

## QGIS Application - Bug report #9202

### Can't write from python in integer fields of sqlite (not spatial) tables

2013-12-11 09:38 AM - Francisco Puga

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Category:</b>	Python plugins	
<b>Affected QGIS version:</b>	2.0.1	<b>Regression?:</b> No
<b>Operating System:</b>		<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>		<b>Resolution:</b>
<b>Crashes QGIS or corrupts data:</b>		<b>Copied to github as #:</b> 17830
<b>Description</b>		
<p>I'm using the python console to automatice a process that includes copy values from a dbf layer to a sqlite (non spatial) table. All works nice except when i need to write a value in a integer field of the sqlite table. The value of the integer fiels is always NULL in the target table, the rest of the values are correctly written.</p> <p>Integer fields in the properties of the table are represented as <b>qlonglong</b> and i think that the python encapsulation of these fields is failing, because when i try to write to a integer field of a sqlite (spatial) layer it works correctly, and in this case the type of the field appears as <b>int</b> in the properties</p> <p>I attach a zip with a ssce, that contains a sqlite file created with this commands:</p> <pre>spatialite -bail test_qlonglong.sqlite "SELECT InitSpatialMetaData();" spatialite -bail test_qlonglong.sqlite "CREATE TABLE spatial (gid INTEGER PRIMARY KEY, intfield INTEGER); SELECT addgeometrycolumn('spatial', 'geom', 4326, 'POINT', 2);" spatialite -bail test_qlonglong.sqlite "CREATE TABLE non_spatial (gid INTEGER PRIMARY KEY, intfield INTEGER);"</pre> <p>and a script.py file that shows how in spatial layers integer values are correctly written and in non spatial not, with this content:</p> <pre>spatial = [x for x in iface.legendInterface().layers() if x.name() == 'spatial'][0] spatial.dataProvider().clearErrors() spatialFields = spatial.dataProvider().fields() feat = QgsFeature() feat.setFields(spatialFields) feat.setAttribute('intfield', 10) (res, foo) = spatial.dataProvider().addFeatures([feat]) if not res: print "Error writing spatial layer"</pre> <pre>non_spatial = [x for x in iface.legendInterface().layers() if x.name() == 'non_spatial'][0] non_spatial.dataProvider().clearErrors() non_spatialFields = non_spatial.dataProvider().fields() feat = QgsFeature() feat.setFields(non_spatialFields) feat.setAttribute('intfield', 10) (res, foo) = non_spatial.dataProvider().addFeatures([feat]) if not res: print "Error writing non spatial layer"</pre> <p>I'm using Ubuntu 13.04 32bits, with qgis 2.01 installed from the ppa ubuntugis-unstable</p>		

---

## History

---

**#1 - 2013-12-12 01:20 AM - Matthias Kuhn**

- *Status changed from Open to Feedback*

Just tested here (64bit Fedora)

- spatial layer: has no intfield attribute and therefore the code fails for this
- non spatial layer: 2.0.1 problem reproducible
- non spatial layer: master problem solved

So it seems this has been fixed already.

**#2 - 2013-12-12 07:11 AM - Francisco Puga**

- *File screenshot\_issue9202.png added*

I build qgis from source, commit 221263db95fc on Dec 8, and i'm still having this issue. Maybe a 32bit problem?

I attach a screenshot that show how the value is correctly written in the spatial table and NULL is written to non\_spatial table.

**#3 - 2013-12-12 08:16 AM - Francisco Puga**

Ouch. Shame on me. It has been fixed already. I was launching "installed" qgis instead of the builded qgis.

Should i close the bug myself?

**#4 - 2013-12-12 08:26 AM - Francisco Puga**

- *Status changed from Feedback to Closed*

## Files

test_qlonglong.tgz	312 KB	2013-12-11	Francisco Puga
screenshot_issue9202.png	225 KB	2013-12-12	Francisco Puga