

Ruby master - Bug #14670

Object method_missing super

04/08/2018 01:39 AM - keiju (Keiju Ishitsuka)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.3: DONTNEED, 2.4: DONTNEED, 2.5: DONTNEED
Description <pre>class Object def method_missing(name, *args, &block) super end end Object.new.foo</pre> <p>SystemStackError.</p> <p>Object(Object)</p>	
Related issues: Related to Ruby master - Feature #14313: Support creating KeyError with recei... Closed	

Associated revisions

Revision a72581d7 - 04/12/2018 03:48 AM - nobu (Nobuyoshi Nakada)

error.c: super in method_missing

- error.c (nometh_err_initialize): do not shortcut rb_call_super, to push proper control frame. [ruby-dev:50522] [Bug #14670]
- error.c (rb_nomethod_err_new): allocate and initialize a new NoMethodError instance.
- vm_eval.c (rb_make_no_method_exception): create a new exception instance directly without method calls, to prevent influence of ruby level method definitions, which can cause an unpredictable behavior, e.g., infinite recursion.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63136 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63136 - 04/12/2018 03:48 AM - nobu (Nobuyoshi Nakada)

error.c: super in method_missing

- error.c (nometh_err_initialize): do not shortcut rb_call_super, to push proper control frame. [ruby-dev:50522] [Bug #14670]
- error.c (rb_nomethod_err_new): allocate and initialize a new NoMethodError instance.
- vm_eval.c (rb_make_no_method_exception): create a new exception instance directly without method calls, to prevent influence of ruby level method definitions, which can cause an unpredictable behavior, e.g., infinite recursion.

Revision 63136 - 04/12/2018 03:48 AM - nobu (Nobuyoshi Nakada)

error.c: super in method_missing

- error.c (nometh_err_initialize): do not shortcut rb_call_super, to push proper control frame. [ruby-dev:50522] [Bug #14670]
- error.c (rb_nomethod_err_new): allocate and initialize a new NoMethodError instance.
- vm_eval.c (rb_make_no_method_exception): create a new exception instance directly without method calls, to prevent influence of ruby level method definitions, which can cause an unpredictable behavior, e.g., infinite recursion.

Revision 76e3af82 - 12/06/2018 11:06 PM - nobu (Nobuyoshi Nakada)

Warn redefinitions of some methods on Object

[Bug #5473] [Bug #14670] [Bug #15382]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66262 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 66262 - 12/06/2018 11:06 PM - nobu (Nobuyoshi Nakada)

Warn redefinitions of some methods on Object

[Bug #5473] [Bug #14670] [Bug #15382]

Revision 66262 - 12/06/2018 11:06 PM - nobu (Nobuyoshi Nakada)

Warn redefinitions of some methods on Object

[Bug #5473] [Bug #14670] [Bug #15382]

History

#1 - 04/10/2018 05:37 AM - nobu (Nobuyoshi Nakada)

- Related to Feature #14313: Support creating KeyError with receiver and key from Ruby added

#2 - 04/12/2018 02:18 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: DONTNEED, 2.4: DONTNEED, 2.5: DONTNEED

- Description updated

#3 - 04/12/2018 03:48 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset [trunk|r63136](#).

error.c: super in method_missing

- error.c (nometh_err_initialize): do not shortcut rb_call_super, to push proper control frame. [ruby-dev:50522] [Bug [#14670](#)]
- error.c (rb_nomethod_err_new): allocate and initialize a new NoMethodError instance.
- vm_eval.c (rb_make_no_method_exception): create a new exception instance directly without method calls, to prevent influence of ruby level method definitions, which can cause an unpredictable behavior, e.g., infinite recursion.