

Ruby 1.8 - Bug #2161

Kernel.eval ignores binding in 1.9

09/30/2009 07:04 AM - awieland (Aaron Wieland)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	1.9.1p243 (2009-07-16 revision 24175) [i386-darwin9.8.0]
Description	
<pre>=begin Kernel.eval no longer evaluates its string in the context of the binding passed to it. In an irb session in 1.8.6: irb(main):001:0> eval 'self' => main irb(main):002:0> obj = Object.new => #Object:0x12aaf20 irb(main):003:0> eval 'self', obj.send(:binding) => #Object:0x12aaf20 In 1.9.1 (also tested in the latest nightly build for 1.9.2): irb(main):001:0> obj = Object.new => #Object:0x5210f8 irb(main):002:0> eval 'self', obj.send(:binding) => main I can't help but feel I'm missing something, because a bug of this magnitude should have been identified and fixed by now (it breaks a lot of code that uses ERb). =end</pre>	

History

#1 - 10/08/2009 07:35 PM - wanabe (_ wanabe)

```
=begin
It seems that Kernel#eval works fine.
```

```
$. /ruby -ve 'class Foo;def foo;binding;end;end;eval "p self", Foo.new.foo'
ruby 1.9.2dev (2009-10-02 trunk 25185) [i386-mingw32]
#Foo:0xb23f80
```

```
I guess obj.send(:binding)' is same asbinding' in 1.9.
And it is undefined behavior, because Kernel#binding is a private method.
How about Object#instance_eval instead of it?
=end
```

#2 - 10/21/2009 11:11 PM - naruse (Yui NARUSE)

- Status changed from Open to Feedback

```
=begin
=end
```

#3 - 04/16/2010 10:40 PM - mame (Yusuke Endoh)

```
=begin
Hi,
```

Rdoc of Kernel#binding says:

Returns a +Binding+ object, describing the variable and method bindings at the point of call.

In this case, "the point of call" is top level.
So I think 1.9 is correct.

I don't know it is 1.8's bug or spec change since 1.9.
So I move this ticket to 1.8 subproject.

As wanabe said, you can use `instance_eval` if you want 1.8's current behavior:

```
eval "self", obj.instance_eval { binding }
```

--

Yusuke Endoh mame@tsg.ne.jp
=end