Ruby master - Bug #5304

Array#pack handles objects for eg format 'E' differently than 1.8

09/10/2011 05:32 AM - brixen (Brian Shirai)

Status:	Rejected		
Priority:	Normal		
Assignee:	mrkn (Kenta Murata)		
Target version:			
ruby -v:	ruby 1.9.2p290 (2011-07-09 revision 32553) [x86_64-darwin10.8.0]	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description

In 1.9, Array#pack when passed an object that defines #to_f and one of the Float formats raises a TypeError. This is a change from 1.8. The code in 1.9 calls rb_to_float()

1.9

```
VALUE
rb_to_float(VALUE val)
{
if (TYPE(val) == T_FLOAT) return val;
if (!rb_obj_is_kind_of(val, rb_cNumeric)) {
rb_raise(rb_eTypeError, "can't convert %s into Float",
NIL_P(val) ? "nil" :
val == Qtrue ? "true" :
val == Qfalse ? "false" :
rb_obj_classname(val));
}
return rb_convert_type(val, T_FLOAT, "Float", "to_f");
}
```

The rb_to_float() imposes the (arbitrary) restriction that the object be kind of Numeric. The 1.8 pack code used rb_Float(), which merely requires the object respond to #to_f and return a Float.

The treatment of objects in 1.9 for the float formats is thus different than for eg the integer formats, where the object simply must respond to #to_int. The requirement that the object be a kind of Numeric breaks the uniform ducktyping and imposes an arbitrary and unnecessary type/class requirement.

Was this change intended, and if so, why? Apologies if this was discussed on ruby-core prior to the change; I could not find any discussion through searching.

The following code illustrates the issue:

```
o = Object.new

def o.to_int
puts "to_int called"
1
end

def o.to_f
puts "to_f called"
1.1
end

p [o].pack("I") # => "\001..." (depends on endianness)
p [o].pack("E") # => 1.8 - ""; 1.9 - TypeError
```

```
$ ruby1.8.7 -v pack_test.rb
ruby 1.8.7 (2011-02-18 patchlevel 334) [i686-darwin10.8.0]
to_int called
"\001\000\000\000"
```

```
to_f called
"\232\231\231\231\231\231\361?"

$ ruby1.9.2 -v pack_test.rb
ruby 1.9.2p290 (2011-07-09 revision 32553) [x86_64-darwin10.8.0]
to_int called
"\x01\x00\x00\x00"
pack_test.rb:14:in pack': can't convert Object into Float (TypeError)
from pack_test.rb:14:in'
```

History

#1 - 03/11/2012 03:33 PM - ko1 (Koichi Sasada)

- Status changed from Open to Assigned

- Assignee set to mrkn (Kenta Murata)

Please discuss with matz.
However, I assign this ticket to Murata-san, because matz will ignore assignment.

Murata-san:
Please lead discussion about this issue.

#2 - 03/28/2018 07:41 AM - mrkn (Kenta Murata)

- Status changed from Assigned to Rejected

Rejected because this had already been meaningless now.