I get a segfault running test -V -R qgis_analyzertest.

The standalone test is output/bin/qgis_analyzertest, which segfaults on exit:

```plaintext
PASS  : TestQgsVectorAnalyzer::cleanupTestCase()
Totals: 10 passed, 0 failed, 0 skipped
******* Finished testing of TestQgsVectorAnalyzer *********
```

Program received signal SIGSEGV, Segmentation fault.

```
0x0000000000000141 in ?? ()
(gdb) bt
#0  0x0000000000000141 in ?? ()
#1  0x00007fffbfbc71ad in qgsConnectionPool_ConnectionDestroy (c=0x954500) at
    /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrconnpool.h:45
#2  0x00007fffbfbc7fca in QgsConnectionPoolGroup<QgsOgrConn*>::~QgsConnectionPoolGroup (this=0x90c410,
    __in_chrg=<optimized out>)
    at /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrconnpool.h:77
#3  0x00007fffbfbd973c in QgsOgrConnPoolGroup::~QgsOgrConnPoolGroup (this=0x90c400, __in_chrg=<optimized out>)
    at /usr/src/qgis/build/master/src/providers/ogr/../../core/qgsconnectionpool.h:59
#4  0x00007fffbfbd9778 in QgsOgrConnPoolGroup::~QgsOgrConnPoolGroup (this=0x90c400, __in_chrg=<optimized out>)
    at /usr/src/qgis/build/master/src/providers/ogr/../../core/qgsconnectionpool.h:59
#5  0x00007fffbfbd75d4 in QgsConnectionPool<QgsOgrConn*, QgsOgrConnPoolGroup>::~QgsConnectionPool (this=0x7fffbfde300
    <QgsOgrConnPool::sInstance>, __in_chrg=<optimized out>)
    at /usr/src/qgis/qgis-master/src/providers/ogr/../../core/qgsconnectionpool.h:238
#6  0x00007fffbfbd726d in QgsOgrConnPool::~QgsOgrConnPool (this=0x7fffbfde300 <QgsOgrConnPool::sInstance>,
    __in_chrg=<optimized out>)
    at /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrconnpool.cpp:31
#7  0x00007fffb5328259 in __run_exit_handlers (status=0, listp=0x7fffb56aa6c8 <__exit_funcs>,
    run_list_atexit=run_list_atexit@entry=true)
    at exit.c:82
```

Happens to me as of commit 80e3f8fc749e31d19667665fb90c9fb1a64d7f3f

2021-08-01
Ensure GDAL deinitialization runs after last possible use

Closes #14176

History

#1 - 2016-02-01 09:45 AM - Sandro Santilli

Still happening as of today. Any idea?

#2 - 2016-02-01 09:48 AM - Sandro Santilli

Valgrind view on the matter:

```
src/providers/postgres/qgspostgresconnpool.cpp: 33: (~QgsPostgresConnPool) [36ms] Entering.
src/providers/postgres/qgspostgresconnpool.cpp: 33: (~QgsPostgresConnPool) [1ms] Leaving.
src/providers/ogr/qgsogrconnpool.cpp: 33: (~QgsOgrConnPool) [14ms] Entering.
src/providers/ogr/qgsogrconnpool.cpp: 33: (~QgsOgrConnPool) [1ms] Leaving.
==17630== Invalid read of size 8
==17630== at 0x95CF5A5: OGR_DS_Destroy (ogrdatasource.cpp:69)
==17630== by 0x2E1EB54C: qgsConnectionPool_ConnectionDestroy(QgsOgrConn*) (qgsogrconnpoolpool.h:45)
==17630== by 0x2E1EC369: QgsConnectionPoolGroup<QgsOgrConn*>::~QgsConnectionPoolGroup() (qgsconnectionpoolpool.h:77)
==17630== by 0x2E1FDB89: QgsOgrConnPoolGroup::~QgsOgrConnPoolGroup() (in /usr/src/qgis/build/0-master/output/lib/qgis/plugins/libogrprovider.so)
==17630== by 0x2E1FDBC5: QgsOgrConnPoolPoolGroup::~QgsOgrConnPoolPoolGroup() (qgsogrconnpoolpool.h:59)
==17630== by 0x2E1FBA21: QgsConnectionPoolPool<QgsOgrConn*, QgsOgrConnPoolGroup>::~QgsConnectionPoolPool() (qgsconnectionpoolpool.h:238)
==17630== by 0x2E1F6B6A: QgsOgrConnPool::~QgsOgrConnPool() (qgsogrconnpool.cpp:31)
==17630== by 0x75A0258: __run_exit_handlers (exit.c:82)
==17630== by 0x75A02A4: exit (exit.c:104)
==17630== by 0x7585ECB: (below main) (libc-start.c:321)
==17630==  Address 0x320543c0 is 0 bytes inside a block of size 280 free'd
==17630== at 0xC2C131: operator delete(void*) (vg_replace_malloc.c:510)
==17630== by 0x9379FF5: GDALDriverManager::~GDALDriverManager() (gdaldrivermanager.cpp:183)
==17630== by 0x937A158: GDALDriverManager::~GDALDriverManager() (gdaldrivermanager.cpp:289)
==17630== by 0x2BC2EB4D: cleanupProvider (qsgdalprovider.cpp:3024)
==17630== by 0x5D0F0F1: QgsProviderRegistry::clean() (qgsproviderregistry.cpp:244)
==17630== by 0x5D0F205: QgsProviderRegistry::~QgsProviderRegistry() (qgsproviderregistry.cpp:253)
==17630== by 0x5D0F2A5: QgsProviderRegistry::~QgsProviderRegistry() (qgsproviderregistry.cpp:254)
==17630== by 0x5AF7EA2: QgsApplication::exitQgis() (qgsapplication.cpp:868)
==17630== by 0x4066064: TestQgsVectorAnalyzer::cleanupTestCase() (testqgsvectoranalyzer.cpp:89)
==17630== by 0x406566: TestQgsVectorAnalyzer::qt_static_metacall(QObject*, QMetaObject::Call, int, void**) (testqgsvectoranalyzer.cpp:89)
==17630== by 0x4FBB907: QMetaMethod::invoke(QObject*, QMetaObject::Call, int, void**) (testqgsvectoranalyzer.cpp:89)
==17630== by 0x53240BE: ??? (in /usr/lib/x86_64-linux-gnu/libQtCore.so.4.8.6)
==17630== by 0x53240BE: ??? (in /usr/lib/x86_64-linux-gnu/libQtTest.so.4.8.6)
==17630==
```
I'm not seeing that locally, and it works on Travis under Linux/OSX and appveyor under windows.

Maybe try fully deleting your build folder and rebuilding?

The only tests with issues (that I'm aware of) are:

- QgsBlendModes - occasionally fails on Windows. I'm having a hard time tracking this down.
- PyQgsComposerMap - as above, likely same issue
- PyQgsRuleBasedRenderer - occasionally crashes on exit.

My build configuration (while I clean-rebuild):

```c
mkdir \n-cmake \n  -D CMAKE_BUILD_TYPE=Debug \n  -D WITH_SERVER=ON \n  -D WITH_STAGED_PLUGINS=ON \n  -D WITH_PYSPATIALITE=ON \n  -D ENABLE_TESTS=1 \n  -D CMAKE_CXX_COMPILER:FILEPATH=/usr/lib/ccache/g++ \n  -D WITH_ASTYLE=1 \n  -D WITH_INTERNAL_QWTPOLAR=1
```

Still happens on a clean rebuild. GDAL version is 2.1.0. Segfault is on exit, so might be related to how the compiler chooses to order deinitialized:

```
(gdb) bt
#0 0x0000000000000161 in ?? ()
#1 0x00007fadac66577 in qgsConnectionPool_ConnectionDestroy (c=0x16fad60) at
    /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrconnpool.h:45
#2 0x00007fadac67394 in QgsConnectionPoolGroup<QgsOgrConn*>::QgsConnectionPoolGroup (this=0x15fde30, __in_chrg=<optimized out>)
    at /usr/src/qgis/build/master/src/providers/ogr/../../core/qgsconnectionpool.h:77
#3 0x00007fadac67394 in QgsOgrConnPoolGroup::QgsOgrConnPoolGroup (this=0x15fde30, __in_chrg=<optimized out>)
    at /usr/src/qgis/build/master/src/providers/ogr/../../../../../qgis-master/src/providers/ogr/qgsogrconnpool.h:59
#4 0x00007fadac6788f0 in QgsOgrConnPoolGroup::QgsOgrConnPoolGroup (this=0x15fde20, __in_chrg=<optimized out>)
    at /usr/src/qgis/build/master/src/providers/ogr/../../../../../qgis-master/src/providers/ogr/qgsogrconnpool.h:59
#5 0x00007fadac676a4c in QgsOgrConnectionPool<QgsOgrConn*, QgsOgrConnPoolGroup>::QgsOgrConnectionPool (this=0x7fadca688310 <QgsOgrConnPool::sInstance>, __in_chrg=<optimized out>)
    at /usr/src/qgis/qgis-master/src/providers/ogr/../../core/qgsconnectionpool.h:238
#6 0x00007fadac6766e5 in QgsOgrConnPool::QgsOgrConnPool (this=0x7fadca68b310 <QgsOgrConnPool::sInstance>, __in_chrg=<optimized out>)
```
I see code in QgsOgrConnPool class that seem to protect against a call to :instance() happening after a call to the destructor, and it surprises me such an occurrence may actually happen (why should it?).

Allocating the singleton on the heap and letting it leak fixes the segfault for me, see https://github.com/qgis/QGIS/pull/2754

Next catch.

For completeness - there is one other test with issues. QgsLegendRenderer fails occasionally under windows.

There is also output/bin/qgis_diagramtest (#14212) which seems to be another symptom of the same bug.

Fixed in changeset commit:"051253888810b06f6b055bfeae57a7c6a009e3f4c".