Crash after bulk change of attribute value in shapefile

Description

Following the following steps will crash QGIS:

1. Open a new QGIS project and open the attached shapefile
2. Open the attribute table
3. Enable editing in the attribute table
4. Use the textbox for bulk changes to change the SUBTYPE attribute to NULL. (Use the 'Update All' button.)
5. Save the changes
6. QGIS crashed

This probably has something to do with the shapefile itself, as resaving the file removes the issue.

Notes:

- It affects also master
- It corrupts the data (qgis crashes when trying to reopen the shape)
- It affects also the field calculator
- It does not happen on qgis 2.2

History

#1 - 2014-10-16 05:02 AM - Giovanni Manghi
- Affected QGIS version changed from 2.4.0 to master
- Category set to Vectors
- Priority changed from Normal to Severe/Regression
- Target version set to Version 2.6

Notes:

- It affects also master
- It corrupts the data (qgis crashes when trying to reopen the shape)
- It affects also the field calculator
- It does not happen on qgis 2.2

#2 - 2014-10-16 07:18 AM - Salvatore Larosa
- Priority changed from Severe/Regression to Normal
- Category changed from Vectors to Data Provider/OGR
- Status changed from Open to Feedback

Notes:

- Ogr (1.11.0) is crashing too:
  
  #0 0x00007ff1b9b025 in ?? () from /lib/x86_64-linux-gnu/libc.so.6
  #1 0x00007ff78638e8 in SHPReadOGRObj ect (hSHP=hSHP@entry=0x60de70, iShape=iShape@entry=1, psShape=0x60e970,
Giovanni Manghi wrote:

I confirm the regression (it is also affected the field calculator).

ogrinfo shows it's a 3D Line string shape with all Z coordinate too . big. After the update, QGIS built with 1.11.1 crashes when reading the file, but so does ogrinfo 1.10.1 however still lists it (although all Z coordinates are reported 0 after the update) and QGIS built against 1.10.1 also shows it. Smells like a GDAL regression, although I just tried 2.0.1 and didn't build master against 1.10.1.
Jürgen Fischer wrote:

Giovanni Manghi wrote:

I confirm the regression (it is also affected the field calculator).

ogrinfo shows it’s a 3D Line string shape with all Z coordinate too_big. After the update, QGIS built with 1.11.1 crashes when reading the file, but so does ogrinfo 1.11.1. ogrinfo 1.10.1 however still lists it (although all Z coordinates are reported 0 after the update) and QGIS built against 1.10.1 also shows it. Smells like a GDAL regression, although I just tried 2.0.1 and didn’t build master against 1.10.1.

should we close this?

#8 - 2014-10-31 05:55 AM - Jürgen Fischer
- Target version changed from Version 2.6 to Future Release - High Priority

#9 - 2015-05-27 07:57 AM - Giovanni Manghi
- Priority changed from Normal to High

#10 - 2016-01-19 05:00 AM - Sandro Santilli

Still happens as of commit:69cb0c4ed3174946c82e32dad4af5a12275079fc (2.14-pre) built and running against GDAL 1.11.1

Backtrace:

#0 __memcpy_sse2_unaligned () at ../sysdeps/x86_64/multiarch/memcpy-sse2-unaligned.S:33
#1 0x00007f0c2d8eb858 in SHPReadOGRObject (hSHP=hSHP@entry=0x7f0b6c00cc00, iShape=iShape@entry=1, psShape=psShape@entry=0x7f0b6c019080) at shape2ogr.cpp:232
#2 0x00007f0c2d8ed1fd in SHPReadOGRFeature (hSHP=0x7f0b6c00cc00, hDBF=0x7f0b6c0299c0, poDefn=0x7f0b6c029cf0, iShape=1, psShape=0x7f0b6c019080, pszSHPEncoding=0x7f0c30fe63d8 <std::string::_Rep::_S_empty_rep_storage+24> **) at shape2ogr.cpp:1018
#3 0x00007f0c2d85f73a in OGRShapeLayer::GetNextFeature (this=0x7f0b6c00dea0) at ogrshapelayer.cpp:751
#4 0x00007f0b99e1a494 in QgsOgrFeatureIterator::fetchFeature (this=0x7f0b6c009b90, feature=...) at /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrfeatureiterator.cpp:213
#5 0x00007f0c3258f8c5 in QgsAbstractFeatureIterator::nextFeature (this=0x7f0b6c009b90, f=...) at /usr/src/qgis/qgis-master/src/core/qgsfeatureiterator.cpp:76
#6 0x00007f0c3270f8c5 in QgsVectorLayerFeatureIterator::fetchFeature (this=0x7f0b6c00d2c8, f=...) at /usr/src/qgis/qgis-master/src/core/qgsfeatureiterator.h:234
#7 0x00007f0c3270f8c5 in QgsVectorLayerFeatureIterator::nextFeature (this=0x7f0b6c00d2c8, f=...) at /usr/src/qgis/qgis-master/src/core/qgsvectorlayerfeatureiterator.cpp:237
#8 0x00007f0c3258f8c5 in QgsAbstractFeatureIterator::nextFeature (this=0x7f0b6c00d170, f=...) at /usr/src/qgis/qgis-master/src/core/qgsfeatureiterator.cpp:76
#9 0x00007f0c324f801e in QgsFeatureIterator::nextFeature (this=0x7f0b7effc8c0, f=...) at /usr/src/qgis/qgis-master/src/core/qgsfeatureiterator.h:234
#10 0x00007f0c32721684 in QgsVectorLayerRenderer::drawRendererV2 (this=0x2f604b0, ftt=...) at /usr/src/qgis/qgis-master/src/core/qgsvectorlayerrenderer.cpp:290

#11 - 2016-01-19 05:02 AM - Sandro Santilli
#12 - 2016-01-19 06:59 AM - Sandro Santilli

Filed GDAL ticket here: [https://trac.osgeo.org/gdal/ticket/6317](https://trac.osgeo.org/gdal/ticket/6317)

#13 - 2016-01-19 07:14 AM - Sandro Santilli

GDAL is responsible for the crash on reading, but I still don't know who's responsible for corruption on writing

#14 - 2016-01-19 07:46 AM - Sandro Santilli

So the GDAL bug is that the M values are dropped when updating XYM geometries, but the output is still advertised as containing them, which makes the reader crash.

If we want to implement a workaround on our side we need to drop the M value upfront, when we know the version of GDAL in use is bogus in that regard.

Note that if the input is an XYZM shapefile the M is dropped anyway.

#15 - 2016-01-19 08:01 AM - Even Rouault

Will be fixed in upcoming GDAL 1.11.4 & 2.0.2

#16 - 2016-01-19 08:14 AM - Sandro Santilli

- Status changed from In Progress to Feedback

I confirm current GDAL trunk fixes the issue.

A workaround is not trivial, maybe we should just refuse to allow editing of XYM input shapefiles IFF GDAL version is known to be broken, and be noisy about it.

Worth it?

#17 - 2016-01-19 08:44 AM - Sandro Santilli

So now that I finally got it I'll leave a note here:

OGR will not help in knowing if the input is or not affected by the bug in that it will transparently map the XYM input as an XYZ one, so to use the input will look like a normal XYZ. But will produce a corrupted output or not based on whether the actual shapefile being read was an XYM or a real XYZ, which we could only check by independently checking the shapefile input. Then, once the condition is spot, we should be rewriting the shapefile into a clean XYZ or XY in order to deal with it.

A lot of work for a workaround (would imply designing an interface to let the user know what we're doing with her data etc.).

#18 - 2016-01-19 08:48 AM - Sandro Santilli

- Status changed from Feedback to Closed
closing as fixed upstream -- remember that the M value is still lost, if you want to retain it it'd still take a refactoring in OGR.

#19 - 2016-01-19 08:56 AM - Sandro Santilli

I filed #14142 to not loose track of the lost-M issue

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>qgis-20141016-132831-7480-7088-8fdd08a.7z</td>
<td>2 MB</td>
<td>2014-10-16</td>
<td>Dieter De Paepe</td>
</tr>
<tr>
<td>file.zip</td>
<td>1.92 KB</td>
<td>2014-10-16</td>
<td>Dieter De Paepe</td>
</tr>
</tbody>
</table>