

Ruby master - Bug #15432

Float NaN

12/18/2018 08:46 AM - shuujii (Shuji KOBAYASHI)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.6.0rc2 (2018-12-15 trunk 66408) [x86_64-linux]	Backport: 2.4: UNKNOWN, 2.5: UNKNOWN

Description

Float NaN

```
## spec/ruby/core/array/equal_value_spec.rb:47
```

```
# As per bug #1720
it "returns false for [NaN] == [NaN]" do
  [nan_value].should_not == [nan_value]
end
```

[#1720](#)

```
[Float::NaN] == [Float::NaN] #=> true
[Float::NaN] == [0/0.0]      #=> false
```

Associated revisions

Revision a0af60c7 - 06/19/2019 07:20 PM - jeremyevans (Jeremy Evans)

Remove spec testing undefined behavior

Fixes [\[Bug #15432\]](#)

History

#1 - 06/13/2019 01:45 AM - jeremyevans0 (Jeremy Evans)

I agree that keeping this spec does not make sense. If nobody objects within the next week, I will delete the spec.

#2 - 06/19/2019 07:21 PM - jeremyevans (Jeremy Evans)

- Status changed from Open to Closed

Applied in changeset [git|a0af60c7f2d852faa6d3263874224dd7950bda43](https://github.com/ruby/ruby/commit/a0af60c7f2d852faa6d3263874224dd7950bda43).

Remove spec testing undefined behavior

Fixes [\[Bug #15432\]](#)

#3 - 06/27/2019 05:44 PM - Eregon (Benoit Daloze)

It makes sense in the context of [#1720](#).

I.e., all Ruby implementations behave that way, because `Float::NaN.equal? Float::NaN` must be true, and `Array#==` uses an `#equal?` + a `#==` check, not just `#==`. I want to restore the spec and add that explanation.

IMHO there is no "undefined behavior" in Ruby, there is MRI behavior and that's what other Ruby implementations have to comply to, unless proven a bug or desirable to differ.

#4 - 06/27/2019 05:58 PM - Eregon (Benoit Daloze)

Actually, the `should_not` above indeed doesn't make much sense, because it relies on `nan_value` returning different NaNs, which is probably platform-specific.

I added a spec for this case: [Float::NAN].should == [Float::NAN].