Tomoyuki Chikanaga)
- Target version changed from 2.0.0 to 1.9.3

Sorry, I don't have extra Mac. But I can show a patch.
I'll commit it.

#7 - 07/05/2011 12:55 AM - nagachika (Tomoyuki Chikanaga)

- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset [r32404](#).
Xavier, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- ext/coverage/coverage.c: resurrect [r32071](#) + add GC guard for rb_coverages. [ruby-core:37352] [Bug [#4927](#)] [ruby-core:36539] [Feature [#4796](#)]
 - test/coverage/test_coverage.rb resurrect [r32071](#).

#8 - 07/05/2011 12:55 AM - mame (Yusuke Endoh)

That's the best! Thank you nagachika!

--
Yusuke Endoh mame@tsg.ne.jp