

## Ruby master - Bug #10758

### undefined method `year' for nil:NilClass

01/20/2015 09:37 AM - rohandaxini (Rohan Daxini)

<b>Status:</b> Third Party's Issue	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b> 2.2.0	<b>Backport:</b> 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN
<b>Description</b>	
<p>I am facing this error only in Ruby 2.2.0 as I recently upgraded to latest Ruby 2.2.0</p> <p>Here is the console log</p> <pre>2.2.0 :008 &gt; Time.zone.parse("2014-12-29 20:16:32 -0400") NoMethodError: undefined method `year' for nil:NilClass     from /.rvm/gems/ruby-2.2.0/gems/activesupport-3.2.18/lib/active_support/values/time_zone.rb:27:in `parse'     from (irb):8     from /.rvm/gems/ruby-2.2.0/gems/railties-3.2.18/lib/rails/commands/console.rb:47:in `start'     from /.rvm/gems/ruby-2.2.0/gems/railties-3.2.18/lib/rails/commands/console.rb:8:in `start'     from /.rvm/gems/ruby-2.2.0/gems/railties-3.2.18/lib/rails/commands.rb:41:in `<top (required)="">'</top></pre> <p>Whereas same works fine in Ruby 2.1.2</p> <pre>2.1.2 :001 &gt; Time.zone.parse("2014-12-29 20:16:32 -0400") =&gt; Tue, 30 Dec 2014 00:16:32 UTC +00:00</pre> <p>Am I doing something wrong here? Do I need to use any other gem to make this work. Some 700 specs fail for my project after upgrading to Ruby 2.2.0 due to this undefined method 'year' errors.</p> <p>Would highly appreciate your immediate help in this regard. Thanks in advance.</p>	

#### History

##### #1 - 01/20/2015 11:28 AM - Hanmac (Hans Mackowiak)

try to use newer activesupport version, it seems to be fixed in newer versions

##### #2 - 01/20/2015 01:21 PM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Third Party's Issue

You should report rails/rails repository in github at first

##### #3 - 01/20/2015 01:26 PM - rohandaxini (Rohan Daxini)

Hiroshi SHIBATA wrote:

You should report rails/rails repository in github at first

Sorry but I did not know this is Rails issue. As the same code works fine on Ruby 2.1.2 but throws error on Ruby 2.2.0 despite keeping same Active record.

##### #4 - 01/24/2015 11:48 PM - Anonymous

Could the behavior of the following code have changed between Ruby 2.1.2 and Ruby 2.2.0?

In the Rails 3.2.21 with Ruby 2.1.2 version of activesupport the following works:

```
def parse(str, now=now)
```

With Ruby 2.2.0 it no longer works as the **now** on the right side is evaluated as **nil**. In the latest version of activesupport on master on github that line

has changed to the following:

```
def parse(str, now=now())
```

With 2.2.0 is it now necessary to use the () when referencing a function with the same name a default value assignment?

**#5 - 01/25/2015 12:14 AM - nobu (Nobuyoshi Nakada)**

- *Description updated*

From NEWS in 2.2.0

---

## Changes since the 2.1.0 release

### Language changes

- default argument

fixed a very longstanding bug that an optional argument was not accessible in its default value expression. [Bug [#9593](#)]