

Ruby - Bug #1843

Symbol#inspect raises exception

07/31/2009 02:22 AM - candlerb (Brian Candler)

Status:	Closed	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
ruby -v:	ruby 1.9.2dev (2009-07-18 trunk 24186) [i686-linux]	Backport:

Description

=begin

If you can create an object, I think you should always be able to inspect it.

However Symbol#inspect raises an exception if it was made from a string with an invalid encoding.

```
"hello\xff".to_sym.inspect
ArgumentError: invalid byte sequence in UTF-8
from (irb):26:in inspect' from (irb):26 from /usr/local/bin/irb19:12:in '
```

Note that the to_sym is quite happy; it's the inspect which fails.

```
"hello\xff".to_sym; nil
=> nil
```

This means you cannot handle such objects in irb.

```
s = :"hello\xff"
ArgumentError: invalid byte sequence in UTF-8
from /usr/local/lib/ruby/1.9.1/irb/inspector.rb:84:in inspect' from /usr/local/lib/ruby/1.9.1/irb/inspector.rb:84:in block in
module:IRB'
from /usr/local/lib/ruby/1.9.1/irb/inspector.rb:30:in call' from /usr/local/lib/ruby/1.9.1/irb/inspector.rb:30:in inspect_value'
from /usr/local/lib/ruby/1.9.1/irb/context.rb:259:in inspect_last_value' from /usr/local/lib/ruby/1.9.1/irb.rb:301:in
output_value'
from /usr/local/lib/ruby/1.9.1/irb.rb:150:in block (2 levels) in eval_input' from /usr/local/lib/ruby/1.9.1/irb.rb:263:in
signal_status'
from /usr/local/lib/ruby/1.9.1/irb.rb:146:in block in eval_input' from /usr/local/lib/ruby/1.9.1/irb/ruby-lex.rb:244:in block (2
levels) in each_top_level_statement'
from /usr/local/lib/ruby/1.9.1/irb/ruby-lex.rb:230:in loop' from /usr/local/lib/ruby/1.9.1/irb/ruby-lex.rb:230:in block in
each_top_level_statement'
from /usr/local/lib/ruby/1.9.1/irb/ruby-lex.rb:229:in catch' from /usr/local/lib/ruby/1.9.1/irb/ruby-lex.rb:229:in
each_top_level_statement'
from /usr/local/lib/ruby/1.9.1/irb.rb:145:in eval_input' from /usr/local/lib/ruby/1.9.1/irb.rb:69:in block in start'
from /usr/local/lib/ruby/1.9.1/irb.rb:68:in catch' from /usr/local/lib/ruby/1.9.1/irb.rb:68:in start'
from /usr/local/bin/irb19:12:in ``Maybe IRB bug!!
```

=end

History

#1 - 07/31/2009 02:41 AM - shyouhei (Shyouhei Urabe)

=begin

My impression is that such symbols should be forbidden to exist.

=end

#2 - 11/04/2009 10:50 PM - naruse (Yui NARUSE)

- Category set to core
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)

=begin
I agree with shyouhei.
=end

#3 - 03/01/2010 11:57 PM - mame (Yusuke Endoh)

=begin
Hi,

Brian Candler wrote:

However Symbol#inspect raises an exception if it was made from a string with an invalid encoding.

Shyouhei Urabe wrote:

My impression is that such symbols should be forbidden to exist.

Yui NARUSE wrote:

I agree with shyouhei.

I guess we have already agreed the conclusion. Must we wait for matz?

```
diff --git a/parse.y b/parse.y
index e338c6f..6a0e431 100644
--- a/parse.y
+++ b/parse.y
@@ -9505,6 +9505,10 @@ rb_intern3(const char *name, long len, rb_encoding *enc)
str = (VALUE)&fake_str;
rb_enc_associate(str, enc);
```

- if (rb_enc_str_coderange(str) == ENC_CODERANGE_BROKEN) {
- [REDACTED]
- }
- if (st_lookup(global_symbols.sym_id, str, (st_data_t *)&id))
return id;

```
diff --git a/test/ruby/test_m17n_comb.rb b/test/ruby/test_m17n_comb.rb
index ab89136..cf80377 100644
--- a/test/ruby/test_m17n_comb.rb
+++ b/test/ruby/test_m17n_comb.rb
@@ -1040,10 +1040,12 @@ class TestM17NComb < Test::Unit::TestCase
STRINGS.each {|s|
if /\0/ =~ a(s)
assert_raise(ArgumentError) { s.intern }
```

- [REDACTED]
- [REDACTED]

```
sym = s.intern
assert_equal(s, sym.to_s, "#{encdump s}.intern.to_s")
assert_equal(sym, s.to_sym)
```
- [REDACTED]
- [REDACTED]

```
end

}
end
```

--

Yusuke ENDOH mame@tsg.ne.jp
=end

#4 - 03/16/2010 05:48 PM - matz (Yukihiro Matsumoto)

=begin
Hi,

In message "Re: [\[ruby-core:28384\]](#) [Bug [#1843](#)] Symbol#inspect raises exception"
on Mon, 1 Mar 2010 23:58:47 +0900, Yusuke Endoh redmine@ruby-lang.org writes:

|Brian Candler wrote:

|> However Symbol#inspect raises an exception if it was made from a string with an invalid encoding.

|

|Shyouhei Urabe wrote:

|> My impression is that such symbols should be forbidden to exist.

|

|Yui NARUSE wrote:

|> I agree with shyouhei.

|

|I guess we have already agreed the conclusion. Must we wait for matz?

OK, OK. Go ahead and check it in.

matz.

=end

#5 - 03/17/2010 01:27 AM - mame (Yusuke Endoh)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

=begin

This issue was solved with changeset r26957.

Brian, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

=end

#6 - 03/17/2010 02:51 AM - mame (Yusuke Endoh)

=begin

2010/3/16 Yukihiro Matsumoto matz@ruby-lang.org:

OK, OK. Go ahead and check it in.

Done. Thanks!

--

Yusuke ENDOH mame@tsg.ne.jp

=end