

QGIS Application - Bug report #16927

Can't commit changes to postgis 2D linestring layer when snapping to 3D point layer

2017-07-25 12:14 PM - Jochen Huber

<b>Status:</b>	Closed	
<b>Priority:</b>	High	
<b>Assignee:</b>	Luigi Pirelli	
<b>Category:</b>	Data Provider/PostGIS	
<b>Affected QGIS version:</b>	2.18.11	<b>Regression?:</b> Yes
<b>Operating System:</b>	Windows 7	<b>Easy fix?:</b> No
<b>Pull Request or Patch applied:</b>	Yes	<b>Resolution:</b>
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 24826
<b>Description</b>		
Situation: <ul style="list-style-type: none"><li>- 2D linestring layer from postgis db in editing mode</li><li>- New line is drawn, the points are selected by snapping to a 3D point layer in the same db</li><li>- Committing the changes fails with the error "geometry type is not compatible with the current layer."</li><li>-&gt; When drawing a line with arbitrary points (snapping may be active, but the new points are not created by snapping to a point layer)no error occurs;</li><li>-&gt; When snapping to points of a 2D point layer, no error occurs</li></ul> QGIS seems to try to add 3D coordinates to the new 2D line feature instead of using only the first two dimensions.		

Associated revisions

Revision d3d8f4dd - 2017-09-27 04:45 PM - Larry Shaffer

Merge pull request #5223 from boundlessgeo/geom\_compatibility\_check\_release-2\_18-fix15741

On behalf of Giovanni: Geom compatibility check release 2.18 fixes #15741 #16927

History

#1 - 2017-07-25 12:16 PM - Giovanni Manghi

- Status changed from Open to Feedback

Did it happened in an older qgis release?

#2 - 2017-07-25 05:45 PM - Jochen Huber

- Status changed from Feedback to Open

With 2.18.4 on Windows, the issue is basically the same, but the error message is marked as a PostGIS data provider error and is more useful: "ERROR: Geometry has Z dimension but column does not".

With 2.10.1 on Windows, the issue does NOT occur - so it seems to be a regression.

These are the only older version I currently have available, but I could install a specific version and check again if this would be useful.

#3 - 2017-07-25 05:46 PM - Giovanni Manghi

Jochen Huber wrote:

|

*With 2.18.4 on Windows, the issue is basically the same, but the error message is marked as a PostGIS data provider error and is more useful:  
"ERROR: Geometry has Z dimension but column does not".  
With 2.10.1 on Windows, the issue does NOT occur - so it seems to be a regression.  
These are the only older version I currently have available, but I could install a specific version and check again if this would be useful.*

it would be nice if you could test on the actual LTR release, 2.14.17. Thanks!

**#4 - 2017-07-25 07:22 PM - Giovanni Manghi**

- Priority changed from Normal to High  
- Regression? changed from No to Yes

**#5 - 2017-07-26 09:52 AM - Jochen Huber**

Tested on LTR release (2.14.17): Issue confirmed, error message: "Geometry has Z dimension but column does not"  
Tested on Master (2.99.0, code version cb088a2): Issue confirmed, error message: "geometry type is not compatible with the current layer."

**#6 - 2017-08-11 04:19 PM - Jürgen Fischer**

- Description updated

**#7 - 2017-08-29 09:21 AM - Tudor Băráscu**

Just hit this issue today. Tested it under 2.14 and it gives this error:

PostGIS error while adding features: ERROR: Geometry has Z dimension but column does not

It's also present under 2.18 and master but with a different error message:  
Just hit this issue today. Tested it under 2.14 and it gives this error:

ERROR: 1 feature(s) not added - geometry type is not compatible with the current layer.

I would like to add that the issue only appears if you start from a 3d point (it isn't present if you clicked somewhere without snapping and then end your snap to a 3d point).

**#8 - 2017-09-05 04:08 PM - Luigi Pirelli**

so what is the error here? IMHO seems correct that a 3D feature shouldn't be committed on 2D layer. Is it the problem related only to the better message?  
In case of 2.18, the message is more generic because QGIS prevent to send features to provider if it is not compatible. So the more detailed provider message is never risen.  
It has to be considered a regression the change in message detail?  
BTW We can add the geometry type of source and target geometry if this can give more useful feedback. Any opinion?

**#9 - 2017-09-05 04:10 PM - Giovanni Manghi**

Luigi Pirelli wrote:

|

| so what is the error here? IMHO seems correct that a 3D feature shouldn't be committed on 2D layer.

they are two different layers, from the description:

*2D **linestring** layer from postgis db in editing mode*

*New line is drawn, the points are selected by snapping to a 3D **point** layer in the same db*

#### #10 - 2017-09-05 04:39 PM - Luigi Pirelli

Sincerely I almost scared if the commitChanges function have to decide what would be the user intention. IMHO the fix shouldn't be inside commitChanges (that can give more detailed message). e.g. a compatible geometry would arrive during commit, if not, commitChanges would still fail (as in this case).

The fix should be in some other places. e.g:

A) during snap (e.g. adding an option to strictly follow in source geom type => no 3d type conversion)

B) outside snap, but inside addFeature (I didn't investigate the code)... in this way the error is raised as soon a feature is added (but not yet committed)

any opinion? I would prefer B to leave Snapping more general and hide checks at "business function" level

Giovanni Manghi wrote:

| Luigi Pirelli wrote:

| so what is the error here? IMHO seems correct that a 3D feature shouldn't be committed on 2D layer.

| they are two different layers, from the description:

| *2D **linestring** layer from postgis db in editing mode*

| *New line is drawn, the points are selected by snapping to a 3D **point** layer in the same db*

#### #11 - 2017-09-05 04:51 PM - Jochen Huber

As described, in this case no error should be raised, since snapping delivers 3D coordinates for a 2D feature, so the third dimension should simply be ignored.

The "regression" mark is not there because the kind of error has changed, but because it used to work without error (at least until version 2.10.1). Probably addFeature should discard the Z coordinate (might apply to M values as well) if it is not needed.

#### #12 - 2017-09-07 07:35 PM - Luigi Pirelli

IMHO a solution should leave the snapping to modify 2D to 3D. The reason is to allow user to rise up a flat vector if this is the user intention.

some considerations:

- 1) IMHO if there is a single-2-multy type conversion have to be notified
- 2) We can add a snapping option like "mantain type cardinality" (True by default) that avoid to change geom dimension
- 3) add a generic check in the editing buffer that check geom compatibility as soon as a geom is changed.

3) can be solved as stated here: [https://qgis.org/api/2.18/qgsvectorlayereditbuffer\\_8cpp\\_source.html#l00197](https://qgis.org/api/2.18/qgsvectorlayereditbuffer_8cpp_source.html#l00197)

"// TODO: check compatible geometry"

in this place can be added the same controls set in commitChanges() of the same class... or moved these controls in a reusable methods.

Would be greate to have this control outside EditBuffed or as static method and with SIP biunding to allow other application to do this check for their business functions.-

**#13 - 2017-09-08 01:35 PM - Jürgen Fischer**

- Assignee deleted (*Jürgen Fischer*)

**#14 - 2017-09-08 01:44 PM - Luigi Pirelli**

- Assignee set to *Luigi Pirelli*

**#15 - 2017-09-27 04:44 PM - Anonymous**

- % Done changed from 0 to 100

- Status changed from *Open* to *Closed*

Applied in changeset commit:qgis|d3d8f4dd437f132cd6f73831d845f6b7286e2d30.

**#16 - 2017-09-27 05:14 PM - Luigi Pirelli**

- Pull Request or Patch supplied changed from *No* to *Yes*