Ruby master - Bug #6644

Weird behavior of defined?(super) check invoked from a metaprogrammatically defined class method

06/25/2012 09:04 PM - alexisowl (Alexey Smolianinov)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 1.9.3p0 (2011-10-30 revision 33570) [i686-linux]</td>
</tr>
</tbody>
</table>

Description

Lately I came across some weird behavior with the (({defined?})) operator used to check if (({super})) keyword can be invoked in current context. Usually it works fine, but when I tried to combine the (({defined? super})) check with a tiny bit of metaprogramming, it gave me unexpected results.

Here's an example:

class A;
def self.def_f!;
singleton_class.send(:define_method, :f) { defined? super }
end
end
class AA < A; end

A.def_f!
A.f # => nil
AA.f # => nil
AA.def_f!
AA.f # => "super"
A.f # => "super" - WHY???

The last result is really strange, isn't it?
A.f has no super method, so I would expect the last (({A.f})) to return (({nil})).
Is it a bug?
=end

Related issues:

Related to Backport200 - Backport #8367: regression in defined?(super) starti... Closed 05/04/2013
Has duplicate Ruby master - Bug #6722: Weird behavior of defined?(super) chec... Closed 07/11/2012

Associated revisions

Revision 35784d10 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36369 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 36369 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]
Revision 36369 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]

Revision 36369 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]

Revision 36369 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]

Revision 36369 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]

Revision 36369 - 07/11/2012 08:11 PM - nobu (Nobuyoshi Nakada)
defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]

History

#1 - 06/29/2012 08:30 PM - alexisowl (Alexey Smolianinov)
Hello? Is anyone here? :)

#2 - 07/12/2012 05:09 AM - nobu (Nobuyoshi Nakada)
- Category set to core
- Target version changed from 1.9.3 to 2.0.0

#3 - 07/12/2012 05:11 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r36369.
Alexey, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

defined: me in cfp

- insns.def (defined): use method entry and id in cfp for proper superclass, since klass in iseq is shared by dynamically defined methods from the same block. [ruby-core:45831][Bug #6644]