

Ruby trunk - Bug #5179

Complex#rationalize and to_r with approximate zeros

08/10/2011 10:41 AM - marcandre (Marc-Andre Lafortune)

Status: Assigned	
Priority: Normal	
Assignee: mrkn (Kenta Murata)	
Target version:	
ruby -v: r32354	Backport:
Description	
Currently, Complex#rationalize and Complex#to_r raise a RangeError if the imaginary part is nonzero <i>or is a Float</i> . Note that a BigDecimal(0) is accepted, though:	
<pre>Complex(1, 0).to_r # => Rational(1,1) Complex(1, BigDecimal("0.0")).to_r # => Rational(1,1) Complex(1, 0.0).to_r # => RangeError</pre>	
This is inconsistent. I recommend not raising an error for 0.0 (Float or BigDecimal). Any objection?	
Related issues:	
Blocked by Ruby trunk - Feature #5321: Introducing Numeric#exact? and Numeric...	Assigned

History

#1 - 08/10/2011 07:20 PM - mrkn (Kenta Murata)

- Assignee set to mrkn (Kenta Murata)

0.0 doesn't exactly represent zero. It may be $0.0 \pm 10.0^{*(\text{Float}::\text{MIN_10_EXP}-17)}$.
BigDecimal(0) doesn't exactly represent zero, too.

I believe this issue should be resolved by introducing Numeric#exact? and/or Numeric#inexact? methods.

#2 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned