

Ruby master - Feature #4521

NoMethodError#message may take very long to execute

03/25/2011 04:59 AM - adiel.mittmann (Adiel Mittmann)

Status:	Assigned
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	
Description	
<p>=begin</p> <p>When a non-existing method is called on an object, NoMethodError is risen. If you call #message, however, your code may use up all CPU for a very long time (in my case, up to a few minutes).</p> <p>I narrowed the problem down to this code in error.c (SVN snapshot) in the function name_err_mesg_to_str():</p> <pre>d = rb_protect(rb_inspect, obj, 0); if (NIL_P(d) RSTRING_LEN(d) > 65) { d = rb_any_to_s(obj); }</pre> <p>The problem is that, for a complex object, #inspect may take very long to execute, only to have its results thrown away because they will be larger than 65 characters.</p> <p>Of course I can write a #to_s for all my objects, but the point is that I didn't call #to_s or #inspect, I called #message on an exception object, which then takes a few minutes just to return a short string.</p> <p>Needless to say, this might be easy to spot in a simple example, but once you're writing a web application that suddenly freezes for one minute with no apparent reason, you're all but clueless as to what's going on. (The first time this happened, I didn't even know that something would eventually show up on the screen -- I thought it was an infinite loop).</p> <p>Here's an example code that shows this behavior:</p> <pre>require 'nokogiri' class A def x @xml = Nokogiri::XML(File.new('baz.xml', 'rb').read()) foo() end end A.new().x() a.x</pre> <p>Here, the time it takes for Ruby to print out the message that #foo doesn't exist is proportional to the size of baz.xml.</p> <p>As a comparison, Python doesn't seem to do this. Take the following code:</p> <pre>class Test: def str(self): return "hello" a = Test() print a print a.x()</pre> <p>If you execute it, this is the result:</p> <pre>hello Traceback (most recent call last): File "test.py", line 6, in print a.x() AttributeError: Test instance has no attribute 'x'</pre> <p>It uses the method str to convert the object to a string when necessary, but doesn't use it when printing out the message stating that the attribute doesn't exist.</p>	

One obvious way to fix this would be to always print out the simpler representation given by `rb_any_to_s`.
=end

History

#1 - 06/11/2011 05:12 PM - ko1 (Koichi Sasada)

- *Tracker changed from Bug to Feature*
- *Assignee set to matz (Yukihiko Matsumoto)*

It is not a bug, but a feature request.

#2 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- *Status changed from Open to Assigned*

#3 - 11/20/2012 09:22 PM - mame (Yusuke Endoh)

- *Description updated*
- *Target version set to 2.6*

#4 - 12/25/2017 06:14 PM - naruse (Yui NARUSE)

- *Target version deleted (2.6)*