

Ruby trunk - Feature #14313

Support creating KeyError with receiver and key from Ruby

01/04/2018 05:03 AM - kou (Kouhei Sutou)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
<p>KeyError has readers for error details, receiver and key. They are convenient to process KeyError.</p> <p>We can set receiver and key to KeyError by <code>rb_key_err_new()</code> in C. But we can't set them in Ruby. Because receiver and key use no <code>@</code> instance variables.</p> <p>How about adding <code>KeyError#initialize</code> that accepts receiver and key? Because KeyError is useful in pure Ruby libraries. For example, csv library uses KeyError: https://svn.ruby-lang.org/cgi-bin/viewvc.cgi/trunk/lib/csv.rb?revision=59849&view=markup#l321</p> <pre>def fetch(header, *varargs) # ... raise KeyError, "key not found: #{header}" # ... end</pre> <p>I want to use KeyError for Hash like objects such as <code>Arrow::Table</code> in Red Arrow.</p> <p>The attached patch adds <code>KeyError#initialize</code> that behaves as the following:</p> <pre>p KeyError.new.message # => "KeyError": No change. Keep backward compatibility. p KeyError.new("Message").message # => "Message": No change. Keep backward compatibility. p KeyError.new(receiver: Object.new).receiver # => The Object instance. p KeyError.new(key: :unknown_key).key # => :unknown_key key_error = KeyError.new(receiver: Object.new, key: :unknown_key) p key_error.receiver # => The Object instance. p key_error.key # => :unknown_key key_error = KeyError.new("Message", receiver: Object.new, key: :unknown_key) p key_error.message # => "Message" p key_error.receiver # => The Object instance. p key_error.key # => :unknown_key</pre>		
Related issues:		
Related to Ruby trunk - Bug #14670: Object#method_missing#super		Closed

Associated revisions

Revision [cd83d267](#) - 01/26/2018 04:51 AM - kou (Kouhei Sutou)

KeyError#initialize accepts receiver and key.
[Feature #14313][ruby-core:84626]

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@62049](https://ci.ruby-lang.org/ruby/trunk@62049) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62049 - 01/26/2018 04:51 AM - kou (Kouhei Sutou)

KeyError#initialize accepts receiver and key.
[Feature #14313][ruby-core:84626]

Revision 62049 - 01/26/2018 04:51 AM - kou (Kouhei Sutou)

KeyError#initialize accepts receiver and key.
[Feature #14313][ruby-core:84626]

Revision b56f6a6b - 01/26/2018 05:34 AM - nobu (Nobuyoshi Nakada)

error.c: use already initialized IDs
[Feature #14313]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62050 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62050 - 01/26/2018 05:34 AM - nobu (Nobuyoshi Nakada)

error.c: use already initialized IDs
[Feature #14313]

Revision 62050 - 01/26/2018 05:34 AM - nobu (Nobuyoshi Nakada)

error.c: use already initialized IDs
[Feature #14313]

Revision ba4aba0d - 01/26/2018 06:41 AM - nobu (Nobuyoshi Nakada)

error.c: KeyError missing keyword arguments

- error.c (key_err_initialize): leave attributes for missing keyword arguments unset, so accessors can tell if it is missing or explicit nil. [Feature #14313]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62053 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62053 - 01/26/2018 06:41 AM - nobu (Nobuyoshi Nakada)

error.c: KeyError missing keyword arguments

- error.c (key_err_initialize): leave attributes for missing keyword arguments unset, so accessors can tell if it is missing or explicit nil. [Feature #14313]

Revision 62053 - 01/26/2018 06:41 AM - nobu (Nobuyoshi Nakada)

error.c: KeyError missing keyword arguments

- error.c (key_err_initialize): leave attributes for missing keyword arguments unset, so accessors can tell if it is missing or explicit nil. [Feature #14313]

Revision 98c88786 - 01/26/2018 10:55 AM - nobu (Nobuyoshi Nakada)

error.c: copy keyword arguments

- error.c (rb_key_err_new): pass arguments all arguments to the super method, except for keyword arguments copied to instance variables. [Feature #14313]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62054 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62054 - 01/26/2018 10:55 AM - nobu (Nobuyoshi Nakada)

error.c: copy keyword arguments

- error.c (rb_key_err_new): pass arguments all arguments to the super method, except for keyword arguments copied to instance variables. [Feature #14313]

Revision 62054 - 01/26/2018 10:55 AM - nobu (Nobuyoshi Nakada)

error.c: copy keyword arguments

- error.c (rb_key_err_new): pass arguments all arguments to the super method, except for keyword arguments copied to instance variables. [Feature #14313]

Revision c30aed08 - 01/26/2018 10:55 AM - nobu (Nobuyoshi Nakada)

error.c: receiver kwarg

- error.c (name_err_initialize_options): NameError#initialize accepts receiver. [Feature #14313]
- error.c (nometh_err_initialize_options): pass keyword arguments to the super method.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62057 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62057 - 01/26/2018 10:55 AM - nobu (Nobuyoshi Nakada)

error.c: receiver kwarg

- error.c (name_err_initialize_options): NameError#initialize accepts receiver. [Feature #14313]
- error.c (nometh_err_initialize_options): pass keyword arguments to the super method.

Revision 62057 - 01/26/2018 10:55 AM - nobu (Nobuyoshi Nakada)

error.c: receiver kwarg

- error.c (name_err_initialize_options): NameError#initialize accepts receiver. [Feature #14313]
- error.c (nometh_err_initialize_options): pass keyword arguments to the super method.

History

#1 - 01/05/2018 03:51 AM - nobu (Nobuyoshi Nakada)

Also NameError?

#2 - 01/05/2018 09:18 AM - Hanmac (Hans Mackowiak)

should that also extend to the raise function itself like this?

```
raise KeyError, "key not found: #{header}", key: :header
```

or is that already done?

#3 - 01/07/2018 06:18 AM - kou (Kouhei Sutou)

NameError already supports assigning name:

```
p NameError.new("message", :x).name # => :x
```

Ah, did you mention receiver not name?

#4 - 01/07/2018 06:22 AM - kou (Kouhei Sutou)

Kernel.#raise expects backtrace for the 3rd argument.

If we want to extend the current Kernel.#raise behavior, we should create a new issue instead of discussing in this issue.

#5 - 01/24/2018 06:28 AM - matz (Yukihiro Matsumoto)

I agree with the original proposal. We are not going to enhance raise behavior (yet).

Matz.

#6 - 01/26/2018 04:51 AM - kou (Kouhei Sutou)

- Status changed from Open to Closed

Applied in changeset [trunk|r62049](#).

KeyError#initialize accepts receiver and key.
[Feature #14313][ruby-core:84626]

#7 - 01/26/2018 05:38 AM - nobu (Nobuyoshi Nakada)

Shouldn't unspecified attribute raise an exception?

```
diff --git c/error.c i/error.c
index 990f9f7855..7870e58035 100644
--- c/error.c
+++ i/error.c
@@ -1692,8 +1692,6 @@ key_err_initialize(int argc, VALUE *argv, VALUE self)
 {
     VALUE message;
     VALUE options;
-    VALUE receiver = Qnil;
-    VALUE key = Qnil;

     rb_scan_args(argc, argv, "01:", &message, &options);

@@ -1711,16 +1709,13 @@ key_err_initialize(int argc, VALUE *argv, VALUE self)
     keywords[1] = id_key;
     rb_get_kwargs(options, keywords, 0, 2, values);
     if (values[0] != Qundef) {
-        receiver = values[0];
+        rb_ivar_set(self, id_receiver, values[0]);
     }
     if (values[1] != Qundef) {
-        key = values[1];
+        rb_ivar_set(self, id_key, values[1]);
     }
 }

-    rb_ivar_set(self, id_receiver, receiver);
-    rb_ivar_set(self, id_key, key);
-
     return self;
 }

diff --git c/test/ruby/test_key_error.rb i/test/ruby/test_key_error.rb
index 852bd32712..fe1d5bb5ab 100644
--- c/test/ruby/test_key_error.rb
+++ i/test/ruby/test_key_error.rb
@@ -15,11 +15,15 @@
     receiver = Object.new
     error = KeyError.new(receiver: receiver)
     assert_equal(receiver, error.receiver)
+    error = KeyError.new
+    assert_raise(ArgumentError) {error.receiver}
 end

 def test_key
     error = KeyError.new(key: :key)
     assert_equal(:key, error.key)
+    error = KeyError.new
+    assert_raise(ArgumentError) {error.key}
 end

 def test_receiver_and_key
```

#8 - 04/10/2018 05:37 AM - nobu (Nobuyoshi Nakada)

- Related to Bug #14670: Object#method_missing super added

Files

key_error_new.diff	2.91 KB	01/04/2018	kou (Kouhei Sutou)
--------------------	---------	------------	--------------------