QGIS Application - Bug report #15570
Shapefile data corruption

2016-09-15 05:55 PM - Nyall Dawson

Status:	Closed		
Priority:	Severe/Regression		
Assignee:	Even Rouault		
Category:	Unknown		
Affected QGIS version:	2.16.2	**Regression?:**	No
Operating System:	Windows	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	up/downstream
Crashes QGIS or corrupts data:	Yes	**Copied to github as #:**	23493

Description

Really nasty bug: QGIS 2.16 (+master) can corrupt shapefiles in certain circumstances. I haven't been able to exactly track down a reproducible workflow for replicating this, but I every time I've encountered this has been doing operations like:

1. be working on a project containing a shapefile
2. make the shapefile editable, add some features
3. split one of these features
4. save
5. get a message "Cannot reopen datasource xxx.shp|layerid=0 in read-only mode" and "Data source is invalid (Unable to open xxx.shx or xxx.SHX.Try --config SHAPE_RESTORE_SHX true to restore or create it)" in the log.
6. The data is corrupted - the original shp file can no longer be opened.
7. In the same folder as the file there's additional files "xxx_packed.shp" and "xxx_packed.shx". Renaming the "xxx_packed.shx" file to "xxx.shx" allows the original shapefile to be reopened.

This is a very nasty regression - if users are unaware they can recover the file with the shx rename then they will assume the data is unrecoverable.

History

#1 - 2016-09-16 12:22 AM - Even Rouault

@nyall When this happens,
- do you still have xxx.shp and xxx.shx files in addition to xxx_packed.shp and xxx_packed.shx ?
- do you confirm that you must rename only xxx_packed.shx as xxx.shx, and not also xxx_packed.shp as xxx.shp ? That would then mean that xxx_packed.shp has already been renamed successfull as xxx.shp, in which case xxx_packed.shp shouldn't exist anymore. If xxx_packed.shp still exists, then you might have a (potential) consistency issue

The cause of this is that somethings locks the xxx.shp and/or xxx.shx files which prevents them from being deleted and their _packed versions (the new version that has been packed) to be renamed as xxx.shp/shx. But in that event, you should still be able to open the original xxx.shp/shx files (the ones before packing). So I'm perhaps missing something.

The issue might be instead that something (the browser? an antivirus ?) locks the _packed files, which prevents them from being renaming (but in that case the xxx.shp/shx should be deleted, unless they are locked too).

The related logic in the packing method in GDAL is there https://github.com/OSGeo/gdal/blob/ce6c6098ac4c9e6805a4c9d5c8ee0acace09191a/gdal/ogr/ogrsf_frmts/shape/ogrshapelayer.cpp#L2560 (oTempFile is the basename of the _packed files). I've just committed https://trac.osgeo.org/gdal/changeset/35467 in GDAL trunk and 2.1 branch to increase the verbosity when things go wrong (adding errors for failed VSIUnlink() and changing the debug traces into loud errors).

Hum seeing https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrprovider.cpp#L2951 , which is the only function to call repack(), I'm

wondering if the QgsOgrConnPool::instance()->invalidateConnections(dataSourceUri()) call done at line 2969 shouldn't be added as well to the top of repack(), around https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrprovider.cpp#L144 . The issue might be that a reader (the attribute table e.g) has a connection from the connection pool on the shapefile, which could interfere with deletion/renaming.

#2 - 2016-09-16 04:48 AM - Jukka Rahkonen

I have a feeling that this issue is related #15393. Perhaps the "read only" copies keep some files locked.

#3 - 2016-09-16 08:44 PM - Nyall Dawson

Even - thanks for the hints! Here's some answers:

- do you still have xxx.shp and xxx.shx files in addition to xxx_packed.shp and xxx_packed.shx ?

Yes, both those files are still present

- do you confirm that you must rename only xxx_packed.shx as xxx.shx, and not also xxx_packed.shp as xxx.shp ?

Yes, correct. If I also rename the _packed.shp then the file cannot be opened.

> *That would then mean that xxx_packed.shp has already been renamed successfull as xxx.shp, in which case xxx_packed.shp shouldn't exist anymore. If xxx_packed.shp still exists, then you might have a (potential) consistency issue*

> *The cause of this is that somethings locks the xxx.shp and/or xxx.shx files which prevents them from being deleted and their _packed versions (the new version that has been packed) to be renamed as xxx.shp/shx. But in that event, you should still be able to open the original xxx.shp/shx files (the ones before packing). So I'm perhaps missing something.*

Hmmm - it's possibly related, but the first time I encountered this I had the same project opened in two QGIS instances, so the shapefile would have been opened in both but being edited only in one. Maybe this situation could cause the rename to fail? Unfortunately I tried to reproduce this scenario immediately after but could not reproduce.

> *Hum seeing https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrprovider.cpp#L2951 , which is the only function to call repack(), I'm wondering if the QgsOgrConnPool::instance()->invalidateConnections(dataSourceUri()) call done at line 2969 shouldn't be added as well to the top of repack(), around https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrprovider.cpp#L144 . The issue might be that a reader (the attribute table e.g) has a connection from the connection pool on the shapefile, which could interfere with deletion/renaming.*

When I hit this issue on Friday I definitely had the attribute table window open for the layer - so maybe that is related. Here's the exact steps I did leading up to the issue:

1. was working on a shapefile with just 30 or so line features. 5 or 6 attributes, so not a complex layer.
2. copied and pasted a handful of features from another layer into this layer
3. from the attribute table "quick update" bar I changed a field's value for all these newly pasted features
4. selected a single feature, then used the split feature tool to cut it in half
5. deleted one half of the split feature
6. saved - error

This was the 3rd time I've hit this - the first 2 were a month or so ago (on 2.16), and I couldn't reproduce. This most recent occurance was using the master_2 builds from osgeo4w. Same machine though - using Windows 7 64 bit.

I'll do some more tests on monday and try to narrow it down to a reproducable workflow.

#4 - 2016-09-20 02:15 AM - Even Rouault

I can't make sense of the fact that if you rename both the _packed.shp and _packed.shx as non packed, you get a corrupted dataset. If both exist, they should be consistant between them. Would be potentially helpful that you attach the .shp, .shx, _packed.shp, _packed.shx, etc... files when a corruption occurs.

Have you noticed remains of _packed.dbf files as well ? If you delete records at some steps, which seem to be the case in your last case, there should be compaction of the dbf and a temporary _packed.dbf being creating as well

#5 - 2016-09-20 02:23 AM - Nyall Dawson

Even - I've got a copy of the corrupted data here (as it was immediately afterwards, with the _packed files) but cannot share publicly. I'll email it to you.

#6 - 2016-09-20 03:03 AM - Even Rouault

In the zip sent by Nyall, there's :
- a .dbf file that has 74 records, but with a hole at FID=72
- a .shp file, without .shx, that after using "--config SHAPE_RESTORE_SHX true" has 74 records. However when reading it, there's an error at FID=73 (shx reconstruction somehow failed, possibly because of the deleted feature in it)
- a _packed.shp and _packed.shx that have 73 features and are fully consistent (Nyall confirmed it by email that he wasn't sure about what he did regarding renaming last week)

The fact that there's no _packed.dbf file and the .dbf file has a hole would make think that DBF compaction didn't occur at all, which would be a different bug in itself, for which I don't have any explanation (or that for some weird reason the _packed.dbf file disappeared during the failed renaming to .dbf, but that doesn't make much sense).

Regarding the .shp,.shx, given that the order of operations is :
1. delete .shp
2. delete .shx
3. rename _packed.shp as .shp
3. rename _packed.shx as .shx

my theory is the following one :
1. delete .shp failed: a lock prevents it. But before my latest change in GDAL, this failure wasn't checked so execution continued
2. delete .shx succeeded : the shapefile driver ingests the whole .shx file at opening and don't maintain a file descriptor opened. Hence no reason deletion couldn't occur
3. rename _packed.shp as .shp failed: for the same reason as step 1. But here the failure is detected and the procedure stops here. Which explains for the fact that you have both _packed.shp and _packed.shx, that the old .shp remains and no more .shx file.

#7 - 2016-09-27 03:34 AM - Antoine de Beaurepaire

Having the same issue here on QGIS 2.14.6.

I have noticed a few weeks ago the same bug and managed to avoid it by waiting carfully during saving and not clicking at all in the QGIS interface.

Now, even if being very carful, the issue is still happening. Can't found any explanation.

I can give some more infos on demand if needed.

#8 - 2016-09-28 01:28 PM - Even Rouault

I spent the last hours replicating the issue, and was somehow successful in doing so. There are 2 different reliable scenarios I found:

A) First scenario :
1) Open one QGIS instance and open a shapefile
2) Open another QGIS instance and open the same shapefile
3) In one of the instances, turn on edition, delete a shape and save

Approximatively half of time, in the Log messages panel, the following error will be reported: "Possible corruption after REPACK detected. %1 still exists. This may point to a permission or locking problem of the original DBF.". But even when it is not reported, looking with ogrinfo shows that the .dbf file wasn't succesfully compacted (holes in feature identifiers).
The fact that the error is sometimes reported is due to a Windows particularity. If a process opens a file with FILE_SHARE_DELETE permission, then you can delete it succesfully, but if you check for the existence of the file it will be reported as still existing, until the last process that has a handle on it closes it. So ~ half of the time something transiently causes the _packed.dbf file to be still there when QGIS checks for it as a proof of something that went wrong. I manage to replicate that with a program that creates a file, deletes it, checks for its presence, and sometimes the file is reported to be present. I guess some processes in Windows (explorer, antivirus) opens the newly created file in a transient way.
Anyway in this scenario, the main issue is that the original .dbf file cannot be deleted since the other QGIS process still holds a file descriptor on it.

B) Second scenario :
1) Open one QGIS instance and open a shapefile
2) Open another QGIS instance and open the same shapefile
3) In one of the instances, turn on edition, add a ring to a polygon and save

You'll get a message "Cannot reopen datasource xxx.shp|layerid=0 in read-only mode" and "Data source is invalid (Unable to open xxx.shx or xxx.SHX.Try --config SHAPE_RESTORE_SHX true to restore or create it)" in the log. This is with released versions of GDAL. After the fix I did in https://trac.osgeo.org/gdal/changeset/35467, the failure to delete the old .shp file make it early exit, but the .shp isn't effectively compacted on inspection. For the same reason as the .dbf case: the other QGIS process holds a lock on the .shp file.

I guess that people can also reproduce those issues with a single process, but in a far less reliable way, if other processes transiently lock the files. Besides antivirus and other Windows services, the QGIS Browser panel and its background tasks probing for new files could also be a culprit.

One option I've tried is to add the FILE_SHARE_DELETE share permission (https://msdn.microsoft.com/en-us/library/windows/desktop/aa363858%28v=vs.85%29.aspx) to the permissions used by GDAL when calling CreateFile(). With that, you can effectively delete the file while still being opened elsewhere, but as the file is still reported as present for the above mentionned reason, you can not rename the _packed files over it. So we're stuck. I've tried the ReplaceFile() Win32 API rather than the delete + rename approach and it seems to work, even if the file to be replaced is opened, but that requires that all handles on it have FILE_SHARE_DELETE set, which we cannot control easily for external code.

I come to think that the only reliable solution is to modify the PACK implementation in the OGR Shapefile driver to :
1) Generate the packed files as currently done
2) Do not close the working .dbf, .shp, .shx file descriptors that we got in update mode at dataset opening (actually that will require changing the way we handle the .shx since it is currently closed just after shapefile opening)
3) Copy the _packed content onto the regular .dbf, .shp and .shx files, and truncate the files is they are shorter than before
4) If everything is successful, delete the _packed versions. And do something at the low level shapelib code to refresh its state.

And modify QGIS code to actually detect GDAL errors emitted by the repack function.

#9 - 2016-10-06 04:58 AM - Even Rouault

- Assignee set to Even Rouault

#10 - 2016-10-06 07:24 AM - Even Rouault

- Status changed from Open to Closed

- Resolution set to up/downstream

The fix is in GDAL https://trac.osgeo.org/gdal/ticket/6672. This will land in GDAL 2.1.2

I've committed an improvement in QGIS to raise a QGIS error when GDAL emits a GDAL error + test.

#11 - 2016-10-07 02:54 PM - Jamie Portman

Any idea when GDAL 2.1.2 will be released?
Also, has the new QGIS error check been added to both the LTR (2.14)and 2.16 versions, or just in 2.16?

#12 - 2017-09-22 09:55 AM - Jürgen Fischer

- Category set to Unknown