Ruby master - Bug #13331

Inconsistent conversion of Float into Decimal

03/19/2017 06:55 PM - msatkiewicz (Maciej Satkiewicz)

Status:	Assigned		
Priority:	Normal		
Assignee:	mrkn (Kenta Murata)		
Target version:			
ruby -v:	ruby 2.4.0	**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

One would reasonably expect the equality 1.to_d * f == f.t_d to hold for every Float value f, i.e. multiplying by DigDecimal's unit can be viewed as an alternative way of casting Float into Decimal; or simply one could expect that BigDecimal#* should always invoke Float#to_d on it's argument. Yet for some Floats (e.g. 64.4) the results differ, as showed in the provided script. This can cause a bug in some application contexts.

Originally I asked about that on Stackoverflow:
http://stackoverflow.com/questions/40472933/inconsistent-conversion-of-float-into-decimal-in-ruby/40473007

See the comprehensive answer by Stefan who points out that it is due to different precisions being used in Float#to_d and BigDecimal#*.

History

#1 - 03/19/2017 09:57 PM - msatkiewicz (Maciej Satkiewicz)

- Description updated

#2 - 03/29/2017 02:16 PM - mrkn (Kenta Murata)

- Assignee set to mrkn (Kenta Murata)

- Status changed from Open to Assigned

Continue this in https://github.com/ruby/bigdecimal/issues/70

Files

multiplication_by_unit.rb	1.9 KB	03/19/2017	msatkiewicz (Maciej Satkiewicz)