

## Ruby master - Feature #7414

### Now that `const_get` supports "`Foo::Bar`" syntax, so should `const_defined?`.

11/21/2012 01:06 AM - robertgleeson (Robert Gleeson)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	tenderlovmaking (Aaron Patterson)
<b>Target version:</b>	2.1.0
<b>Description</b> class Foo end Object.const_get "Object::Foo" => Foo Object.const_defined? "Object::Foo" => NameError (expected true).	

#### Associated revisions

##### Revision 241ca7bf - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44194 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 44194 - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

##### Revision 44194 - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

##### Revision 44194 - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

##### Revision 44194 - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

##### Revision 44194 - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

##### Revision 44194 - 12/14/2013 02:25 AM - nobu (Nobuyoshi Nakada)

object.c: nested path `const_defined?`

- object.c (rb\_mod\_const\_defined): support nested class path as well as `const_get`. [Feature #7414]

#### History

##### #1 - 11/21/2012 03:17 AM - robertgleeson (Robert Gleeson)

I realise `defined?` with a constant works just as well but this is about API consistency. I'd expect `const_get` & `const_defined?` to support the same syntax ("`Object::Foo`").

##### #2 - 11/24/2012 11:49 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to tenderlovmaking (Aaron Patterson)

Aaron, can you fix it easily?

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#3 - 12/04/2012 04:35 PM - sferik (Erik Michaels-Ober)**

I also noticed this inconsistency. It seems very odd to have different behavior for `Module#const_get` and `Module#const_defined?`. I hope it's possible to get this change into 2.0.0. It's been a while since I've written any C but I'd be happy to contribute failing specs if that would be helpful.

**#4 - 12/18/2012 04:35 AM - benolee (Ben Holley)**

I have been working on a patch, but I have some questions about expected behavior.

In the case when the parent is missing, for example:

```
# Foo is missing
Object.const_defined?("Foo::Bar")
```

should it return false? Or raise `NameError` for "Foo"? Or call `Object.const_missing(:Foo)`?

Second, in the case when the parent is not a `Module` or `Class`, for example:

```
Foo = Object.new
Object.const_defined?("Foo::Bar")
```

should it return false? Or raise error "Foo is not a class/module"?

**#5 - 01/09/2013 02:31 AM - nobu (Nobuyoshi Nakada)**

I vote for false in both cases, like `defined?` operator.

**#6 - 02/24/2013 09:21 PM - ko1 (Koichi Sasada)**

aaron, could you continue this discussion?

**#7 - 02/24/2013 09:22 PM - ko1 (Koichi Sasada)**

- Target version changed from 2.0.0 to 2.1.0

**#8 - 12/14/2013 11:25 AM - nobu (Nobuyoshi Nakada)**

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r44194.

Robert, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

object.c: nested path `const_defined?`

- object.c (`rb_mod_const_defined`): support nested class path as well as `const_get`. [Feature [#7414](#)]