Ruby trunk - Bug #6291

Backtrace printout halts for an extremely long time when large amounts of data are allocated

04/14/2012 04:50 AM - cuthbert (Christoffer Lervag)

Status:	Closed		
Priority:	Normal		
Assignee:	matz (Yukihiro Matsumoto)		
Target version:	1.9.3		
ruby -v:	1.9.3	**Backport:**	

Description

=begin
When large amounts of data are allocated, it can take an eternity for the backtrace printout to complete. Here's a simple example:

```
class SomeClass
  def initialize
    @a = Array.new(100000, "just some random string")
  end
end
h = Hash.new
200.times {|i| h[i] = SomeClass.new}
h.asdf
```

This will halt with the following printout:
test.rb:9

And then, after a very long time (depending on how much data is in memory), it will eventually finish:
test.rb:9: undefined method `asdf' for #Hash:0x1001bc348 (NoMethodError)

For me, this has caused a lot of frustration when running a test suite on an application which loads a lot of data in memory, as in many cases I will never get to see the actual backtrace. This makes debugging much more difficult. Perhaps there is some fundamental property of the Ruby language itself that makes this an 'expected behaviour', but if not, I would really appreciate to see this adressed.
=end

Related issues:

Has duplicate Ruby trunk - Bug #6783: Infinite loop in inspect, not overridin...	**Open**

History

#1 - 04/20/2012 04:44 AM - mame (Yusuke Endoh)

Hello,

I have no idea except adding a length limitation or timeout to #inspect.
Any other ideas?

--
Yusuke Endoh mame@tsg.ne.jp

#2 - 04/22/2012 11:31 AM - mame (Yusuke Endoh)

- *Status changed from Open to Assigned*
- *Assignee set to matz (Yukihiro Matsumoto)*

Matz,

Do you think that #inspect has a length limitation or timeout?

--
Yusuke Endoh mame@tsg.ne.jp

#3 - 07/14/2012 05:45 PM - matz (Yukihiro Matsumoto)

- *Status changed from Assigned to Closed*

This one is not easy to fix.
We try to address it by improving #inspect, but we cannot make promise.

Matz.

#4 - 10/11/2016 07:43 AM - shyouhei (Shyouhei Urabe)

- Has duplicate Bug #6783: Infinite loop in inspect, not overriding inspect, to_s, and no known circular references. Stepping into inspect in debugger locks it up with 100% CPU. added