

QGIS Application - Bug report #17332

QGIS unable to save a layer with measures

2017-10-25 05:48 PM - TJ Rockefeller

Status: Closed	
Priority: Normal	
Assignee:	
Category: Unknown	
Affected QGIS version: 2.18.13	Regression?: No
Operating System: Windows 10	Easy fix?: No
Pull Request or Patch supplied: No	Resolution: end of life
Crashes QGIS or corrupts data: No	Copied to github as #: 25229

Description

QGIS seems to be unable to save vector layers with M-values (measures). I have a LineStringZM shapefile, and if I just do a simple save as operation in QGIS and save it as a shapefile, sqlite file, or geojson file (other file types may be affected, but this is what I have tried so far) the resulting file is a LineStringZ geometry type.

I have experimented with the QgsVectorFileWriter, and it appears to have the same issue. I tried the following in the python console.

```
||| writer = QgsVectorFileWriter('c:/test/test_layer.shp', 'UTF8', QgsFields(), QgsWKBTypes.PointZM,  
||| QgsCoordinateReferenceSystem('EPSG:3857'))  
||| feature = QgsFeature()  
||| feature.setGeometry(QgsGeometry(QgsPointV2(QgsWKBTypes.PointZM, 1, 2, 3, 4)))  
||| writer.addFeature(feature)
```

True

```
||| del writer
```

and the resulting test_layer.shp file had a single point and the geometry type is PointZ.

I have a gis stackexchange question here <https://gis.stackexchange.com/q/227794/56050> that doesn't have any valid answers for it yet, which has lead me to writing up this question here.

According to GDAL http://www.gdal.org/drv_shapefile.html the shapefile driver as of version 2.1 should be capable of handling measures, and I am using GDAL/OGR 2.2.1 with QGIS 2.18.13

Thanks,
TJ

History

#1 - 2017-10-30 08:04 PM - TJ Rockefeller

The GIS stackexchange question above now has an answer, and I found that the QgsVectorFileWriter correctly writes out shapefiles with measures if the layerOptions parameter is used and the layer type is explicitly specified there as well as the QgsWKBTypes parameter. Rewriting the example from above it would be

```
>>> writer = QgsVectorFileWriter('c:/test/test_layer.shp', 'UTF8', QgsFields(), QgsWKBTypes.PointZM,  
QgsCoordinateReferenceSystem('EPSG:3857'), layerOptions=['SHPT=POINTZM'])  
>>> feature = QgsFeature()
```

```
>>> feature.setGeometry(QgsGeometry(QgsPointV2(QgsWKBTypes.PointZM, 1, 2, 3, 4)))
>>> writer.addFeature(feature)
True
>>> del writer
```

The problem still exists, though, when doing a save as operation on a layer with measures and saving as a shapefile, sqlite file, geojson file, and potentially others.

#2 - 2017-11-02 11:44 AM - Giovanni Manghi

- Status changed from Open to Feedback

Sorry for the dumb question... what is a layer "with measures"?

#3 - 2017-11-02 02:19 PM - TJ Rockefeller

- Status changed from Feedback to Open

A layer with measures is any geometry type with an "M" in the name like PointM, PointZM, LineStringM, LineStringZM, etc. In a LineString geometry each vertex just has an X and a Y value, in a LineStringZ there is an X, Y, and Z value for every vertex, and in a LineStringZM there is an X, Y, Z, and M(measure) value for each vertex. The M value can be whatever you want. It is just a measure value associated with each vertex in the geometry. I work with road surveying, so typically for geometries that I work with the M values are distance along the curve. This distance is calculated using very accurate distance measuring instruments (DMI), so the distance stored in the M value of each vertex is much more accurate than geometric distance as calculated by the actual LineString geometry, and the distance actually represents distance along the road as driven by a vehicle. It doesn't seem to be a very common thing in GIS circles, but it is becoming a lot more common I believe. I know there are some major updates coming in QGIS 3.0, and hopefully one of the major changes is better support for geometries with measures.

Hopefully this answers your question clearly enough. Let me know if you have any other questions.

#4 - 2017-11-02 02:38 PM - Giovanni Manghi

| *Hopefully this answers your question clearly enough. Let me know if you have any other questions.*

thanks. Have you raised your question in the dev mailing list?

#5 - 2017-11-02 02:45 PM - TJ Rockefeller

I have not. I just recently joined this bug tracker site to report this, so I don't know what you mean by the dev mailing list.

#6 - 2017-11-02 02:49 PM - Giovanni Manghi

TJ Rockefeller wrote:

| *I have not. I just recently joined this bug tracker site to report this, so I don't know what you mean by the dev mailing list.*

developers mailing list. That is the right place were to ask questions. Is possible here that there is a bug, but maybe the code to save a PointM, PointZM,

LineStringM, LineStringZM, etc. is just not the one you are using. Better check with the developers.

#7 - 2019-01-21 12:34 AM - Jürgen Fischer

- Status changed from Open to Feedback

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

#8 - 2019-02-23 02:19 PM - Alessandro Cristofori

- File *M_multiline_test.zip* added

- File *m_values_qgis.png* added

- File *MvaluesArc.JPG* added

I created a Multiline with M values in ArcGIS, saved it as shape file and opened it in QGIS 3.4.4. The dataset is showing as Multiline and when opening the editing dialog no M values are shown (see attached).

The test data set is attached.

#9 - 2019-02-23 10:40 PM - Giovanni Manghi

Alessandro Cristofori wrote:

I created a Multiline with M values in ArcGIS, saved it as shape file and opened it in QGIS 3.4.4. The dataset is showing as Multiline and when opening the editing dialog no M values are shown (see attached).

The test data set is attached.

this seems to be a different issue (or maybe related). The original description is about qgis not being able to save M values.

#10 - 2019-03-09 03:10 PM - Giovanni Manghi

- Resolution set to end of life

- Status changed from Feedback to Closed

End of life notice: QGIS 2.18 LTR

Source:

<http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/>

QGIS 3.4 has recently become our new Long Term Release (LTR) version. This is a major step in our history – a long term release version based on the massive updates, library upgrades and improvements that we carried out in the course of the 2.x to 3x upgrade cycle.

We strongly encourage all users who are currently using QGIS 2.18 LTR as their preferred QGIS release to migrate to QGIS 3.4. This new LTR version will receive regular bugfixes for at least one year. It also includes hundreds of new functions, usability improvements, bugfixes, and other goodies. See the relevant changelogs for a good sampling of all the new features that have gone into version 3.4

Most plugins have been either migrated or incorporated into the core QGIS code base.

We strongly discourage the continued use of QGIS 2.18 LTR as it is now officially unsupported, which means we'll not provide any bug fix releases for it.

You should also note that we intend to close all bug tickets referring to the now obsolete LTR version. Original reporters will receive a notification of the ticket closure and are encouraged to check whether the issue persists in the new LTR, **in which case they should reopen the ticket.**

If you would like to better understand the QGIS release roadmap, check out our roadmap page! It outlines the schedule for upcoming releases and will help you plan your deployment of QGIS into an operational environment.

The development of QGIS 3.4 LTR has been made possible by the work of hundreds of volunteers, by the investments of companies, professionals, and administrations, and by continuous donations and financial support from many of you. We sincerely thank you all and encourage you to collaborate and support the project even more, for the long term improvement and sustainability of the QGIS project.

Files

M_multiline_test.zip	1.5 KB	2019-02-23	Alessandro Cristofori
m_values_qgis.png	228 KB	2019-02-23	Alessandro Cristofori
MvaluesArc.JPG	84.1 KB	2019-02-23	Alessandro Cristofori