

QGIS Application - Bug report #13952

QGIS node tool causes snapping another feature randomly

2015-12-07 02:50 AM - Regis Haubourg

<b>Status:</b>	Closed	
<b>Priority:</b>	Low	
<b>Assignee:</b>	Sandro Santilli	
<b>Category:</b>	Digitising	
<b>Affected QGIS version:</b>	2.14.3	<b>Regression?:</b> No
<b>Operating System:</b>		<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b>
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 21967
<b>Description</b>		
<p>Hi,</p> <p>very blocking issue here with 2.12.1 (BTW could someone add it to target versions in redmine?).</p> <p>with snapping options and topological editing enabled, moving some endpoints of polylines snaps the target line, but also will snapp vertex from ANOTHER unrelated feature to it.</p> <p>Hard to explain, see video <a href="#">here</a></p> <p>I have a complex editing project, created with 2.10 at start.</p> <p>This bug prevents me from doing the edits I need... blocker then. It deserves backports if reproduced and solved.</p>		

Associated revisions

Revision 6076451a - 2016-06-29 07:18 PM - Sandro Santilli

Only insert segment snap points in the layer they belong

Fixes #13952

Revision 647ab4e9 - 2016-06-30 11:55 AM - Sandro Santilli

Only insert segment snap points in the layer they belong

Fixes #13952

History

#1 - 2015-12-07 03:10 AM - Regis Haubourg

Tried to test on master, and add a crash when changing default projection (was otf 3857 and layer in 2154). So ... I could not test

Tested on 2.8.4, no issue there.

#2 - 2015-12-22 02:38 PM - Jakub Kosik

It is same issue I've noticed in earlier versions (look at issue #10957 dig3.png).

More, it happens only if I have more than one legend entry to same dataset.

**#3 - 2015-12-23 02:00 AM - Giovanni Manghi**

- Target version changed from Future Release - High Priority to Version 2.14
- Affected QGIS version changed from 2.12.0 to 2.12.1

**#4 - 2016-01-23 11:58 AM - Martin Dobias**

- Status changed from Open to Feedback

Hi Regis, would it be possible to share the project, cut down to bare minimum of layers (maybe 1-2) when it is still possible to reproduce the bug? Feel free to send directly if the data cannot be shared publicly.

**#5 - 2016-03-07 11:57 PM - Daan Goedkoop**

Recently I have come across something similar. It might be the same bug. While digitising, with snapping but without topological editing, the pink snapping "+" would suddenly appear outside of any feature and not at any existing vertex. When clicking it, the created vertex would (after finishing digitising) end up snapped to some seemingly random vertex way outside view. Interestingly, the node tool did not seem to be affected in this case.

Unfortunately I forgot to save that particular version of the shapefile and now I cannot reproduce the problem anymore.

**#6 - 2016-03-21 01:27 AM - Daan Goedkoop**

Just now I noticed this issue again. It is not reproducible for me. For what it is worth: the problem remained after saving any edits to the shapefile. However, when I save the project, close QGIS and open the project again, the problem disappeared. Does that mean that this problem is a different one, than the one described in this bug report?

**#7 - 2016-05-23 08:19 AM - Giovanni Manghi**

This issue seems serious, but I haven't found a way to replicate. Could the reporter attach a sample project with data? Without it is difficult to at least replicate. Thanks.

**#8 - 2016-05-24 03:49 AM - Sylvain Beorchia**

Hi. I've got the same issue, and i'm able to reproduce it every time. I will provide a Qgis project with a small postgres DB.

**#9 - 2016-05-24 03:55 AM - Paolo Cavallini**

Which snapping settings are you using? Which CRS?

**#10 - 2016-05-24 04:07 AM - Sylvain Beorchia**

The problem occurs only when "Activate topologic edition" is enable in snapping options. CRS is 2154.

**#11 - 2016-05-24 04:39 AM - Sylvain Beorchia**

- File bug\_qgis\_13952.zip added

Here is a ZIP containing 2 shapefiles and a Qgis project.

Open the qgis project, and search for the branch id=1212. Edit the layer and try to connect the top vertex of the branch(1212) to the closest arc. You have to enable snapping, and "Enable topological editing".

**#12 - 2016-05-24 04:43 AM - Sylvain Beorchia**

- File Sans\_titre.png added

See the image joined to see the bug.

**#13 - 2016-05-24 05:30 AM - Regis Haubourg**

Sylvain Beorchia wrote:

*Here is a ZIP containing 2 shapefiles and a Qgis project.*

*Open the qgis project, and search for the branch id=1212. Edit the layer and try to connect the top vertex of the branch(1212) to the closest arc. You have to enable snapping, and "Enable topological editing".*

Good catch Sylvain! I can reproduce it here at first try.

I never managed to reproduced the bug outside of my complex postgis project and share a sample project.

This is pretty serious indeed.

**#14 - 2016-05-25 09:31 AM - Saber Razmjooei**

- Status changed from Feedback to Open

**#15 - 2016-05-26 01:17 AM - Saber Razmjooei**

- Status changed from Open to Feedback

- Affected QGIS version changed from 2.12.1 to 2.14.3

If you change the snapping setting to Vertex only, it works fine.

There was another bug related to the precision of snapping tolerance for the node tool, which I think could be related to this issue. Will try to find and link it here.

**#16 - 2016-06-21 06:02 AM - Giovanni Manghi**

- Status changed from Feedback to Open

the error as described in #13952-11

is still valid on the latest master.

**#17 - 2016-06-21 06:03 AM - Giovanni Manghi**

see also [https://issues.qgis.org/attachments/9959/Sans\\_titre.png](https://issues.qgis.org/attachments/9959/Sans_titre.png)

#18 - 2016-06-22 06:52 AM - Sandro Santilli  
- Assignee set to Sandro Santilli

#19 - 2016-06-22 07:05 AM - Sandro Santilli

Confirmed as of 2.14.3 -- Snap to Segment Only is enough to reproduce (#note-11)

#20 - 2016-06-22 07:07 AM - Sandro Santilli

2.8.9 is not affected, 2.12.1 supposedly was (as for original submission)

#21 - 2016-06-22 07:24 AM - Sandro Santilli

2.10.1 is NOT affected either, so new range is good:2.10.1 bad:2.12.1

#22 - 2016-06-22 07:40 AM - Sandro Santilli  
- Status changed from Open to In Progress

I've tested 2.12.0 to be also bad, so bisect range is 2.10.1..2.12.0 -- bisecting started

#23 - 2016-06-22 09:16 AM - Sandro Santilli

```
9c2d70186f054b71f1b792d13133f3856c855bf3 is the first bad commit
commit 9c2d70186f054b71f1b792d13133f3856c855bf3
Author: Marco Hugentobler <marco.hugentobler@sourcepole.ch>
Date: Wed Sep 16 05:19:26 2015 +0200

Node tool without click-click mode

:040000 040000 eb8ad3fa7fc599731bbcb89067d32a44eef25dbd 1f3b3ce634bb47ed00837ca35a7278de83496099 M src
```

#24 - 2016-06-22 09:19 AM - Sandro Santilli

Marco, could you take a look at this ? There's been lots of changes but no testcase:

```
commit 9c2d70186f054b71f1b792d13133f3856c855bf3
Author: Marco Hugentobler <marco.hugentobler@sourcepole.ch>
Date: Wed Sep 16 05:19:26 2015 +0200

Node tool without click-click mode

src/app/nodetool/qgsmaptoolnodetool.cpp | 500
+++++-----
src/app/nodetool/qgsmaptoolnodetool.h | 49 +++++----
src/app/nodetool/qgsnodeeditor.cpp | 23 ++---
src/app/nodetool/qgsnodeeditor.h | 1 -
```

```
src/app/nodetool/qgsselectedfeature.cpp | 53 ++-----  
src/app/nodetool/qgsselectedfeature.h | 3 -  
6 files changed, 391 insertions(+), 238 deletions(-)
```

#### #25 - 2016-06-27 09:58 PM - Sandro Santilli

- % Done changed from 0 to 40

A QGIS version with the offending commit reverted is available on <https://github.com/qgis/QGIS/pull/3248>.

I confirm that reverting the commit fixes the random snap.

#### #26 - 2016-06-28 03:04 AM - Sandro Santilli

- % Done changed from 40 to 50

Adding debugging prints I came across this computation in QgsLineStringV2::closestSegment that does look wrong:

```
src/core/geometry/qgslinestringv2.cpp: 775: (closestSegment) [1ms] XXX ST_Distance('POINT(771938 6.95593e+06)::geometry,  
'LINESTRING(771946 6.95593e+06,771904 6.95595e+06)::geometry'); -- 0.0220404
```

That's coming from this snippet:

```
testDist = QgsGeometryUtils::sqrDistToLine( pt.x(), pt.y(), prevX, prevY, currentX, currentY, segmentPtX, segmentPtY, epsilon );  
if ( testDist < sqrDist )  
{  
    QgsDebugMsg( QString("XXX ST_Distance('POINT(%1 %2)::geometry, 'LINESTRING(%3 %4,%5 %6)::geometry'); -- %7 ")  
        .arg(pt.x()) .arg(pt.y())  
        .arg(prevX) .arg(prevY)  
        .arg(currentX) .arg(currentY)  
        .arg(testDist)  
    );  
}
```

The debug suggests that QgsGeometryUtils::sqrDistToLine() is broken, because PostGIS gives, for the ST\_Distance call, a value of

3.43946864321886

, which squared becomes 11.8299445476857856634618596996

.

Qgis instead gets 0.0220404

which is way below sqrSnappingTolerance:6.40146, explaining the problem.

Next step would be adding a testcase for QgsGeometryUtils::sqrDistToLine().

#### #27 - 2016-06-28 03:26 AM - Sandro Santilli

A focused testcase, pushed with commit:f0e0ba5bb0f844e9ee28eb9572340f31ae13c6ee, does not give the same bogus result. I'm still debugging how could it be possible to obtain the bogus result (some state implied?)

#28 - 2016-06-28 03:58 AM - Sandro Santilli

Printing more digits gives a matching result between PostGIS and QGIS:

ST\_Distance('POINT'::geometry, 'LINESTRING'::geometry); -- 0.143113367679915

( it's actually 3.378 which squared gives 0.1431 )

But I found that the LINESTRING part is the FID of the geometry being moved, not FID=11, so this makes me think that the distanc used for FID=11 (the one which should not move) is just a leftover value for the variable computed for FID=27) -- or something along those lines.

#29 - 2016-06-28 09:18 AM - Sandro Santilli

New finding: the distance to the fid is correct but the *layer* is different. We have 2 layers, both contain a fid=11. The one bogusly snapped is from layer "arc", and is too distance. The fid=11 from the layer being moved (branch) is instead close (as we actually move the vertex right on top of it). The code gets confused as for which vector layer to snap geometries from.

Setting snapping options to "currentLayer" rather than "All Visible Layer" fixes the issue.  
Adding another layer with the same geometry as fid=11 from "branch" adds another arbitrary FID to the list of those snapped in any visible layer.

Basically, if I'm seeing this right, all geometries with a given FID value are snapped from any visible layer, rather than FIDs being isolated for each layer.

#30 - 2016-06-28 10:07 AM - Sandro Santilli

- % Done changed from 50 to 80

Pull request ready with a fix: <https://github.com/qgis/QGIS/pull/3251>  
Can be fetched from <https://github.com/strk/QGIS/tree/snap-to-proper-layer>

With the fix, behavior is back to the 2.8.9 one, which seems sane enough (maybe not 100% expected, but safe).  
The only thing missing would now be an automated testcase to save the many hours spent on this ...

#31 - 2016-06-30 02:56 AM - Sandro Santilli

- Status changed from In Progress to Closed

Fixed in changeset commit:"647ab4e947edec24b8babe51e188d95270275ea4".

#32 - 2017-05-19 09:13 AM - Jürgen Fischer

- Description updated  
- Priority changed from Severe/Regression to Low

Files			
bug_qgis_13952.zip	12.4 KB	2016-05-24	Sylvain Beorchia
Sans_titre.png	167 KB	2016-05-24	Sylvain Beorchia