

Ruby trunk - Feature #12041

Change the initializer of NameError to take a receiver as the third argument

02/01/2016 11:42 AM - yuki24 (Yuki Nishijima)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>I would like to change NameError#initialize to take a receiver object as the third argument. An example would be like this:</p> <pre>receiver_object = Object.new name_error = NameError.new("Error message", "name", receiver_object) name_error.receiver.equal?(receiver_object) # => true</pre> <p>The reason I wanted this change is Rails overrides Module#const_missing and raises an NameError without a receiver object. It has actually affected one of the did_you_mean's spell checkers and thus no suggestions for constant names will be made on Rails. One more use case I can think of is when there's a class that inherits from NameError or NoMethodError and an arbitrary receiver needs to be passed to the initializer of the class.</p> <p>Please let me know if you have any thoughts.</p>	

History

#1 - 02/01/2016 12:25 PM - nobu (Nobuyoshi Nakada)

What's about NoMethodError?

Currently, it is

```
NoMethodError.new(msg, name [, args]) #-> no_method_error
```

If NameError#initialize has third argument, where will it be placed?

The last?

```
NoMethodError.new(msg, name [, args [,receiver]]) #-> no_method_error
```

or same as NameError#initialize?

```
NoMethodError.new(msg, name [, receiver [, args]]) #-> no_method_error
```

A patch for the latter.

```
diff --git i/error.c w/error.c
index f88c19a..0ae9b58 100644
--- i/error.c
+++ w/error.c
@@ -1127,7 +1127,7 @@ rb_name_error_str(VALUE str, const char *fmt, ...)
/*
 * call-seq:
- *   NameError.new(msg [, name]) -> name_error
+ *   NameError.new(msg [, name [, receiver]]) -> name_error
 *
 * Construct a new NameError exception. If given the <i>name</i>
 * parameter may subsequently be examined using the <code>NameError.name</code>
@@ -1138,8 +1138,10 @@ static VALUE
name_err_initialize(int argc, VALUE *argv, VALUE self)
{
  VALUE name;
+  VALUE recv;
  VALUE iseqw = Qnil;

+  recv = (argc > 2) ? argv[--argc] : Qundef;
  name = (argc > 1) ? argv[--argc] : Qnil;
  rb_call_super(argc, argv);
  rb_ivar_set(self, id_name, name);
```

```

@@ -1150,6 +1152,7 @@ name_err_initialize(int argc, VALUE *argv, VALUE self)
     if (cfp) iseql = rb_iseql_new(cfp->iseql);
     }
     rb_ivar_set(self, id_iseql, iseql);
+   if (recv != Qundef) rb_ivar_set(self, id_receiver, recv);
     return self;
 }

@@ -1192,7 +1195,7 @@ name_err_local_variables(VALUE self)

/*
 * call-seq:
- *   NoMethodError.new(msg, name [, args]) -> no_method_error
+ *   NoMethodError.new(msg, name [, receiver [, args]]) -> no_method_error
 *
 * Construct a NoMethodError exception for a method of the given name
 * called with the given arguments. The name may be accessed using
diff --git i/test/ruby/test_exception.rb w/test/ruby/test_exception.rb
index 91732dd..e546988 100644
--- i/test/ruby/test_exception.rb
+++ w/test/ruby/test_exception.rb
@@ -699,6 +699,11 @@
     assert_equal(:foo, e.name)
     assert_same(obj, e.receiver)
     assert_equal(%i[a b c d e f g], e.local_variables.sort)
+
+   obj = Object.new
+   e = NameError.new("error", :foo, obj)
+   assert_equal(:foo, e.name)
+   assert_same(obj, e.receiver)
   end

   def test_name_error_info_parent_iseq_mark

```

#2 - 02/02/2016 01:39 PM - yuki24 (Yuki Nishijima)

Speaking of NoMethodError, I think the former (NoMethodError.new(msg, name [, args [,receiver]])) makes more sense to me since the latter may break existing apps.

#3 - 02/03/2016 05:35 AM - nobu (Nobuyoshi Nakada)

Nobuyoshi Nakada wrote:

or same as NameError#initialize?

```
NoMethodError.new(msg, name [, receiver [, args]]) #-> no_method_error
```

This was wrong.

The previous patch is:

```
NoMethodError.new(msg, name [[, receiver], args]) #-> no_method_error
```

#4 - 02/03/2016 05:44 AM - nobu (Nobuyoshi Nakada)

<https://github.com/ruby/ruby/compare/trunk...nobu:feature/12041-NameError%23initialize-receiver>

#5 - 02/12/2016 05:37 AM - nobu (Nobuyoshi Nakada)

I found this causes drb test failures, since NoMethodError now refers DRbObject and Marshal.dump fails.