

QGIS Application - Bug report #20337

Crash loading layer from DB2 spatial table

2018-11-02 08:13 PM - David Adler

Status:	Open	
Priority:	High	
Assignee:	David Adler	
Category:	Data Provider	
Affected QGIS version:	3.3(master)	Regression?: Yes
Operating System:	Windows	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution:
Crashes QGIS or corrupts data:	Yes	Copied to github as #: 28158

Description

User Feedback

Attempting to load a layer from a DB2 spatial table when double-clicking on a DB2 table name.
This information is from running without debugging.
When running the debugger, it appears to be crashing in the ODBC code called by the QSqlQuery::prepare method.

It worked fine in QGIS 2.18 but fails most but not all the time in QGIS 3.3 and when built from QGIS Master.

I'm wondering if the problem could have something to do with threading in the DB2 provider code.
When the code crashes, it appears that the qgsdb2featureiterator and qgsdb2provider code is running in different threads for the same request to access a layer.
This is a snip of the trace when it fails:
E:\qgisdev\src\core\qgsmaplayer.cpp(178) : (QgsMapLayer::name) [2ms] returning name 'CUSTOMERS'
E:\qgisdev\src\core\qgscoordinatetransform_p.cpp(210) : (QgsCoordinateTransformPrivate::initialize) [3ms] Source/Dest CRS not equal, shortcircuit is not set.
E:\qgisdev\src\core\qgscoordinatetransform_p.cpp(210) : (QgsCoordinateTransformPrivate::initialize) [2ms] Source/Dest CRS not equal, shortcircuit is not set.
E:\qgisdev\src\core\qgscoordinatetransform_p.cpp(210) : (QgsCoordinateTransformPrivate::initialize) [3ms] Source/Dest CRS not equal, shortcircuit is not set.
E:\qgisdev\src\core\qgsmaprenderparalleljob.cpp(64) : (QgsMapRendererParallelJob::start) [2ms] QThreadPool max thread count is 8
E:\qgisdev\src\core\qgsmaprenderparalleljob.cpp(253) : (QgsMapRendererParallelJob::renderLayerStatic) [4ms]
[thread:0x2eb55698840] job 2eb57cecc0 start (layer CUSTOMERS_e0a5be6a_6d31_4690_a52d_d10756f7ce67)
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(235) : (QgsDb2FeatureIterator::BuildStatement) [4ms] [thread:0x2eb55698840]
compileExpressions: true
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(288) : (QgsDb2FeatureIterator::BuildStatement) [3ms] [thread:0x2eb55698840]
SELECT OBJECTID,DB2GSE.ST_ASBINARY(GEOM) AS GEOM FROM OSUSER.CUSTOMERS WHERE DB2GSE.
ENVELOPESINTERSECT = 1
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(164) : (QgsDb2Provider::getDatabase) [3ms] [thread:0x2eb55698840] expanded
connInfo: dbname='OSTEST' service='OSTEST64'
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(174) : (QgsDb2Provider::getDatabase) [4ms] [thread:0x2eb55698840] driver: "; host:
"; databaseName: 'OSTEST'
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(188) : (QgsDb2Provider::getDatabase) [3ms] [thread:0x2eb55698840]
connectionName: OSTEST64
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(197) : (QgsDb2Provider::getDatabase) [3ms] [thread:0x2eb55698840] found existing
connection, use the existing one
E:\qgisdev\src\core\qgscredentials.cpp(68) : (QgsCredentials::put) [1095ms] [thread:0x2eb5590cf90] inserting realm:OSTEST
username:osuser password:osuserpw
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(418) : (QgsDb2FeatureIterator::rewind) [0ms] [thread:0x2eb5590cf90] Execute
mStatement: SELECT OBJECTID,DB2GSE.ST_ASBINARY(GEOM) AS GEOM FROM OSUSER.CUSTOMERS WHERE DB2GSE.
ENVELOPESINTERSECT = 1
E:\qgisdev\src\core\qgscredentials.cpp(48) : (QgsCredentials::get) [8ms] [thread:0x2eb55698840] retrieved realm:OSTEST
username:osuser password:osuserpw
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(242) : (QgsDb2Provider::getDatabase) [7ms] [thread:0x2eb55698840] ODBC
connection string: OSTEST64
E:\qgisdev\src\app\qgscrashhandler.cpp(34) : (QgsCrashHandler::handle) [336ms] [thread:0x2eb5590cf90] CRASH!!!
E:\qgisdev\src\app\qgscrashhandler.cpp(57) : (QgsCrashHandler::handle) [9ms] [thread:0x2eb5590cf90]
C:/Users/davea2/AppData/Local/Temp/qgis-crash-info-12560
E:\qgisdev\src\app\qgscrashhandler.cpp(102) : (QgsCrashHandler::handle) [14ms] [thread:0x2eb5590cf90]
E:/qgisdev/ms-windows/osgeo4w/build-qgis-test-x86_64/output/bin/RelWithDebInfo/qgiscrashhandler.exe

This is a snip of the trace when it succeeds:

```
E:\qgisdev\src\core\qgsmaplayer.cpp(178) : (QgsMapLayer::name) [3ms] returning name 'CUSTOMERS'
E:\qgisdev\src\core\qgscoordinatetransform_p.cpp(210) : (QgsCoordinateTransformPrivate::initialize) [2ms] Source/Dest CRS not equal,
shortcircuit is not set.
E:\qgisdev\src\core\qgscoordinatetransform_p.cpp(210) : (QgsCoordinateTransformPrivate::initialize) [2ms] Source/Dest CRS not equal,
shortcircuit is not set.
E:\qgisdev\src\core\qgscoordinatetransform_p.cpp(210) : (QgsCoordinateTransformPrivate::initialize) [3ms] Source/Dest CRS not equal,
shortcircuit is not set.
E:\qgisdev\src\core\qgsmaprenderparalleljob.cpp(64) : (QgsMapRendererParallelJob::start) [2ms] QThreadPool max thread count is 8
E:\qgisdev\src\core\qgsmaprenderparalleljob.cpp(253) :
(QgsMapRendererParallelJob::renderLayerE:\qgisdev\src\core\qgsmaprenderparalleljob.cpp(253) :
(QgsMapRendererParallelJob::renderLayerStatic) [0ms] [thread:0x24fda6eb760] job 24fdd489b70 start (layer
SpatialTable_1daffcad_555d_4c84_8b91_ada8a3d72729)
E:\qgisdev\src\core\qgsmaplayer.cpp(178) : (QgsMapLayer::name) [3ms] returning name 'SpatialTable'
E:\qgisdev\src\providers\mssql\qgsmssqlfeatureiterator.cpp(293) : (QgsMssqlFeatureIterator::BuildStatement) [1ms]
[thread:0x24fda6eb760] SELECT [id],[GeomCol1] FROM [dbo].[SpatialTable] WHERE [GeomCol1].STIsValid() = 1 AND
[GeomCol1].Filter([geometry]::STGeomFromText('POLYGON',4326)) = 1
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(235) : (QgsDb2FeatureIterator::BuildStatement) [0ms] [thread:0x24fdb506a80]
compileExpressions: true
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(288) : (QgsDb2FeatureIterator::BuildStatement) [2ms] [thread:0x24fdb506a80]
SELECT OBJECTID,DB2GSE.ST_ASBINARY(GEOM) AS GEOM FROM OSUSER.CUSTOMERS WHERE DB2GSE.
ENVELOPESINTERSECT = 1
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(164) : (QgsDb2Provider::getDatabase) [3ms] [thread:0x24fdb506a80] expanded
connInfo: dbname='OSTEST' service='OSTEST64'
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(174) : (QgsDb2Provider::getDatabase) [3ms] [thread:0x24fdb506a80] driver: ""; host: "";
databaseName: 'OSTEST'
E:\qgisdev\src\core\qgsmaprenderparalleljob.cpp(273) : (QgsMapRendererParallelJob::renderLayerStatic) [1ms]
[thread:0x24fda6eb760] job 24fdd489b70 end [21 ms] (layer SpatialTable_1daffcad_555d_4c84_8b91_ada8a3d72729)
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(188) : (QgsDb2Provider::getDatabase) [2ms] [thread:0x24fdb506a80]
connectionName: OSTEST64
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(197) : (QgsDb2Provider::getDatabase) [3ms] [thread:0x24fdb506a80] found existing
connection, use the existing one
E:\qgisdev\src\core\qgscredentials.cpp(48) : (QgsCredentials::get) [1322ms] [thread:0x24fdb506a80] retrieved realm:OSTEST
username:osuser password:osuserpw
E:\qgisdev\src\providers\db2\qgsdb2provider.cpp(242) : (QgsDb2Provider::getDatabase) [4ms] [thread:0x24fdb506a80] ODBC
connection string: OSTEST64
E:\qgisdev\src\core\qgscredentials.cpp(68) : (QgsCredentials::put) [1874ms] [thread:0x24fdb506a80] inserting realm:OSTEST
username:osuser password:osuserpw
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(419) : (QgsDb2FeatureIterator::rewind) [13ms] [thread:0x24fdb506a80] Execute
mStatement: SELECT OBJECTID,DB2GSE.ST_ASBINARY(GEOM) AS GEOM FROM OSUSER.CUSTOMERS WHERE DB2GSE.
ENVELOPESINTERSECT = 1
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(426) : (QgsDb2FeatureIterator::rewind) [342ms] [thread:0x24fdb506a80] leaving
rewind
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(427) : (QgsDb2FeatureIterator::rewind) [8ms] [thread:0x24fdb506a80]
E:\qgisdev\src\providers\db2\qgsdb2featureiterator.cpp(310) : (QgsDb2FeatureIterator::fetchFeature) [8ms] [thread:0x24fdb506a80]
Entering fetchFeature
E:\qgi
```

Report Details

Crash ID: 597ecc7423c2116f73ac10c08f473806a26f2353

Stack Trace

```
RtlEnterCriticalSection :
ODBCSetTryWaitValue :
SQLExecute :
QgsDb2FeatureIterator::rewind qgsdb2featureiterator.cpp:427
QgsDb2FeatureIterator::QgsDb2FeatureIterator qgsdb2featureiterator.cpp:68
QgsDb2FeatureSource::getFeatures qgsdb2featureiterator.cpp:485
QgsVectorLayerFeatureIterator::QgsVectorLayerFeatureIterator qgsvectorlayerfeatureiterator.cpp:264
QgsVectorLayerFeatureSource::getFeatures qgsvectorlayerfeatureiterator.cpp:98
QgsVectorLayerRenderer::render qgsvectorlayerrenderer.cpp:237
QgsMapRendererParallelJob::renderLayerStatic qgsmaprendererparalleljob.cpp:256
QtConcurrent::FunctionWrapper1<void,LayerRenderJob & __ptr64>::operator() qtconcurrentfunctionwrappers.h:84
QtConcurrent::MapKernel<QList<LayerRenderJob>::iterator,QtConcurrent::FunctionWrapper1<void,LayerRenderJob & __ptr64>
>::runIteration qtconcurrentmapkernel.h:70
QtConcurrent::MapKernel<QList<LayerRenderJob>::iterator,QtConcurrent::FunctionWrapper1<void,LayerRenderJob & __ptr64>
>::runIterations qtconcurrentmapkernel.h:79
QtConcurrent::IterateKernel<QList<LayerRenderJob>::iterator,void>::forThreadFunction qtconcurrentiteratekernel.h:256
QtConcurrent::IterateKernel<QList<LayerRenderJob>::iterator,void>::threadFunction qtconcurrentiteratekernel.h:218
QtConcurrent::ThreadEngineBase::run :
QThreadPoolPrivate::reset :
QThread::start :
BaseThreadInitThunk :
RtlUserThreadStart :
```

QGIS Info

QGIS Version: 3.3.0-Master
QGIS code revision: commit:dfb28fed44
Compiled against Qt: 5.9.2
Running against Qt: 5.9.2
Compiled against GDAL: 2.2.4
Running against GDAL: 2.2.4

System Info

CPU Type: x86_64
Kernel Type: winnt
Kernel Version: 10.0.17134

History

#1 - 2018-11-02 09:52 PM - David Adler

- Assignee set to David Adler

I think this has to do with threading between getting database connection and executing query - researching.

#2 - 2018-11-03 02:58 AM - Nyal Dawson

There's been **many** bug fixes added to the mssql provider since it was forked for the DB2 provider. I'd suggest starting by looking at the git history of the mssql provider and porting all the relevant fixes.

#3 - 2018-11-08 01:31 PM - Giovanni Manghi

- Status changed from Open to Feedback

Please try on QGIS 3.4.1, if the issue is still valid change the affected version, thanks.

#4 - 2018-11-08 11:00 PM - David Adler

- Status changed from Feedback to Open

Nyall - thank you for the feedback.

Migrating over the mssql changes to DB2 seems to have fixed the problem which had to do with reusing the connection with different threads.

#5 - 2018-11-09 02:10 AM - Nyall Dawson

Ah - that thorny issue! It's a direct result of change in behavior with Qt 5.11, but fixing it should also make the provider more stable with earlier Qt versions.

Do you have a PR inbound with your fixes?

I'd also encourage you to try running the db2 provider tests again if you've got an up to date build. There's been a large number of additional checks added to that test since the db2 provider was added, but unfortunately I'm not aware of any core developers with access to a db2 instance in order to run these newer checks on a real db2 database. The provider test is likely to expose some more issues which may need addressing to further improve the provider's stability.