

## Ruby master - Bug #9593

### Keyword arguments default argument assignment behaviour not consistent with optional argument

03/05/2014 06:00 AM - chendo (Jack Chen)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	matz (Yukihiro Matsumoto)	
<b>Target version:</b>	2.2.0	
<b>ruby -v:</b>	2.1.1	<b>Backport:</b> 1.9.3: DONTNEED, 2.0.0: DONTNEED, 2.1: DONTNEED

#### Description

Given the following code:

```
def var
  100
end

def foo(var: var + 1)
  puts "var: #{var.inspect}"
end

def bar(var = var + 1)
  puts "var: #{var.inspect}"
end

foo(var: 1)
foo rescue p $!

bar(1)
bar
```

#### Ruby 2.0.0:

```
var: 1
var: 101
var: 1
var: 101
```

#### Ruby 2.1.1:

```
var: 1
#<NoMethodError: undefined method `+' for nil:NilClass>
var: 1
var: 101
```

What appears to be happening is that since 2.1.1, the keyword argument defines var as a variable before evaluating the default argument. Personally, I prefer 2.0.0 behaviour, but the way 2.1.1 handles default arguments in non keyword arguments is inconsistent.

#### Related issues:

Related to Ruby master - Bug #10314: Default argument lookup fails in Ruby 2....	<b>Closed</b>	
Related to Ruby master - Bug #11074: Block with optional parameter of same cl...	<b>Rejected</b>	<b>04/18/2015</b>
Has duplicate Ruby master - Bug #10280: Regression while evaluating default a...	<b>Rejected</b>	<b>09/22/2014</b>

#### Associated revisions

##### Revision 01740f0c - 03/05/2014 06:56 AM - nobu (Nobuyoshi Nakada)

parse.y: optional arguments in rhs

- parse.y (f\_arg\_asgn): define optional arguments as argument variables in the rhs default expressions. [ruby-core:61299] [Bug #9593]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45272 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 45272 - 03/05/2014 06:56 AM - nobu (Nobuyoshi Nakada)**

parse.y: optional arguments in rhs

- parse.y (f\_arg\_asgn): define optional arguments as argument variables in the rhs default expressions. [ruby-core:61299] [Bug #9593]

**Revision 45272 - 03/05/2014 06:56 AM - nobu (Nobuyoshi Nakada)**

parse.y: optional arguments in rhs

- parse.y (f\_arg\_asgn): define optional arguments as argument variables in the rhs default expressions. [ruby-core:61299] [Bug #9593]

**Revision 45272 - 03/05/2014 06:56 AM - nobu (Nobuyoshi Nakada)**

parse.y: optional arguments in rhs

- parse.y (f\_arg\_asgn): define optional arguments as argument variables in the rhs default expressions. [ruby-core:61299] [Bug #9593]

**Revision bd4aa40f - 10/03/2014 03:09 PM - nobu (Nobuyoshi Nakada)**

NEWS: mentioned about [Bug #9593]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@47776 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 47776 - 10/03/2014 03:09 PM - nobu (Nobuyoshi Nakada)**

NEWS: mentioned about [Bug #9593]

**Revision 47776 - 10/03/2014 03:09 PM - nobu (Nobuyoshi Nakada)**

NEWS: mentioned about [Bug #9593]

**Revision 47776 - 10/03/2014 03:09 PM - nobu (Nobuyoshi Nakada)**

NEWS: mentioned about [Bug #9593]

**Revision 2de3281a - 12/14/2014 11:11 AM - nobu (Nobuyoshi Nakada)**

parse.y: warn reference after method definition

- parse.y (primary): restore current\_arg so that circular reference after a method definition is also warned. [ruby-core:61299] [Bug #9593]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@48835 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 48835 - 12/14/2014 11:11 AM - nobu (Nobuyoshi Nakada)**

parse.y: warn reference after method definition

- parse.y (primary): restore current\_arg so that circular reference after a method definition is also warned. [ruby-core:61299] [Bug #9593]

**Revision 48835 - 12/14/2014 11:11 AM - nobu (Nobuyoshi Nakada)**

parse.y: warn reference after method definition

- parse.y (primary): restore current\_arg so that circular reference after a method definition is also warned. [ruby-core:61299] [Bug #9593]

**Revision 48835 - 12/14/2014 11:11 AM - nobu (Nobuyoshi Nakada)**

parse.y: warn reference after method definition

- parse.y (primary): restore current\_arg so that circular reference after a method definition is also warned. [ruby-core:61299] [Bug #9593]

**History**

---

**#1 - 03/05/2014 06:18 AM - nobu (Nobuyoshi Nakada)**

The behaviors in 2.0 are incorrect both.

**#2 - 03/05/2014 06:56 AM - nobu (Nobuyoshi Nakada)**

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset r45272.

---

parse.y: optional arguments in rhs

- parse.y (f\_arg\_asgn): define optional arguments as argument variables in the rhs default expressions. [ruby-core:61299] [Bug #9593]

### #3 - 03/05/2014 06:58 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN to 1.9.3: REQUIRED, 2.0.0: REQUIRED, 2.1: REQUIRED

### #4 - 03/12/2014 06:17 PM - tenderlovmaking (Aaron Patterson)

Hi,

Before r45272 this code worked:

```
require 'minitest/autorun'

class DefaultArg < MiniTest::Unit::TestCase
  class Default
    def foo; 'foo'; end

    def bar foo = foo
      foo
    end
  end

  def test_default
    assert_equal 'foo', Default.new.bar
  end
end
```

Was this expected to break? Unfortunately we have real code that depends on the behavior:

[https://github.com/rails/rails/blob/81d08abcccf2ef1b0ea6e98daf00f6ca375f3d8a/activerecord/lib/active\\_record/associations/has\\_many\\_association.rb#L74](https://github.com/rails/rails/blob/81d08abcccf2ef1b0ea6e98daf00f6ca375f3d8a/activerecord/lib/active_record/associations/has_many_association.rb#L74)

but we can change it

### #5 - 03/13/2014 07:26 AM - nobu (Nobuyoshi Nakada)

Aaron Patterson wrote:

Was this expected to break?

Yes. It was a bug.

An assignment creates a variable and it hides same name method in its RHS, as you can't call foo method inside bar method:

```
def foo; 'foo'; end

def bar
  foo = foo # nil
end
```

Unfortunately we have real code that depends on the behavior:

[https://github.com/rails/rails/blob/81d08abcccf2ef1b0ea6e98daf00f6ca375f3d8a/activerecord/lib/active\\_record/associations/has\\_many\\_association.rb#L74](https://github.com/rails/rails/blob/81d08abcccf2ef1b0ea6e98daf00f6ca375f3d8a/activerecord/lib/active_record/associations/has_many_association.rb#L74)

Sorry for the bug.

but we can change it

Always you can call the method explicitly with ().

### #6 - 03/13/2014 05:22 PM - tenderlovmaking (Aaron Patterson)

Ok! I will fix the code in Rails. Thank you!

### #7 - 03/13/2014 05:52 PM - tenderlovmaking (Aaron Patterson)

Nobu, are you *sure* this is a bug? This test has worked since 1.9.3, and this code was committed in 2008:

[https://github.com/rails/rails/blob/master/activesupport/lib/active\\_support/values/time\\_zone.rb#L285](https://github.com/rails/rails/blob/master/activesupport/lib/active_support/values/time_zone.rb#L285)

It's been the behavior for at least 6 years. I don't mind changing Rails to work with this, but people who don't test with trunk may get a nasty surprise when Ruby 2.2 is released.

Edit: here's a better link:

[https://github.com/rails/rails/blob/15720df1808b249bae91e5600d6f0676990a7de0/activesupport/lib/active\\_support/values/time\\_zone.rb#L285](https://github.com/rails/rails/blob/15720df1808b249bae91e5600d6f0676990a7de0/activesupport/lib/active_support/values/time_zone.rb#L285)

**#8 - 03/13/2014 06:05 PM - nobu (Nobuyoshi Nakada)**

Yes, it's a very old bug.  
Optional arguments have same semantics as assignments.

**#9 - 06/16/2014 03:19 PM - nagachika (Tomoyuki Chikanaga)**

Hi,

I think this change should not be backported to stable branch.  
I agree about that it is a long standing bug. But it lives too long to change a behavior during stable releases.  
I'm going to fill 'DONTNEED' in 'Backport' field if there's no objection.

**#10 - 06/28/2014 05:55 AM - nagachika (Tomoyuki Chikanaga)**

- Backport changed from 1.9.3: *REQUIRED*, 2.0.0: *REQUIRED*, 2.1: *REQUIRED* to 1.9.3: *DONTNEED*, 2.0.0: *DONTNEED*, 2.1: *DONTNEED*

No objection. I changed Backport field to DONTNEED.

**#11 - 09/23/2014 12:22 AM - nobu (Nobuyoshi Nakada)**

- Has duplicate Bug #10280: Regression while evaluating default arguments of a method added

**#12 - 10/02/2014 06:15 AM - hsbt (Hiroshi SHIBATA)**

- Related to Bug #10314: Default argument lookup fails in Ruby 2.2 for circular shadowed variable names added

**#13 - 04/18/2015 06:06 AM - nobu (Nobuyoshi Nakada)**

- Related to Bug #11074: Block with optional parameter of same closure variable name added