

QGIS Application - Bug report #15982

Data not being saved to PostGIS database consistently

2016-12-13 12:15 PM - John Zielinski

Status: Closed	
Priority: Normal	
Assignee:	
Category: Data Provider/PostGIS	
Affected QGIS version: 2.18.0	Regression?: No
Operating System:	Easy fix?: No
Pull Request or Patch supplied: No	Resolution: end of life
Crashes QGIS or corrupts data: No	Copied to github as #: 23898

Description

I'm using QGIS Desktop with a PostGIS database. I've created a polygon with 4 points (A to D) and then another polygon inside it with 3 points. Those three points were then snapped to 3 of the points in the first polygon (A, C and D). The vertex editor shows this (I've changed the first column here to label the points):

```
1A -127.524251009317 51.4891731834931
1B -127.556950974351 51.445477703804
1C -127.563090533222 51.4681981465143
1D -127.572335727596 51.4754127899731
1A -127.524251009317 51.4891731834931
```

```
2A -127.524251009317 51.4891731834931
2D -127.572335727596 51.4754127899731
2C -127.563090533222 51.4681981465143
2A -127.524251009317 51.4891731834931
```

As you can see 1A = 2A, 1C = 2C and 1D = 2D. In my database however:

```
1A -127.52425100931688 51.489173183493094
1B -127.55695097435051 51.44547770380399
1C -127.5630905332221 51.46819814651429
1D -127.57233572759607 51.47541278997306
1E -127.52425100931688 51.489173183493094
```

```
2A -127.524251009317 51.4891731834931
2B -127.572335727596 51.4754127899731
2C -127.563090533222 51.4681981465143
2D -127.524251009317 51.4891731834931
```

The three common points are not the same! They round to the values shown in QGIS but cause random errors when I do ST_Contains on the polygons. Sometimes polygon 1 contains polygon 2 and sometimes it doesn't. The reason for the randomness is that by resnapping just one point and saving I get:

```
1A -127.52425100931701 51.48917318349311
1B -127.556950974351 51.44547770380399
1C -127.5630905332221 51.4681981465143
```

1D -127.57233572759607 51.47541278997306
1E -127.52425100931701 51.48917318349311

2A -127.524251009317 51.4891731834931
2B -127.57233572759606 51.475412789973056
2C -127.56309053322208 51.468198146514275
2D -127.524251009317 51.4891731834931

Totally different values as far as PostGIS is concerned.

Just to be sure it's not PostGIS I tried this:

```
UPDATE shapes SET geog = ST_GeomFromEWKT('SRID=4326;MULTIPOLYGON((( -127.524251009317  
51.4891731834931, -127.556950974351 51.445477703804, -127.563090533222 51.4681981465143, -127.572335727596  
51.4754127899731, -127.524251009317 51.4891731834931))))') WHERE id = 1;  
UPDATE shapes SET geog = ST_GeomFromEWKT('SRID=4326;MULTIPOLYGON((( -127.524251009317  
51.4891731834931, -127.572335727596 51.4754127899731, -127.563090533222 51.4681981465143, -127.524251009317  
51.4891731834931))))') WHERE id = 2;
```

And I always get the following in the database no matter how many times I do the UPDATES:

1A -127.524251009317 51.4891731834931
1B -127.556950974351 51.445477703804
1C -127.563090533222 51.4681981465143
1D -127.572335727596 51.4754127899731
1A -127.524251009317 51.4891731834931

2A -127.524251009317 51.4891731834931
2D -127.572335727596 51.4754127899731
2C -127.563090533222 51.4681981465143
2A -127.524251009317 51.4891731834931

Note: The values in the database I've show were read by a C program that used a custom Postgres function to read the internal PostGIS GSERIALIZED binary data. This made sure nothing along the line mucked with the values (no conversions to or from WKT/WKB). The coordinates were printed using the very accurate and precise David M. Gay's floating-point conversion library.

History

#1 - 2016-12-13 12:36 PM - John Zielinski

Just for reference, here's the range of doubles (float8) that 1A.x has in the tables above:

541afa4e -127.52425100931688
541afa4f -127.5242510093169
541afa50 -127.52425100931691
541afa51 -127.52425100931693
541afa52 -127.52425100931694
541afa53 -127.52425100931696
541afa54 -127.52425100931697

541afa55 -127.52425100931698

541afa56 -127.524251009317

541afa57 -127.52425100931701

#2 - 2016-12-13 07:16 PM - John Zielinski

Found something new. Two snapped points again ended up being not the same but this time visible in the vertex editor (just the x value):

-90.237137558

-90.2371375579999

I suspect it's because the shortest form was less than the maximum digits displayed.

#3 - 2017-05-01 01:01 AM - Giovanni Manghi

- *Easy fix? set to No*

- *Regression? set to No*

#4 - 2017-09-08 01:35 PM - Jürgen Fischer

- *Assignee deleted (Jürgen Fischer)*

#5 - 2019-03-09 03:09 PM - Giovanni Manghi

- *Status changed from Open to Closed*

- *Resolution set to end of life*

End of life notice: QGIS 2.18 LTR

Source:

<http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/>

QGIS 3.4 has recently become our new Long Term Release (LTR) version. This is a major step in our history – a long term release version based on the massive updates, library upgrades and improvements that we carried out in the course of the 2.x to 3x upgrade cycle.

We strongly encourage all users who are currently using QGIS 2.18 LTR as their preferred QGIS release to migrate to QGIS 3.4. This new LTR version will receive regular bugfixes for at least one year. It also includes hundreds of new functions, usability improvements, bugfixes, and other goodies. See the relevant changelogs for a good sampling of all the new features that have gone into version 3.4

Most plugins have been either migrated or incorporated into the core QGIS code base.

We strongly discourage the continued use of QGIS 2.18 LTR as it is now officially unsupported, which means we'll not provide any bug fix releases for it.

You should also note that we intend to close all bug tickets referring to the now obsolete LTR version. Original reporters will receive a notification of the ticket closure and are encouraged to check whether the issue persists in the new LTR, **in which case they should reopen the ticket.**

If you would like to better understand the QGIS release roadmap, check out our roadmap page! It outlines the schedule for upcoming releases and will help you plan your deployment of QGIS into an operational environment.

The development of QGIS 3.4 LTR has been made possible by the work of hundreds of volunteers, by the investments of companies, professionals, and administrations, and by continuous donations and financial support from many of you. We sincerely thank you all and encourage you to collaborate and support the project even more, for the long term improvement and sustainability of the QGIS project.

