

Ruby master - Bug #13062

2.4.0-rc1 BigDecimal regression - to_d inconsistent with other numeric classes

12/22/2016 01:34 AM - jhawthorn (John Hawthorn)

Status:	Closed	
Priority:	Normal	
Assignee:	mrkn (Kenta Murata)	
Target version:		
ruby -v:	2.4.0-rc1	Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

To solve [#10286](#), `BigDecimal()` was changed to raise exceptions on invalid input, which is more consistent with the other numeric types.

Unfortunately, `String#to_d` now also raises errors, which is inconsistent with the other `to_*` methods (`to_i`, `to_f`), which return 0 on error.

Under ruby 2.4.0-rc1:

```
> require 'bigdecimal'
> require 'bigdecimal/util'
> "invalid".to_d
ArgumentError: invalid value for BigDecimal(): "invalid"
> "invalid".to_i
=> 0
> "invalid".to_f
=> 0.0
```

Under ruby 2.3.3 or 2.4.0preview3:

```
> "invalid".to_d
=> #<BigDecimal:55871ca1f808,'0.0',9(9)>
> "invalid".to_i
=> 0
> "invalid".to_f
=> 0.0
```

There's also a further problem that `BigDecimal()` still doesn't behave the same as `Integer()` when given a string with the number at the start:

Under ruby 2.4.0-rc1:

```
> BigDecimal("2 turtle doves")
=> 0.2e1
> Integer("2 turtle doves")
ArgumentError: invalid value for Integer(): "2 turtle doves"
> Float("2 turtle doves")
ArgumentError: invalid value for Float(): "2 turtle doves"
```

So `BigDecimal` is still inconsistent.

History

#1 - 12/25/2016 12:17 AM - Esse (Piotr Szmielew)

- File `fix_13062_string_to_d.patch` added

In attachment you will find patch for this issue.

This patch fixes issue with `String#to_d` method which was inconsistent with other `to_d` methods (`String` one returned error, while other simply returns 0).

Also, there tests added for this case.

#2 - 01/16/2017 03:14 AM - mrkn (Kenta Murata)

- Assignee set to mrkn (Kenta Murata)
- Status changed from Open to Assigned

We continue this issue at <https://github.com/ruby/bigdecimal/issues/51>.

#3 - 02/22/2017 01:27 AM - mrkn (Kenta Murata)

- Status changed from Assigned to Closed

Fixed in bigdecimal 1.3.1 and ruby trunk.

#4 - 02/24/2017 04:24 PM - rafaelfranca (Rafael França)

Is this going to be backported to 2.4?

#5 - 03/23/2017 07:44 AM - abezzub (Andrew Bezzub)

I can reproduce this in 2.4.1. Is this going to be fixed in 2.4.2? This is a significant issue that makes it hard to upgrade to Ruby 2.4.

#6 - 03/24/2017 12:27 AM - hsbt (Hiroshi SHIBATA)

You can install and use bigdecimal-1.3.1 via standalone gem with Ruby 2.4

#7 - 03/24/2017 12:37 AM - abezzub (Andrew Bezzub)

1.3.1 still had the problem, but 1.3.2 worked, thanks! It would still be nice if this was ported over to 2.4.x

Files

fix_13062_string_to_d.patch	1.57 KB	12/25/2016	Esse (Piotr Szmielew)
-----------------------------	---------	------------	-----------------------