

## QGIS Application - Bug report #14036

### graduated rendering: classes' upper and lower values precision truncated when saving to DOM

2015-12-25 01:04 AM - Mathieu Pellerin - nIRV

<b>Status:</b> Closed	
<b>Priority:</b> High	
<b>Assignee:</b>	
<b>Category:</b> Symbology	
<b>Affected QGIS version:</b> master	<b>Regression?:</b> No
<b>Operating System:</b>	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	<b>Resolution:</b> fixed/implemented
<b>Crashes QGIS or corrupts data:</b>	<b>Copied to github as #:</b> 22049

#### Description

This one is quite odd, and can be quite tricky for users (as it will change the map symbology rendering on project re-opening).

~~Basically, when using the natural break classification against a field for graduated rendering, the classes' range are constantly changing every time you hit the classify button. It's a problem as whatever is causing this will silently lead to a visually different symbology output when re-opening a project. (See below for correct description of the issue)~~

It's slightly difficult to explain, hoping the test project attached will make it more obvious.

#### Steps to reproduce:

1. Open the attached natural\_break\_issue project
2. Open the property window for the "communes" layer and go to the symbology panel
3. Hit the "classify" button once, note how the values of classes are changing
4. Hit the "classify" button again, note how the values of classes are still changing
5. Hit the button again and again to see values changing seemingly randomly.

#### History

##### #1 - 2015-12-25 01:18 AM - Mathieu Pellerin - nIRV

- File natural\_break\_issue-v2.zip added

- Subject changed from graduated rendering: natural break class values change on same expression at every classification to graduated rendering: natural break class values change on field at every classification

##### #2 - 2015-12-25 01:18 AM - Mathieu Pellerin - nIRV

- File deleted (natural\_break\_issue.zip)

##### #3 - 2015-12-25 07:58 PM - Mathieu Pellerin - nIRV

Ok, I figured out where the problem was, and opened a PR which fixes things (see <https://github.com/qgis/QGIS/pull/2613>).

First, TIL, the seemingly random upper and lower class values are actually by design as the natural break algorithm only takes a random sample.

Now, as for the problem: when saving the graduated symbology settings to the DOM, the code would truncate the lower and upper double values to 6 decimals, which is a problem when (as it is the case with the attached sample project) your double values have a higher precision than 6.

Take for e.g. those three values: 0.00000101, 0.00002301, 0.0000501 -- since the upper double value of the highest class would truncate to 6 decimals, 0.000050, the feature with a value of 0.0000501 would not be rendered when re-opening a project.

##### #4 - 2015-12-25 08:02 PM - Mathieu Pellerin - nIRV

- Subject changed from graduated rendering: natural break class values change on field at every classification to graduated rendering: classes' upper

and lower values precision truncated when saving to DOM

**#5 - 2015-12-25 09:14 PM - Mathieu Pellerin - nIRV**

- Status changed from Open to Closed
- % Done changed from 0 to 100
- Resolution set to fixed/implemented

Fixed in commit:8ff4c78.

**#6 - 2015-12-26 04:51 AM - Regis Haubourg**

I already funded the same bug in early 2.x versions. We absolutely need a unit test here.

**#7 - 2015-12-26 04:59 AM - Mathieu Pellerin - nIRV**

Regis, I noticed that the natural break function will only deal with the full dataset if it has less than 1,000 features. That limit was there since this classification was added 5 years ago. I am wondering whether we should raise that limit now. CPUs have gotten much better & I feel 1,000 is unreasonably low nowadays.

Thoughts?

Full disclosure: I have a project dealing with a dataset of about 1,600 features and I would love to have it dealt with in its entirety :)

**Files**

---

natural_break_issue-v2.zip	4.02 MB	2015-12-25	Mathieu Pellerin - nIRV
----------------------------	---------	------------	-------------------------