### QGIS Application - Bug report #14176

**test -V -R qgis_analyzertest segfaults**

2016-01-27 04:01 AM - Sandro Santilli

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>Sandro Santilli</td>
</tr>
<tr>
<td>Category:</td>
<td>Data Provider/OGR</td>
</tr>
<tr>
<td>Affected QGIS version:</td>
<td>master</td>
</tr>
<tr>
<td>Regression?:</td>
<td>No</td>
</tr>
<tr>
<td>Easy fix?:</td>
<td>No</td>
</tr>
<tr>
<td>Resolution:</td>
<td></td>
</tr>
<tr>
<td>Copied to github as #:</td>
<td>22178</td>
</tr>
</tbody>
</table>

### Description

I get a segfault running `test -V -R qgis_analyzertest`. The standalone test is output/bin/qgis_analyzertest, which segfaults on exit:

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

Program received signal SIGSEGV, Segmentation fault.

```
0x000000000001411 in ?? ()
(gdb) bt
#0 0x000000000001411 in ?? ()
#1 0x000000000001411 in ?? ()
#2 0x000000000001411 in qgsConnectionPool_ConnectionDestroy (c=0x954500) at
/usr/src/qgis/qgis-master/src/providers/qgsogrconnpool.h:45
#3 0x000000000001411 in qgsOgrConnGroup<qgsOgrConn*>::~qgsOgrConnGroup (this=0x90c410,
__in_chrg=<optimized out>)
at /usr/src/qgis/qgis-master/src/providers/qgsogrconnpool.h:77
#4 0x000000000001411 in qgsOgrConnPoolGroup<qgsOgrConnPool*>::~qgsOgrConnPoolGroup (this=0x90c400, __in_chrg=<optimized out>)
at /usr/src/qgis/qgis-master/src/providers/qgsogrconnpool.h:59
#5 0x000000000001411 in QgsConnectionPool<qgsOgrConn*, QgsOgrConnPoolGroup>::~QgsConnectionPool (this=0x7ffbfdeb300 <QgsOgrConnPool::sInstance>, __in_chrg=<optimized out>)
at /usr/src/qgis/qgis-master/src/providers/qgsogrconnpool.cpp:238
#6 0x000000000001411 in QgsOgrConnPool<qgsOgrConnPool*>::~QgsOgrConnPool (this=0x7ffbfdeb300 <QgsOgrConnPool::sInstance>,
__in_chrg=<optimized out>)
at /usr/src/qgis/qgis-master/src/providers/qgsogrconnpool.cpp:31
#7 0x000000000001411 in __run_exit_handlers (status=0, listp=0x7ffbf56a6c8 <__exit_funcs>,
run_list_atexit=run_list_atexit@entry=true)
at exit.c:82
```

Happens to me as of commit 80e3f8fc749e31d19667665fb90c9fb1a64d7f3f
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:

```c
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*) (qgsogrconnpool.h:45)
==17630==    by 0x2E1EC369: QgsConnectionPoolGroup<QgsOgrConn*>::~QgsConnectionPoolGroup() (qgsconnectionpool.h:77)
==17630==    by 0x2E1FD889: QgsOgrConnPoolGroup::~QgsOgrConnPoolGroup() (qgsogrconnpool.cpp:31)
==17630==     by 0x75A0258: __run_exit_handlers (exit.c:82)
==17630==     by 0x75A02A4: exit (exit.c:104)
```

2021-05-10
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 clearn-rebuild):

```bash
make \\  \\ -D CMAKE_BUILD_TYPE=Debug \\  \\ -D WITH_SERVER=ON \\  \\ -D WITH_STAGED_PLUGINS=ON \\  \\ -D WITH_PYSPATIALITE=ON \\  \\ -D ENABLE_TESTS=1 \\  \\ -D CMAKE_CXX_COMPILER:FILEPATH=/usr/lib/ccache/g++ \\  \\ -D WITH_ASTYLE=1 \\  \\ -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 deinitialization:

```
(gdb) bt
#0  0x0000000000000161 in ?? ()
#1  0x00007fadcac66577 in qgsConnectionPool_ConnectionDestroy (c=0x16fad60) at
  /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrconnpool.h:45
#2  0x00007fadcac67394 in QgsConnectionPoolGroup<QgsOgrConn>::~QgsConnectionPoolGroup (this=0x15fde30, __in_chrg=<optimized out>)
  at /usr/src/qgis/qgis-master/src/providers/ogr/../../core/qgsconnectionpool.h:77
#3  0x00007fadcac78bb4 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  0x00007fadcac78bf0 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
#5  0x00007fadcac76a4c in QgsConnectionPool<Product<QgsOgrConn*, QgsOgrConnPoolGroup>*>(::QgsOgrConnPool (this=0x7fadcae8b310 <QgsOgrConnPool::sInstance>, __in_chrg=<optimized out>)
  at /usr/src/qgis/qgis-master/src/providers/ogr/../../core/qgsconnectionpool.h:238
#6  0x00007fadcac766e5 in QgsOgrConnPool::~QgsOgrConnPool (this=0x7fadcae8b310 <QgsOgrConnPool::sInstance>, __in_chrg=<optimized out>)
```
at /usr/src/qgis/qgis-master/src/providers/ogr/qgsogrconnpool.cpp:31
    #7  0x0000000000000000 in __run_exit_handlers (status=0, listp=0x7fae007ce6c8 <__exit_funcs>, run_list_atexit=run_list_atexit@entry=true)  at exit.c:82
    #8  0x0000000000000000 in __GI_exit (status=<optimized out>) at exit.c:104
    #9  0x0000000000000000 in __libc_start_main (main=0x406440 <main(int, char**)>, argc=1, argv=0x7fff95784758, init=<optimized out>,
        fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fff95784748) at libc-start.c:321
#10 0x0000000000000000 in _start ()

Compiler is g++ (Ubuntu 4.8.4-2ubuntu1~14.04) 4.8.4

#6 - 2016-02-02 12:42 AM - Sandro Santilli

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

#7 - 2016-02-02 12:46 AM - Sandro Santilli

- Status changed from Feedback to In Progress
- Pull Request or Patch supplied changed from No to Yes
- % Done changed from 0 to 90
- Target version set to Version 2.14
- Assignee set to Sandro Santilli
- Category set to Data Provider/OGR

#8 - 2016-02-02 01:04 AM - Nyall Dawson

Nice catch.

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

#9 - 2016-02-02 01:39 AM - Sandro Santilli

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

#10 - 2016-02-02 08:25 AM - Sandro Santilli

- Status changed from In Progress to Closed

Fixed in changeset commit:"051253888810b06f6b055bfe457a7c6a009e3fc8".

2021-05-10