

Ruby master - Bug #5067

BasicObject's constant lookup documentation

07/21/2011 11:11 PM - shyouhei (Shyouhei Urabe)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: doesn't matter	Backport:
Description	
Take a look at https://github.com/ruby/ruby/pull/31 .	
It's OK for me to merge this request -- no technical difficulties -- but is it a right thing to be documented?	
Related issues:	
Related to Ruby master - Feature #3768: Constant Lookup doesn't work in a sub...	Rejected 08/31/2010

Associated revisions

Revision 3eb7d2b3 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32700 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 32700 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

Revision 32700 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

Revision 32700 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

Revision 32700 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

Revision 32700 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

Revision 32700 - 07/27/2011 06:44 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

Revision b727b2a4 - 07/27/2011 06:49 AM - drbrain (Eric Hodel)

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug #5067]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@32701 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 07/21/2011 11:26 PM - mame (Yusuke Endoh)

Matz explicitly stated in [#3768](#), that it is intended that BasicObject does not resolve constants beyond itself. But he does not make mention to the workaround

by using `const_missing`.

--

Yusuke Endoh mame@tsg.ne.jp

#2 - 07/22/2011 01:29 AM - matz (Yukihiko Matsumoto)

I agree to merge this documentation clarification, but without the last five lines. The work around may or may not work, depending on the applications's requirement. I don't want to encourage the half-baked solution in the document.

matz.

#3 - 07/22/2011 02:23 AM - trans (Thomas Sawyer)

But that was the main point of adding the documentation! And how can it be "half-baked"? What other solution is there?

The documentation states that it is "if" you need to reference constant normally from within a basic object, then the solution is `#const_missing`. Maybe it needs to be worded better, but that's the ONLY solution there is.

Maybe you do not write DSL often, I work with them very often. Having known this upfront could have saved me many hours and days of headache. And saved me from reporting it as a bug. It sure seems like bug at first. But you and others have explained to me why it is not a bug. It only seems right to explain that in the documentation and to give the ONLY work around if it is needed.

#4 - 07/22/2011 02:40 AM - matz (Yukihiko Matsumoto)

Well, `BasicObject` is a blank slate, and adding `const_missing` to it could affect all classes in the Ruby class hierarchy. Even though it might solve your problem, that could have huge side effect, e.g. entering infinite loop.

OKay, as a compromise, I can accept more clarified version of workaround.

```
class Foo < BasicObject
  def self.const_missing(name)
    ::Object.const_get(name)
  end
end
```

How do you think?

matz.

#5 - 07/22/2011 02:47 AM - trans (Thomas Sawyer)

Ah, I see. Yes, your example is what is meant.

#6 - 07/22/2011 03:20 AM - drbrain (Eric Hodel)

In light of the workaround required to access Ruby classes descending from `Object` I think "very useful as the base class for DSLs" is not quite accurate.

I think the first paragraph should say something more like "useful for creating object systems that won't be interfered with Ruby's object system" and the second paragraph should explain various workarounds required.

It might be useful to mention you can include `Kernel` in `BasicObject` descendants to provide access to methods like `open`, `require`, etc.

#7 - 07/22/2011 10:33 AM - trans (Thomas Sawyer)

[eric \(Eric Anderson\)](#) Makes sense. I think "system" might be too overloaded a word, maybe "hierarchy"?

Should I work on wording and update example?

#8 - 07/22/2011 09:35 PM - trans (Thomas Sawyer)

Here is my improved documentation:

<https://github.com/trans/ruby/commit/70a66e08780b5bc77fb47a91cbefc372147a716e>

If overall satisfactory, please merge and tweak as needed.

#9 - 07/27/2011 03:44 PM - drbrain (Eric Hodel)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r32700.

Shyouhei, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- object.c: Add usage documentation for BasicObject. Based on patch by Thomas Sawyer. [Ruby 1.9 - Bug [#5067](#)]