

## Ruby - Bug #4480

### Thread-local variables issue: Thread#[] returns nil when called first time

03/08/2011 02:26 AM - dre3k (Andrei Kulakov)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	nahi (Hiroshi Nakamura)	
<b>Target version:</b>	1.9.3	
<b>ruby -v:</b>	-	
		<b>Backport:</b>
<b>Description</b> =begin In ruby 1.9.2p180 t = Thread.new do Thread.current[:var] = "var" Thread.stop end  p t[:var] #=> nil p t[:var] #=> "var"  In ruby 1.8.7 p t[:var] #=> "var" p t[:var] #=> "var" =end		

#### Associated revisions

Revision 4a42dda4 - 06/27/2011 08:09 PM - Hiroshi Nakamura

- thread.c (rb\_thread\_local\_aref): RDoc fix. Thread#[] example had a race. See #4480.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32260 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### History

#1 - 03/12/2011 05:42 PM - robertgleeson (Robert Gleeson)

=begin  
Hi Andrei,

I don't think this is a bug. You should call Thread#join before you call Kernel.p.  
Your thread may or may not have been run by the time you call Kernel.p, and you may get inconsistent results such as t[:var] being nil sometimes, and being "var" at other times.

Thanks,  
Rob  
=end

#2 - 03/12/2011 06:48 PM - dre3k (Andrei Kulakov)

=begin  
Hi Robert,

Thanks a lot. You are totally right about Thread#join.

But what the solution in my particular case, how do I get thread-local variable when Thread.stop called within a thread.

```
t = Thread.new do
  Thread.current[:var] = "var"
  Thread.stop
end
```

```
t.join
```

```
p t[:var]
p t[:var]
```

```
yields
threads.rb:6:in join': deadlock detected (fatal) from threads.rb:6:in '
and it's supposed to be like that.
```

Apologies if it's a stupid question.  
=end

### #3 - 03/12/2011 08:26 PM - dre3k (Andrei Kulakov)

- File *thread\_rdoc\_fix.patch* added

```
=begin
I also wanted to say this is rdoc's example of Thread#[] the one that tricked me.
```

Right now this is:

```
a = Thread.new { Thread.current["name"] = "A"; Thread.stop }
b = Thread.new { Thread.current[:name] = "B"; Thread.stop }
c = Thread.new { Thread.current["name"] = "C"; Thread.stop }
Thread.list.each {|x| puts "#{x.inspect}: #{x[:name]}" }
produces:
#<Thread:0x401b3b3c sleep>: C
#<Thread:0x401b3bc8 sleep>: B
#<Thread:0x401b3c68 sleep>: A
#<Thread:0x401bdf4c run>:
```

But on some slower machine it will produce different results.

Maybe it's better to change this example to something like following:

```
a = Thread.new { Thread.current["name"] = "A" }
b = Thread.new { Thread.current[:name] = "B" }
c = Thread.new { Thread.current["name"] = "C" }
```

```
Thread.list.each do |thr|
thr.join unless thr == Thread.main
puts "#{thr.inspect}: #{thr[:name]}"
end
produces:
#<Thread:0x88bf918 run>:
#<Thread:0x88b22cc dead>: A
#<Thread:0x88b2290 dead>: B
#<Thread:0x88b2254 dead>: C
```

Or some other more consistent example than current.  
=end

### #4 - 06/26/2011 06:24 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to *nahi* (Hiroshi Nakamura)

### #5 - 06/26/2011 08:23 PM - ko1 (Koichi Sasada)

- ruby -v changed from ruby 1.9.2p180 (2011-02-18 revision 30909) [i686-linux] to -

(2011/06/26 18:24), Yui NARUSE wrote:

```
In ruby 1.9.2p180
t = Thread.new do
  Thread.current[:var] = "var"
  Thread.stop
end
```

```
p t[:var] #=> nil
p t[:var] #=> "var"
```

```
In ruby 1.8.7
p t[:var] #=> "var"
p t[:var] #=> "var"
```

It seems intentional. When the first access, the Thread.current[:var] is not initialized (not reached the initialization code).

--  
// SASADA Koichi at atdot dot net

**#6 - 06/27/2011 04:23 PM - nahi (Hiroshi Nakamura)**

On Sun, Jun 26, 2011 at 20:12, SASADA Koichi [ko1@atdot.net](mailto:ko1@atdot.net) wrote:

It seems intentional. When the first access, the Thread.current[:var] is not initialized (not reached the initialization code).

You're right. I took this ticket since it would be a RDoc issue.

**#7 - 06/28/2011 05:14 AM - nahi (Hiroshi Nakamura)**

- Category set to doc

- Target version changed from 1.9.2 to 1.9.3

Updated RDoc at r32260 based on Andrei's patch. I just wanted to remove Thread.main thing to concentrate Thread#[] behavior. Thanks!

**#8 - 06/28/2011 05:14 AM - nahi (Hiroshi Nakamura)**

- Status changed from Assigned to Closed

**Files**

---

thread_rdoc_fix.patch	1.14 KB	03/12/2011	dre3k (Andrei Kulakov)
-----------------------	---------	------------	------------------------