

QGIS Application - Bug report #14961

Snapping bug

2016-06-06 01:21 AM - Pablo Fernández Moniz

Status:	Closed	
Priority:	High	
Assignee:		
Category:	Digitising	
Affected QGIS version:	2.14.3	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: worksforme
Crashes QGIS or corrupts data:	No	Copied to github as #: 22910
Description		
<p>Hello,</p> <p>We have found an issue with the snapping functionality in QGIS 2.14.3.</p> <p>It seems that after we insert a feature and save the layer, we are no longer able to snap to the saved feature.</p> <p>However if we reload the layers and reconfigure the snapping options, we are able to snap to the previously stored features but not the ones saved in the current session.</p> <p>All data is stored in a PostgreSQL9.3.12 /PostGIS 2.1.8 database</p> <p>We attach a short video showing this behaviour with the name "QGIS_snapping_bug".</p> <p>Kind regards,</p>		

History

#1 - 2016-06-06 03:54 AM - Harrissou Santanna

no video attached.
regards

#2 - 2016-06-06 08:00 AM - Pablo Fernández Moniz

- File snapping.m4v added

attaching video

#3 - 2016-06-22 05:53 AM - Seb Radzimski

Hi,.

I have similar problem in Windows QGIS 2.8.9 and 2.14.3 - when I start QGIS and I open .qgs project, snapping is working fine (according to 'snapping options' saved in .qgs project). But if I reload the same project (without restarting QGIS) snapping is not working. I have to set it again in 'snapping options' window or shut QGIS, start it again and load projekt - then snapping is working.

I don't see this problem on Ubuntu 12.04 (QGIS 2.8.2) but it is in Windows in QGIS 2.8.9 LTR and 2.14.3.

Regards.

#4 - 2016-07-05 04:47 AM - Martin Dobias

Hmm I am unable to reproduce... here is what I tried:

- 1) start QGIS, create a new temporary scratch layer (with line geometry)
- 2) open snapping options, use advanced mode, using snapping to vertex, 20 pixels tolerance
- 3) create a new line feature in the layer
- 4) snapping works as expected
- 5) save the new line feature
- 6) snapping still works as expected

Could you please verify whether this works for you too? Maybe there is something else involved, something to do with project configuration or data storage...

#5 - 2016-07-05 04:51 AM - Martin Dobias

- Status changed from Open to Feedback

#6 - 2016-07-19 06:00 AM - Pablo Fernández Moniz

Hi again,

We can confirm that the steps above work as expected for scratch layers. Regarding the QGIS project, all our tests have been done starting from an empty project (blank paper icon).

In our case the problem seems to be related with the interaction between QGIS and the PostgreSQL database regarding primary keys. In [1] we share a minimal setup to replicate the exact bug.

The steps to replicate the error are:

- 1) Execute the setup shared in [1] on PostgreSQL (for example using PgAdmin3).
- 2) Execute "Layer" > "Add Layer" > "Add PostGIS Layers"
- 3) Load both "test.snapping_working" and "test.snapping_broken" tables.
- 4) Open snapping options, use advanced mode, using snapping to vertex, 20 pixels tolerance for both layers
- 5) Create a new 'snapping_working' feature and DO NOT save layer yet
- 6) Snapping works as expected
- 7) Save the changes on the layer 'snapping_working'
- 8) Snapping works as expected
- 9) Create a new 'snapping_broken' feature and DO NOT save layer yet
- 10) Snapping works as expected
- 11) Save the changes on the layer 'snapping_broken'
- 12) Snapping does not work

We managed to see that the difference between both cases is the following:

- In step 5, before clicking ok, the id field has the value "nextval('test.snapping_working_id_seq'::regclass)". That means that QGIS already passes an expression to the PostgreSQL databases.
- In step 9, before clicking ok, the id field is empty (i.e., its value is NULL). That means that the condition inside 'test.generate_id()' trigger function is valid and the id is set to 99.

From the previous test case, we concluded that the snapping error happens somehow when the value of the primary key is set up inside a trigger.

[1] Code to replicate the issue

```
-- Create schema to hold the testing tables
DROP SCHEMA IF EXISTS test CASCADE;
```

```
CREATE SCHEMA test;
```

```
/*
```

```
This table has a line geometry and its primary key is defined  
using a sequence of bigints. Therefore, bigserial will always  
automatically increment the id value.
```

```
*/
```

```
CREATE TABLE test.snapping_working
```

```
(
```

```
id bigserial NOT NULL,  
geom geometry(LineString,32628),  
CONSTRAINT snapping_working_pkey PRIMARY KEY (id)
```

```
);
```

```
/*
```

```
This table has a line geometry and its primary key is defined  
using a bigint type. This time, the id value is generated  
by a custom trigger.
```

```
*/
```

```
CREATE TABLE test.snapping_broken
```

```
(
```

```
id bigint NOT NULL,  
geom geometry(LineString,32628),  
CONSTRAINT snapping_broken_pkey PRIMARY KEY (id)
```

```
);
```

```
-- ID generation trigger function
```

```
CREATE OR REPLACE FUNCTION test.generate_id() RETURNS trigger AS
```

```
$$
```

```
DECLARE
```

```
code integer := 99;
```

```
check_code bigint;
```

```
BEGIN
```

```
IF NEW.id IS NULL THEN
```

```
NEW.id := code;
```

```
END IF;
```

```
RETURN NEW;
```

```
END;
```

```
$$
```

```
LANGUAGE plpgsql SECURITY DEFINER;
```

```
-- Define trigger
```

```
CREATE TRIGGER aa_set_id
```

```
BEFORE INSERT
```

```
ON test.snapping_broken
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE test.generate_id();
```

#7 - 2017-01-02 07:54 AM - Giovanni Manghi

- Status changed from Feedback to Open
- Operating System deleted (all)
- Crashes QGIS or corrupts data changed from No to Yes

#8 - 2017-03-07 10:46 AM - Giovanni Manghi

- Resolution set to workforme
- Status changed from Open to Closed

I followed the provided steps and queries (thanks) and unless I really misunderstood the problem this seems really ok to me on 2.18.4

Files

snapping.m4v	443 KB	2016-06-06	Pablo Fernández Moniz
--------------	--------	------------	-----------------------