

QGIS Application - Feature request #18444

Memory layer does not cast attribute types

2018-03-14 10:27 PM - Norwin Roosen

Status:	Feedback	Resolution: Copied to github as #: 26332
Priority:	Normal	
Assignee:		
Category:	Data Provider	
Pull Request or Patch supplied:		
Easy fix?:	Yes	
Description		
<p>Expected Behaviour:</p> <p>Writing values of incorrect type to a memory layer field should cast them to the correct value or throw an error.</p> <p>Actual behaviour:</p> <p>In QGIS 2.18.16, writing a string into a double field of a memory layer writes the value unchanged.</p> <p>Reading the attribute later returns the original value of type unicode.</p> <p>The following python code illustrates this:</p> <pre>from qgis.core import QgsVectorLayer, QgsFeature layer = QgsVectorLayer("Point?field=floatfield:double", "poc", "memory") feat = QgsFeature() feat.setAttributes(['10']) layer.dataProvider().addFeatures([feat]) QgsMapLayerRegistry.instance().addMapLayer(layer) # in my usecase I actually loaded the layer from within another processing algo for f in layer.getFeatures(): print f['floatfield'], type(f['floatfield']) # prints # 10 <type 'unicode'> # should obviously be # 10.0 <type 'float'></pre>		

History

#1 - 2019-01-21 12:35 AM - Jürgen Fischer

- Status changed from Open to Feedback

Please test with QGIS 3.4 - QGIS 2.18 reached it's end of life.

#2 - 2019-02-23 02:03 PM - matteo ghetta

Tested on QGIS 3.4 (with some small changes to the script):

```
from qgis.core import QgsVectorLayer, QgsFeature
```

```
layer = QgsVectorLayer("Point?field=floatfield:double", "poc", "memory")  
feat = QgsFeature()  
feat.setAttributes(['10'])  
layer.dataProvider().addFeatures([feat])  
QgsProject.instance().addMapLayer(layer)
```

```
# in my usecase I actually loaded the layer from within another processing algo  
for f in layer.getFeatures():  
    print(f['floatfield'], type(f['floatfield']))
```

same behavior of your description. IMHO it is not a bug that QGIS casts automatically the data type. Maybe a warning could be thrown, but this could be a feature request?

#3 - 2019-02-23 03:42 PM - Norwin Roosen

Thanks for testing in QGIS 3.4 @matteo!

To me this issue is about semantic correctness of the API & predictability of its results.

Putting a string in a float field is never semantically meaningful, so raising an error may be appropriate.

We shure can argue if following the semantics of the API is a feature or a bugfix, but what matters is that there is working software in the end ;)

#4 - 2019-02-23 05:52 PM - matteo ghetta

- *Tracker changed from Bug report to Feature request*

thanks Norwin Roosen. OK then I'll mark it as feature request. So it will stay open as reminder.