

QGIS Application - Bug report #16393

Crash when removing layer quickly after addition

2017-03-30 03:39 AM - Dario Goetz

Status:	Closed	
Priority:	High	
Assignee:		
Category:	Python plugins	
Affected QGIS version:	2.18.5	Regression?: No
Operating System:	Ubuntu	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: worksforme
Crashes QGIS or corrupts data:	Yes	Copied to github as #: 24302
Description		
The crash occurs when adding new shapefile-layers and quickly afterwards removing them again via the python interface (within a plugin). The crash is caused by an assert message in src/core/qgsconnectionpool.h: Fatal: ASSERT: "it != mGroups.end()" in file homedir/QGIS/src/providers/ogr/../../core/qgsconnectionpool.h, line 276 *How the crash can be produced The layers are added in a python plugin to the registry using "qgis.core.QgsMapLayerRegistry.instance().addMapLayer(layer, False)" and afterwards to the layer tree with "grp.addLayer(layer)", where "grp" is a QgsLayerTreeGroup. If such a layer is being removed using "qgis.core.QgsMapLayerRegistry.instance().removeMapLayers(layer_ids)", "grp.removeAllChildren()" and "grp.parent().removeChildNode(grp)" a race condition can cause the crash, if a feature iterator is still active, e.g. while rendering. Backtrace: <pre>#0 0x00007ffff3efc428 in __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:54 #1 0x00007ffff3efe02a in __GI_abort () at abort.c:89 #2 0x000000000406bfa in myMessageOutput (type=QtFatalMsg, msg=0x7fff240235b8 "ASSERT: \"it != mGroups.end()\" in file homedir/qgis/QGIS/src/providers/ogr/../../core/qgsconnectionpool.h, line 276") at homedir/qgis/QGIS/src/app/main.cpp:417 #3 0x00007ffff558be9f in qt_message_output (QtMsgType, char const*) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4 #4 0x00007ffff558c371 in ?? () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4 #5 0x00007ffff558cc91 in qFatal (char const*, ...) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4 #6 0x00007fff531f3536 in QgsConnectionPool<QgsOgrConn*, QgsOgrConnPoolGroup>::releaseConnection (this=0x5a4a6e0, conn=0x7fff2402eff0) at homedir/qgis/QGIS/src/providers/ogr/../../core/qgsconnectionpool.h:276 #7 0x00007fff531f1c46 in QgsOgrFeatureIterator::close (this=0x7fff240549f0) at homedir/qgis/QGIS/src/providers/ogr/qgsogrfeatureiterator.cpp:269 #8 0x00007fff531f1ac2 in QgsOgrFeatureIterator::fetchFeature (this=0x7fff240549f0, feature=...) at homedir/qgis/QGIS/src/providers/ogr/qgsogrfeatureiterator.cpp:232 #9 0x00007ffff5e7136a in QgsAbstractFeatureIterator::nextFeature (this=0x7fff240549f0, f=...) at homedir/qgis/QGIS/src/core/qgsfeatureiterator.cpp:73 #10 0x00007ffff5da3afa in QgsFeatureIterator::nextFeature (this=0x7fff2402eef0, f=...) at homedir/qgis/QGIS/src/core/qgsfeatureiterator.h:280 #11 0x00007ffff604c54b in QgsVectorLayerFeatureIterator::fetchFeature (this=0x7fff2402ed90, f=...) at homedir/qgis/QGIS/src/core/qgsvectorlayerfeatureiterator.cpp:283 #12 0x00007ffff5e7136a in QgsAbstractFeatureIterator::nextFeature (this=0x7fff2402ed90, f=...) at homedir/qgis/QGIS/src/core/qgsfeatureiterator.cpp:73 #13 0x00007ffff5da3afa in QgsFeatureIterator::nextFeature (this=0x7fff446780d0, f=...) at homedir/qgis/QGIS/src/core/qgsfeatureiterator.h:280 #14 0x00007ffff605f80a in QgsVectorLayerRenderer::drawRendererV2 (this=0x58395d0, fit=...) at</pre>		

```

homedir/qgis/QGIS/src/core/qgsvectorlayerrenderer.cpp:285
#15 0x00007ffff605f263 in QgsVectorLayerRenderer::render (this=0x58395d0) at
homedir/qgis/QGIS/src/core/qgsvectorlayerrenderer.cpp:256
#16 0x00007ffff5eed102 in QgsMapRendererCustomPainterJob::doRender (this=0x5838ef0) at
homedir/qgis/QGIS/src/core/qgsmaprenderercustompainterjob.cpp:273
#17 0x00007ffff5eedc60 in QgsMapRendererCustomPainterJob::staticRender (self=0x5838ef0) at
homedir/qgis/QGIS/src/core/qgsmaprenderercustompainterjob.cpp:230
#18 0x00007ffff5eef6ed in QtConcurrent::StoredFunctorCall1<void, void (*) (QgsMapRendererCustomPainterJob*),
QgsMapRendererCustomPainterJob*>::runFunctor (this=0x55a11d0) at /usr/include/qt4/QtCore/qtconcurrentstoredfunctioncall.h:277
#19 0x00007ffff5eedff5 in QtConcurrent::RunFunctionTask<void*>::run (this=0x55a11d0) at
/usr/include/qt4/QtCore/qtconcurrentrunbase.h:134
#20 0x00007ffff5589e0a in ?? () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#21 0x00007ffff5596e3c in ?? () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#22 0x00007fffdada46ba in start_thread (arg=0x7fff44679700) at pthread_create.c:333
#23 0x00007ffff3fcd82d in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:109

```

Excerpt of log messages that I used in order to investigate the problem:

```

src/providers/ogr/qgsogrprovider.cpp: 427: (QgsOgrProvider) [0ms] ----- Referencing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0 -----
src/providers/ogr/qgsogrprovider.cpp: 427: (QgsOgrProvider) [0ms] ----- Referencing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0 -----
src/providers/ogr/qgsogrprovider.cpp: 427: (QgsOgrProvider) [0ms] ----- Referencing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0 -----
src/providers/ogr/qgsogrprovider.cpp: 427: (QgsOgrProvider) [0ms] ----- Referencing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0 -----
src/providers/ogr/qgsogrfeatureiterator.cpp: 360: (QgsOgrFeatureSource) [1ms] ----- Referencing connection pool (FeatureIterator) -
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' -----
src/providers/ogr/qgsogrfeatureiterator.cpp: 360: (QgsOgrFeatureSource) [0ms] ----- Referencing connection pool (FeatureIterator) -
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' AND SRV = 0 -----
src/providers/ogr/./../core/qgsconnectionpool.h: 255: (acquireConnection) [1ms] [thread:0x5130b10] ----- Acquire connection
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' -----
src/providers/ogr/qgsogrprovider.cpp: 433: (~QgsOgrProvider) [0ms] ----- Closing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' AND SRV = 0 -----
src/providers/ogr/qgsogrprovider.cpp: 433: (~QgsOgrProvider) [1ms] ----- Closing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' AND SRV = 1 -----
src/providers/ogr/qgsogrprovider.cpp: 433: (~QgsOgrProvider) [0ms] ----- Closing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' AND SRV = 2 -----
src/providers/ogr/qgsogrprovider.cpp: 433: (~QgsOgrProvider) [0ms] ----- Closing connection pool (Provider) -
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' -----
src/providers/ogr/./../core/qgsconnectionpool.h: 273: (releaseConnection) [5ms] [thread:0x5130b10] ----- Releasing connection
srv/111230/influence/polygons.shp|layerid=0|subset=CELLID = '54095164' -----
Fatal: ASSERT: "it != mGroups.end()" in file /home/dg/qgis/QGIS/src/providers/ogr/./../core/qgsconnectionpool.h, line 276

```

The release of the connection for which the connection has been unreferenced by the destructor of QgsOgrProvider before causes the crash.

History

#1 - 2017-03-30 03:51 AM - Dario Goetz

Some preliminary findings from my side.

Using log messages and gdb, I observe two issues in my crash example:

- 1. The "QgsOgrProvider" gets destroyed while a "QgsOgrFeatureSource" (of some "QgsOgrFeatureIterator") still has an acquired connection. When trying to release the connection, the corresponding "QgsConnectionPoolGroup" has already been removed, hence the assert failure.
- 2. The connections in the "QgsConnectionPool" are indexed by the providers "dataSourceUri". When using the "setSubString" method of the "QgsOgrProvider", the URI may change. If a connection to the "QgsConnectionPool" has been referenced beforehand, however, that connection still indexed via the old URI. In that case, references to the corresponding "QgsOgrConnPoolGroup" are not handled well anymore. I don't know whether this is how it is supposed to be, but it seems weird to me. I have added a log message for corresponding URI changes in the attached log file ("src/providers/ogr/qgsogrprovider.cpp: 505: (setSubString) [0ms] ----- URI changed ...")

#2 - 2017-04-30 12:03 PM - Giovanni Manghi

- Priority changed from Normal to High
- Affected QGIS version changed from 2.18.4 to 2.18.5
- Status changed from Open to Feedback
- Description updated
- Subject changed from QGIS 2.18.5 - Crash when removing layer quickly after addition to Crash when removing layer quickly after addition

Does it happens on qgis3/master?

#3 - 2017-05-01 01:01 AM - Giovanni Manghi

- Easy fix? set to No
- Regression? set to No

#4 - 2017-05-26 05:18 PM - Giovanni Manghi

- Status changed from Feedback to Closed
- Resolution set to worksforme

Closing for lack of feedback.

Files

log_2017_03_30	329 KB	2017-03-30	Dario Goetz
----------------	--------	------------	-------------