Ruby master - Feature #10881

NoMethodError#receiver

02/21/2015 08:22 PM - yuki24 (Yuki Nishijima)

<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></td>
</tr>
</tbody>
</table>

**Description**

I would like NoMethodError to have receiver method:

```ruby
receiver = "receiver"
exception = receiver.doesnt_exist rescue $!

exception.receiver == receiver #=> true
```

This helps did_you_mean gem implement its features without having native C exceptions, and also makes it easier to add "did you mean?" feature to irb. Let mw know what you think. Thanks!

**Associated revisions**

**Revision 51de63df - 06/18/2015 04:32 AM - nobu (Nobuyoshi Nakada)**

error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]

  git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50945 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 50945 - 06/18/2015 04:32 AM - nobu (Nobuyoshi Nakada)**

error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]

**Revision 50945 - 06/18/2015 04:32 AM - nobu (Nobuyoshi Nakada)**

error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]

**Revision 50945 - 06/18/2015 04:32 AM - nobu (Nobuyoshi Nakada)**

error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]

**Revision 50945 - 06/18/2015 04:32 AM - nobu (Nobuyoshi Nakada)**

error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]

**Revision 50945 - 06/18/2015 04:32 AM - nobu (Nobuyoshi Nakada)**

error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]

**Revision fac04ba9 - 06/18/2015 04:50 AM - nobu (Nobuyoshi Nakada)**

NEWS: NameError#receiver [Feature #10881]

[ci skip]

  git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@50946 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 50946 - 06/18/2015 04:50 AM - nobu (Nobuyoshi Nakada)**
NEWS: NameError#receiver [Feature #10881]

[ci skip]

Revision 50946 - 06/18/2015 04:50 AM - nobu (Nobuyoshi Nakada)
NEWS: NameError#receiver [Feature #10881]

[ci skip]

Revision 50946 - 06/18/2015 04:50 AM - nobu (Nobuyoshi Nakada)
NEWS: NameError#receiver [Feature #10881]

[ci skip]

Revision 50946 - 06/18/2015 04:50 AM - nobu (Nobuyoshi Nakada)
NEWS: NameError#receiver [Feature #10881]

[ci skip]

Revision 50946 - 06/18/2015 04:50 AM - nobu (Nobuyoshi Nakada)
NEWS: NameError#receiver [Feature #10881]

[ci skip]

Revision c23a7b0d - 06/23/2015 05:32 AM - nobu (Nobuyoshi Nakada)
error.c: ArgumentError if no receiver

- error.c (name_err_receiver): raise ArgumentError if no receiver is available on this exception object. [Feature #10881]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@51002 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 51002 - 06/23/2015 05:32 AM - nobu (Nobuyoshi Nakada)
error.c: ArgumentError if no receiver

- error.c (name_err_receiver): raise ArgumentError if no receiver is available on this exception object. [Feature #10881]

Revision 51002 - 06/23/2015 05:32 AM - nobu (Nobuyoshi Nakada)
error.c: ArgumentError if no receiver

- error.c (name_err_receiver): raise ArgumentError if no receiver is available on this exception object. [Feature #10881]

Revision 51002 - 06/23/2015 05:32 AM - nobu (Nobuyoshi Nakada)
error.c: ArgumentError if no receiver

- error.c (name_err_receiver): raise ArgumentError if no receiver is available on this exception object. [Feature #10881]

Revision 51002 - 06/23/2015 05:32 AM - nobu (Nobuyoshi Nakada)
error.c: ArgumentError if no receiver

- error.c (name_err_receiver): raise ArgumentError if no receiver is available on this exception object. [Feature #10881]

Revision 51002 - 06/23/2015 05:32 AM - nobu (Nobuyoshi Nakada)
error.c: ArgumentError if no receiver

- error.c (name_err_receiver): raise ArgumentError if no receiver is available on this exception object. [Feature #10881]

Revision 7a929727 - 10/28/2015 06:23 AM - nobu (Nobuyoshi Nakada)
error.c: rb_name_err_new

- error.c (rb_name_err_new): new function to create NameError exception instance. [Feature #10881]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@52320 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 52320 - 10/28/2015 06:23 AM - nobu (Nobuyoshi Nakada)
- object.c (check_setter_id): use rb_check_id to convert names.
- variable.c (uninitialized_constant): use NameError::message to keep the receiver of uninitialized constant. [Feature #10881]

Revision a4f838c7 - 10/28/2015 06:36 AM - nobu (Nobuyoshi Nakada)
error.c: receiver in NameError
- error.c (rb_name_err_new): store the receiver directly.
- error.c (name_err_receiver): return directly stored receiver. [Feature #10881]

Revision 52322 - 10/28/2015 06:36 AM - nobu (Nobuyoshi Nakada)
error.c: receiver in NameError
- error.c (rb_name_err_new): store the receiver directly.
- error.c (name_err_receiver): return directly stored receiver. [Feature #10881]

Revision 52322 - 10/28/2015 06:36 AM - nobu (Nobuyoshi Nakada)
error.c: receiver in NameError
- error.c (rb_name_err_new): store the receiver directly.
- error.c (name_err_receiver): return directly stored receiver. [Feature #10881]

Revision 52322 - 10/28/2015 06:36 AM - nobu (Nobuyoshi Nakada)
error.c: receiver in NameError
- error.c (rb_name_err_new): store the receiver directly.
- error.c (name_err_receiver): return directly stored receiver. [Feature #10881]

Revision 52322 - 10/28/2015 06:36 AM - nobu (Nobuyoshi Nakada)
error.c: receiver in NameError
- error.c (rb_name_err_new): store the receiver directly.
- error.c (name_err_receiver): return directly stored receiver. [Feature #10881]

Revision 52322 - 10/28/2015 06:36 AM - nobu (Nobuyoshi Nakada)
error.c: receiver in NameError
- error.c (rb_name_err_new): store the receiver directly.
- error.c (name_err_receiver): return directly stored receiver. [Feature #10881]

Revision a2369fc2 - 10/28/2015 07:56 AM - nobu (Nobuyoshi Nakada)
encoding for inspect
- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]

Revision 52323 - 10/28/2015 07:56 AM - nobu (Nobuyoshi Nakada)
encoding for inspect
- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]

Revision 52323 - 10/28/2015 07:56 AM - nobu (Nobuyoshi Nakada)
encoding for inspect
- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]

Revision 52323 - 10/28/2015 07:56 AM - nobu (Nobuyoshi Nakada)
encoding for inspect
- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]

Revision 52323 - 10/28/2015 07:56 AM - nobu (Nobuyoshi Nakada)
encoding for inspect
- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]
Encoding for inspect

- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]

Revision 52323 - 10/28/2015 07:56 AM - nobu (Nobuyoshi Nakada)
Encoding for inspect

- test/ruby/test_{exception,module,object}.rb: inspect result depend on the default external encoding. [Feature #10881]

History

#1 - 06/17/2015 02:04 PM - matz (Yukihiro Matsumoto)
Looks OK for me. I am interested in how JRuby/Rubinius guys feel about the proposal.

Matz.

#2 - 06/17/2015 11:41 PM - yuki24 (Yuki Nishijima)
Thanks Matsumoto-san.

I realized that the last line in the example above is wrong, it has to use #equal? instead of #==:

```ruby
exception.receiver.equal?(receiver) # => true
```

#3 - 06/18/2015 04:33 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset r50945.

---

```
error.c: NameError#receiver

- error.c (name_err_receiver): add NameError#receiver method. [Feature #10881]
```