

QGIS Application - Bug report #7223

QgsVectorFileWriter fails in python

2013-02-22 06:58 PM - Angus Carr

Status:	Closed	
Priority:	High	
Assignee:	Martin Dobias	
Category:	Python plugins	
Affected QGIS version:	master	Regression?: No
Operating System:	All	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution:
Crashes QGIS or corrupts data:	No	Copied to github as #: 16255
Description		
<p>Using the code from the QGIS python cookbook, QgsVectorFileWriter fails, as shown below. Has the API changed? This code works in QGIS 1.8, and worked two weeks ago in the master nightly compile (Probably 20130208 or so, just at a guess).</p> <p>It no longer works.</p> <p>From the Console:</p> <pre>from PyQt4.QtCore import * from PyQt4.QtGui import * fields = { 0 : QgsField("first", QVariant.Int), 1 : QgsField("second", QVariant.String) } writer = QgsVectorFileWriter("/tmp/my_shapes.shp", "CP1250", fields, QGis.WKBPoint, None, "ESRI Shapefile")</pre> <p>Traceback (most recent call last):</p> <p>File "<input>", line 1, in <module></p> <p>TypeError: arguments did not match any overloaded call:</p> <p>QgsVectorFileWriter(QString, QString, QgsFields, QGis.WkbType, QgsCoordinateReferenceSystem, QString driverName="ESRI Shapefile", QStringList datasourceOptions=QStringList(), QStringList layerOptions=QStringList(), QString newFilename=None): argument 3 has unexpected type 'dict'</p> <p>QgsVectorFileWriter(QgsVectorFileWriter): argument 1 has unexpected type 'str'</p>		

History

#1 - 2013-02-22 07:17 PM - Nathan Woodrow

Yeah the API has changed in master. Try this:

```
fields = QgsFields()
fields.append(QgsField("first", QVariant.Int))
...
writer = QgsVectorFileWriter("/tmp/my_shapes.shp", "CP1250", fields, QGis.WKBPoint, None, "ESRI Shapefile")
```

#2 - 2013-02-27 08:59 PM - Angus Carr

Well, I tried what you suggested, and the code now runs, but I get null in all my attributes...

2025-04-27

1/3

```
caps = layer.dataProvider().capabilities()
if caps & QgsVectorDataProvider.AddFeatures:
    fet = QgsFeature()
    fet.setGeometry(QgsGeometry.fromWkt(self.plot_def['plot geometry wkt']))
    fet.setAttribute("Plot_Num", QVariant(self.plot_def['Plot number']))
    fet.setAttribute("Area_ID", QVariant(self.plot_def['Area ID']))
    fet.setAttribute("Size", QVariant(self.plot_def['plot size']))
    (res, outFeats) = layer.dataProvider().addFeatures( [ fet ] )
```

What am I missing now?

#3 - 2013-02-27 11:59 PM - Alexander Bruy

Confirmed. `setAttribute()` by index or by field name don't work. Also I can't assign attribute values using

```
feature["fieldName"] = QVariant(some_value)
```

Only [setAttribute\(\)](#) method works for me.

#4 - 2013-03-03 09:49 PM - Angus Carr

So, the final working example is:

```
fet.setAttributes([QVariant(self.plot_def['Plot number']), \
    QVariant(self.plot_def['Area ID']), \
    QVariant(self.plot_def['plot size'])])
```

For the Record (or for people trying to make this go).

#5 - 2013-03-03 11:38 PM - Giovanni Manghi

Angus Carr wrote:

So, the final working example is:

[...]

For the Record (or for people trying to make this go).

It should be nice make out if it a note for the developers cookbook, here in the tracker I guess will be hard to find.

#6 - 2013-03-03 11:48 PM - Alexander Bruy

Giovanni Manghi wrote:

2025-04-27

It should be nice make out if it a note for the developers cookbook, here in the tracker I guess will be hard to find.

This is already in TODO.

But I'm wondering why using `setAttribute()` and `get["attrName"] = value` doesn't works as expected

#7 - 2013-03-13 05:00 AM - Alexander Bruy

- Assignee set to Martin Dobias
- Operating System changed from Linux to All
- Priority changed from Normal to High

I increase priority to High because this affects many plugins and confusing for plugin developers

#8 - 2013-03-13 01:22 PM - Martin Dobias

The `QgsFeature` instance now must be first initialized with `initAttributes()` to know how many attributes it will contain. Then there's `setFields()` call that allows doing name-to-index mapping in `QgsFeature`. I would like to simplify that before 2.0 is out, so that user could do everything at once when creating the feature: e.g. `"f = QgsFeature(fields)"` would both initialize the array of attributes and the fields pointer.

Before when attributes were stored in a map, it was not necessary to initialize them in any way - but that came with a cost of being less efficient.

#9 - 2013-03-13 11:16 PM - Alexander Bruy

Martin Dobias wrote:

The `QgsFeature` instance now must be first initialized with `initAttributes()` to know how many attributes it will contain. Then there's `setFields()` call that allows doing name-to-index mapping in `QgsFeature`. I would like to simplify that before 2.0 is out, so that user could do everything at once when creating the feature: e.g. `"f = QgsFeature(fields)"` would both initialize the array of attributes and the fields pointer.

Thanks for clarification, Martin! Now I think that ticket can be closed, because me and original author didn't initialize features properly.

#10 - 2013-03-14 07:30 AM - Angus Carr

- % Done changed from 0 to 100
- Status changed from Open to Closed

I agree.

I had a look at #7233, which corrects a similar thing, and the solution to that shows the correct syntax as far as I can tell.

I would also like to see the `"f = QgsFeature(fields)"` syntax. It contains the necessary information to do everything at once. Perhaps it should even have another alternate constructor:

`"f = QgsFeature(fields.geometry)"` has everything that one would need in a bulk data writing exercise.

But that's for another day...