

## Ruby - Feature #6130

### inspect using to\_s is pain

03/12/2012 08:21 AM - trans (Thomas Sawyer)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	mame (Yusuke Endoh)	
<b>Target version:</b>	2.0.0	
<b>Description</b>		
Every time I define #to_s on a class I have to remember to also redefine #inspect. It's annoying and has led me to waste hours debugging b/c the errors that can occur from it often seem strange and unrelated.		
I think #inspect should have an independent definition that outputs the class name, object_id and when possible instance variable settings, and should have nothing to do with #to_s by default. We developers can always alias it as such if it is appropriate.		
The only exception should be for classes that have literal representation in Ruby, such as Array, String, Hash, etc. In those cases the #inspect should give the literal representation.		
<b>Related issues:</b>		
Is duplicate of Ruby - Bug #4453: Overriding #to_s changes #inspect		<b>Closed</b> <b>03/01/2011</b>

#### Associated revisions

##### Revision 795c29dc - 08/14/2012 10:20 AM - Eregon (Benoit Daloze)

Kernel#inspect: improve consistency and do not call #to\_s.

A class can now benefit from the nice default #inspect even if it defines #to\_s. Also, there is no more unexpected change in #inspect result. Internal structures have been adapted so they don't rely on the removed behavior (#inspect calling overridden #to\_s).

- object.c (rb\_obj\_inspect): Kernel#inspect: do not call #to\_s.
- test/ruby/test\_object.rb (test\_inspect): add tests for Kernel#inspect.
- bignum.c, io.c, numeric.c, object.c, proc.c, vm.c (Init\_\*): alias #inspect to #to\_s where it was expected.  
[ruby-core:43238][Feature #6130]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36699 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision e1aa72d4 - 08/14/2012 10:21 AM - Eregon (Benoit Daloze)

update PP with recent Kernel#inspect change. Patch by Yusuke Endoh.

- lib/pp.rb (class PP): do not call #to\_s anymore, as #inspect no more does.
- test/test\_pp.rb (class PPInspectTest): remove related assertion.  
[ruby-core:43238][Feature #6130]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36700 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 780e7d09 - 08/15/2012 01:09 AM - naruse (Yui NARUSE)

Revert r36699 and r36700. [Feature #6130]

Revert "Kernel#inspect: improve consistency and do not call #to\_s."  
Revert "update PP with recent Kernel#inspect change. Patch by Yusuke Endoh."

r36699 cause test-all failure on test/drb/test\_drb.rb and test/drb/test\_drbssl.rb. Run test-all before commit.

Moreover its ChangeLog formst is wrong: see CommitterHowto  
<https://bugs.ruby-lang.org/projects/ruby/wiki/CommitterHowto#ChangeLog>

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36701 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision fd7dc23d - 08/15/2012 11:50 AM - Eregon (Benoit Daloze)

Kernel#inspect: improve consistency and do not call #to\_s.

- object.c (rb\_obj\_inspect): Kernel#inspect: do not call #to\_s. A class can now benefit from the nice default #inspect even if it defines #to\_s. Also, there is no more unexpected change in #inspect result.
- NEWS: Add note about the change.
- bignum.c, io.c, numeric.c, object.c, proc.c, vm.c (Init\_\*): Adapt internal structures (by aliasing #inspect to #to\_s) so they don't rely on the removed behavior (#inspect calling overridden #to\_s).
- test/ruby/test\_object.rb (test\_inspect): add tests for Kernel#inspect.
- lib/pp.rb (class PP): do not call #to\_s anymore, as #inspect no more does (mame).
- test/test\_pp.rb (class PPInspectTest): remove related assertion (mame). [ruby-core:43238][Feature #6130]
- test/drb/drbtest.rb (DRbCore#teardown, DRbAry#teardown): adapt DRb tests with the new change (shirosaki). [ruby-core:47182][Bug #6866]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36709 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

---

### #1 - 03/30/2012 02:00 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned
- Assignee set to matz (Yukihiko Matsumoto)

Thomas, you meant:

```
class Foo
  def to_s; "foo"; end
end
x = Foo.new
```

p "==" #{ x } == " #=> "==" foo ==", as you want

p x #=> "foo", but you want: [#Foo:0xXXXX](#)

, right? If so I agree, though it may be difficult in 2.0.

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

### #2 - 03/30/2012 02:15 AM - trans (Thomas Sawyer)

Yep.

### #3 - 03/31/2012 12:28 AM - Eregon (Benoit Daloze)

mame (Yusuke Endoh) wrote:

If so I agree, though it may be difficult in 2.0.

I also agree, default #inspect is nice and should not be overridden by #to\_s.

If not possible for 2.0, I think it should still be considered later.

### #4 - 03/31/2012 12:35 AM - matz (Yukihiko Matsumoto)

OK, I accept. The time for it would be upto the maintainer.

Matz.

### #5 - 03/31/2012 08:56 AM - mame (Yusuke Endoh)

- Assignee changed from matz (Yukihiko Matsumoto) to mame (Yusuke Endoh)

That's great to hear!

As 2.0 release manager, I'm positive to import this change in 2.0.

I think it does not violate 2.0 compatibility policy; the format of a return string of #inspect is not specially specified. This changes only the format.

However, it should be postponed to 3.0 if it affects any actual existing products. Please let me if you know such a case.

I'd like to hear your opinion,

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#6 - 03/31/2012 08:58 AM - mame (Yusuke Endoh)**

Please let me if you know such a case.  
Oops. Please let me know such a case if you know.

**#7 - 04/02/2012 09:42 PM - mame (Yusuke Endoh)**

I'm not sure if it is trivial to fix this issue.  
Could anyone please create a patch and study the behavior?

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#8 - 04/10/2012 10:31 PM - Eregon (Benoit Daloze)**

mame (Yusuke Endoh) wrote:

I'm not sure if it is trivial to fix this issue.  
Could anyone please create a patch and study the behavior?

I've been working on this and wondered what to do if there is no instance variable. Should it call (dynamically) `#to_s?` or always use `Kernel#to_s` which gives the class name and pointer (which would be called anyway if `#to_s` was not overridden) ?

That's the first solution, use the nice `#inspect` output whenever there are instance variables.  
Another, lighter way would be to use `#to_s` for internal structures (and `T_DATA`, so classes defined in C extensions), but never for other types (all classes defined in Ruby).

What do you think?

**#9 - 04/11/2012 07:56 AM - Eregon (Benoit Daloze)**

- *File 0001-object.c-rb\_obj\_inspect-Kernel-inspect-improve-consi.patch added*

Eregon (Benoit Daloze) wrote:

I've been working on this and wondered what to do if there is no instance variable. Should it call (dynamically) `#to_s?` or always use `Kernel#to_s` which gives the class name and pointer (which would be called anyway if `#to_s` was not overridden) ?

After some thoughts I believe it would not be worth changing the behavior if it was not fully consistent.  
So I propose this patch.

test-all pass with only two modifications in `pretty_print` tests, which relied on `#inspect` calling `#to_s`.  
The internal structures have been updated so `#inspect` is an alias of `#to_s` when it was not defined.  
The last modification is to define `#inspect` on the toplevel self (only `#to_s` was defined).

Of course, tests for the new behavior need to be added (I'll do later if the idea is approved).

What do you think?

**#10 - 04/20/2012 07:00 AM - mame (Yusuke Endoh)**

Benoit, thank you!

After some thoughts I believe it would not be worth changing the behavior if it was not fully consistent.

Agreed.

So I propose this patch.

I'll review. Please wait!

test-all pass with only two modifications in pretty\_print tests, which relied on #inspect calling #to\_s.

Good point. I was missing pretty\_print.  
It must be also changed. Akr, do you have time?

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#11 - 04/21/2012 02:41 AM - mame (Yusuke Endoh)**

Hello,

So I propose this patch.

I'll review. Please wait!

Looks good to me. I'll commit it, except the change of test/test\_pp.rb. Thanks!  
(It surprised me that your patch also passes rubyspec; rubyspec has no test for this behavior?)

Tanaka-san, I made a patch for pp. Unless there is no objection, I'll commit this too.

```
diff --git a/lib/pp.rb b/lib/pp.rb
index 94269ab..6e0c797 100644
--- a/lib/pp.rb
+++ b/lib/pp.rb
@@ -265,8 +265,7 @@ class PP < PrettyPrint
 module ObjectMixin
 # 1. specific pretty_print
 # 2. specific inspect
```

- **3. specific to\_s**

- **4. generic pretty\_print**

- **3. generic pretty\_print**

**A default pretty printing method for general objects.**

**It calls #pretty\_print\_instance\_variables to list instance variables.**

```
@@ -283,18 +282,10 @@ class PP < PrettyPrint
 inspect_method = method_method.call(:inspect)
 rescue NameError
 end
```

- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- ```
if inspect_method && /\(Kernel\)#/ !~ inspect_method.inspect
  q.text self.inspect
elsif !inspect_method && self.respond_to?(:inspect)
  q.text self.inspect
```
- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]
- ```
else
```

```
q.pp_object(self)
end
```

```
diff --git a/test/test_pp.rb b/test/test_pp.rb
index fe65287..acd3e83 100644
--- a/test/test_pp.rb
+++ b/test/test_pp.rb
@@ -118,7 +118,6 @@ class PPInspectTest < Test::Unit::TestCase
def a.to_s() "aaa" end
result = PP.pp(a, ")
assert_equal("#{a.inspect}\n", result)

  • assert_equal("aaa\n", result)
    end
  end
```

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

#### #12 - 04/21/2012 02:52 AM - mame (Yusuke Endoh)

Oh, yeah, Benoit. I'm happy if you write tests for the new behavior :-)  
Do you have a commit bit already?

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

#### #13 - 04/21/2012 03:14 AM - Eregon (Benoit Daloze)

Hello,  
mame (Yusuke Endoh) wrote:

Looks good to me. I'll commit it, except the change of test/test\_pp.rb. Thanks!

Thank you for reviewing!

Oh, yeah, Benoit. I'm happy if you write tests for the new behavior :-)  
Do you have a commit bit already?

No, I don't.

I'll write these tests ASAP.

(It surprised me that your patch also passes rubyspec; rubyspec has no test for this behavior?)

I'd be happy to add some tests to rubyspec too.

#### #14 - 04/21/2012 07:54 AM - Eregon (Benoit Daloze)

I started writing the tests:  
<https://github.com/eregon/ruby/commit/f9ac4cd1daadd7c37c722c79a36296f3d161058b>

I'll improve them tomorrow and write the ones for rubyspec too.

#### #15 - 05/26/2012 10:29 PM - Eregon (Benoit Daloze)

- File 0002-test-ruby-test\_object.rb-test\_inspect-add-tests-for-.patch added

I attach the updated tests (sorry about the delay :/).  
Please comment is something is wrong.

After speaking with Yusuke, it seems best to not include a (complete) specification for #inspect in RubySpec yet, because the exact #inspect format might be an implementation detail.

However I'd like to add the behavior change (not calling #to\_s anymore) to RubySpec after this has been merged.

#### #16 - 06/22/2012 05:47 AM - drbrain (Eric Hodel)

If there are no objections I will commit this Monday (unless Benoit gets a commit bit sooner)

**#17 - 06/22/2012 07:36 PM - Eregon (Benoit Daloze)**

- File 0002-test-ruby-test\_object.rb-test\_inspect-add-tests-for-.patch added

Updated patch for the tests according to nobu's comment in <https://github.com/eregon/ruby/commit/f9ac4cd1daadd7c37c722c79a36296f3d161058b#commitcomment-1490653>.

**#18 - 07/23/2012 11:30 PM - mame (Yusuke Endoh)**

I suggested giving Benoit a commit bit at the developer meeting (7/21), and matz agreed.

Benoit, could you send your PGP public key to cvs-admin AT ruby-lang.org? See the following procedure in detail:

<http://bugs.ruby-lang.org/projects/ruby/wiki/CommitterHowto#What-to-do-for-registering-you-as-a-committer>

Then, please commit your patch yourself.

I'd like to thank you for your continuing contribution!

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#19 - 08/14/2012 07:20 PM - Eregon (Benoit Daloze)**

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

This issue was solved with changeset r36699. Thomas, thank you for reporting this issue. Your contribution to Ruby is greatly appreciated. May Ruby be with you.

Kernel#inspect: improve consistency and do not call #to\_s.

A class can now benefit from the nice default #inspect even if it defines #to\_s. Also, there is no more unexpected change in #inspect result. Internal structures have been adapted so they don't rely on the removed behavior (#inspect calling overridden #to\_s).

- object.c (rb\_obj\_inspect): Kernel#inspect: do not call #to\_s.
- test/ruby/test\_object.rb (test\_inspect): add tests for Kernel#inspect.
- bignum.c, io.c, numeric.c, object.c, proc.c, vm.c (Init\_\*): alias #inspect to #to\_s where it was expected. [\[ruby-core:43238\]](#)[Feature [#6130](#)]

**#20 - 08/14/2012 08:15 PM - Eregon (Benoit Daloze)**

Hi,  
mame (Yusuke Endoh) wrote:

I suggested giving Benoit a commit bit at the developer meeting (7/21), and matz agreed.

[...]  
Then, please commit your patch yourself.

I'd like to thank you for your continuing contribution!

I just committed as r36699 and I committed your PP patch as r36700.

Thank you for proposing it!

**Files**

0001-object.c-rb_obj_inspect-Kernel-inspect-improve-consi.patch	7.48 KB	04/11/2012	Eregon (Benoit Daloze)
0002-test-ruby-test_object.rb-test_inspect-add-tests-for-.patch	1.44 KB	05/26/2012	Eregon (Benoit Daloze)
0002-test-ruby-test_object.rb-test_inspect-add-tests-for-.patch	1.44 KB	06/22/2012	Eregon (Benoit Daloze)