

Ruby trunk - Bug #11027

Named Argument assignment from Hash failure

04/02/2015 10:47 PM - alangano (Alan Gano)

Status: Closed	
Priority: Normal	
Assignee: ko1 (Koichi Sasada)	
Target version:	
ruby -v: ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-linux]	Backport: 2.0.0: DONTNEED, 2.1: DONTNEED, 2.2: DONE
Description	
<p>I am call a method, which has named arguments, a Hash, which then as expected, maps the keys of the Hash to the argument names.</p> <p>I boiled down a problem scenario wherein it fails to do this. In the code below the argument <code>i_field2</code> of the <code>ack()</code> method, is failing to be set from the Hash. It gets set to <code>nil</code>.</p> <p>This also occurs in 2.2.0p0.</p> <pre>require 'base64' def ack(i_field1:, i_field2:, **i_others) puts '-'*79 puts i_field1.inspect puts ">>> #{i_field2.inspect} <<< SHOULD be 'field2'" puts i_others.inspect end # create the marshaled hash in another execution #t_args = {i_field1:"field1", i_field2:"field2", something:"something"} #puts Base64.strict_encode64(Marshal.dump(t_args)) #exit t_marshal = 'BAh7CD0NaV9maWVsZDFJIGtmaWVsZDEGOGZfVDoNaV9maWVsZDJJIGtmaWVsZDIG0wZUOg5zb21ldGhpbmdJIg5zb21ldGhpbmcGOWZU' t_args = Marshal.load(Base64.strict_decode64(t_marshal)) puts t_args.inspect # => {:i_field1=>"field1", :i_field2=>"field2", :something=>"something"} ack(t_args) #_ = { something:'?' } # makes it work #_ = { :something => '?' } # makes it work #_ = :something # makes it work</pre>	
Related issues:	
Related to Ruby trunk - Bug #11382: Keyword arguments randomly nullified	Closed
Has duplicate Ruby trunk - Bug #11091: Symbolized Strings May Break Keyword A...	Closed

Associated revisions

Revision 633574ee - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50172 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 50172 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

Revision 50172 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

Revision 50172 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

Revision 50172 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

Revision 50172 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

Revision 3e199f4c - 05/20/2015 04:04 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 50172,50173: [Backport #11027]

```
* vm_args.c: protect value stack from calling other methods
during complex parameter setting process (splat, kw, and so on).
[Bug #11027]
```

```
* vm_core.h: remove rb_thread_t::mark_stack_len.
With this modification, we don't need to use th->mark_stack_len.
```

```
* test/ruby/test_keyword.rb: add a test.
```

```
* cont.c (cont_capture): catch up this fix.
```

```
* vm.c (rb_thread_mark): ditto.
```

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_2@50562](https://ci.ruby-lang.org/ruby/branches/ruby_2_2@50562) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 50562 - 05/20/2015 04:04 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 50172,50173: [Backport #11027]

```
* vm_args.c: protect value stack from calling other methods
during complex parameter setting process (splat, kw, and so on).
[Bug #11027]
```

```
* vm_core.h: remove rb_thread_t::mark_stack_len.
With this modification, we don't need to use th->mark_stack_len.
```

```
* test/ruby/test_keyword.rb: add a test.
```

* cont.c (cont_capture): catch up this fix.

* vm.c (rb_thread_mark): ditto.

History

#1 - 04/02/2015 10:52 PM - alangano (Alan Gano)

Restatement:

I have a method with named arguments. I am calling it with a Hash, whose keys match up with the names of the method arguments.

This has been working.

I boiled down a problem scenario wherein it fails to do this. In the code below the argument `i_field2` of the `ack()` method, is failing to be set from the Hash. It gets set to `nil`.

This also occurs in 2.2.0p0.

Formatting fix ...

```
require 'base64'

def ack(i_field1:, i_field2:, **i_others)
  puts '-'*79
  puts i_field1.inspect
  puts ">>> #{i_field2.inspect} <<< SHOULD be 'field2'"
  puts i_others.inspect
end

# create the marshaled hash in another execution
#t_args = {i_field1:"field1", i_field2:"field2", something:"something"}
#puts Base64.strict_encode64(Marshal.dump(t_args))
#exit

t_marshal =
'BAh7CD0NaV9maWVsZDFJIGtmaWVsZDEGOgZFVDoNaV9maWVsZDJIIGtmaWVsZDIGOwZUOg5zb21ldGhpbmdJIg5zb21ldGhpbmcGOwZU'

t_args = Marshal.load(Base64.strict_decode64(t_marshal))

puts t_args.inspect
# => {:i_field1=>"field1", :i_field2=>"field2", :something=>"something"}

ack(t_args)

#_ = { something:'?' }      # makes it work
#_ = { :something => '?' }  # makes it work
#_ = :something           # makes it work
```

#2 - 04/03/2015 03:20 AM - nobu (Nobuyoshi Nakada)

- Description updated

Marshal is not mandatory.

It can reproduce with a dynamic symbol keyword argument.

```
ack(i_field1:"field1", i_field2:"field2", "something".intern=>"something")
```

#3 - 04/03/2015 04:46 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

Ko1 is going to fix this issue by advancing `sp` prior to arguments setup instead of `mark_stack_len`.

#4 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN to 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: REQUIRED

#5 - 04/06/2015 07:14 AM - ko1 (Koichi Sasada)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

Applied in changeset [r50172](#).

- `vm_args.c`: protect value stack from calling other methods during complex parameter setting process (splat, kw, and so on). [Bug #11027]
- `vm_core.h`: remove `rb_thread_t::mark_stack_len`. With this modification, we don't need to use `th->mark_stack_len`.
- `test/ruby/test_keyword.rb`: add a test.
- `cont.c` (`cont_capture`): catch up this fix.
- `vm.c` (`rb_thread_mark`): ditto.

#6 - 04/24/2015 06:13 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Bug #11091: Symbolized Strings May Break Keyword Arguments added

#7 - 04/24/2015 06:18 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: REQUIRED to 2.0.0: DONTNEED, 2.1: DONTNEED, 2.2: REQUIRED

#8 - 05/20/2015 04:05 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.0.0: DONTNEED, 2.1: DONTNEED, 2.2: REQUIRED to 2.0.0: DONTNEED, 2.1: DONTNEED, 2.2: DONE

Backported into `ruby_2_2` branch at r50562.

#9 - 08/10/2015 03:17 AM - nagachika (Tomoyuki Chikanaga)

- Related to Bug #11382: Keyword arguments randomly nullified added