

QGIS Application - Bug report #14269
PyQgsPostgresProvider test hangs in absence of test database

2016-02-10 06:54 AM - Sandro Santilli

Status:	Closed	
Priority:	Normal	
Assignee:	Jürgen Fischer	
Category:	Tests Suite	
Affected QGIS version:	master	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:		Resolution: fixed/implemented
Crashes QGIS or corrupts data:		Copied to github as #: 22264
Description		
Running		
ctest -R PyQgsPostgresProvider		
after configuring build with -DENABLE_PGTEST=ONhangs when reaching this point:		
Start 191: PyQgsPostgresProvider		
It just sits there indefinitely and cannot be put in background.		
Note that I hadn't created a qgis_test database (as recommended in tests/README.md		
See also #14268		

Associated revisions

Revision 42bc9387 - 2016-02-16 10:45 AM - Jürgen Fischer

add QgsCredentialsNone to not return any credentials and make it default instead of QgsCredentialsConsole (fixes #14269; reverts f8d2091)

History

#1 - 2016-02-10 07:32 AM - Sandro Santilli

The hang is due to user prompt:

```
#0 000007ff8c2bed82d in read () at ../sysdeps/unix/syscall-template.S:81
#1 0x00007ff8c2b7c6a0 in _IO_new_file_underflow (fp=0x7ff8c2ec1640 <_IO_2_1_stdin_>) at fileops.c:613
#2 0x00007ff8c2b7d62e in __GI__IO_default_uflow (fp=0x7ff8c2ec1640 <_IO_2_1_stdin_>) at genops.c:435
#3 0x00007ff8c2b73bae in _IO_getc (fp=0x7ff8c2ec1640 <_IO_2_1_stdin_>) atgetc.c:39
#4 0x00007ff8bf4041c7 in ?? () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#5 0x00007ff8bf3b444a in QFile::readData(char*, long long) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#6 0x00007ff8bf3bb3b5 in QIODevice::read(char*, long long) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#7 0x00007ff8bf3cb80b in ?? () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#8 0x00007ff8bf3cc1bc in ?? () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#9 0x00007ff8bf3d0e18 in QTextStream::operator>>(QString&) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#10 0x00007ff8bfd16470 in QgsCredentialsConsole::request (this=0x18252b0, realm=..., username=..., password=..., message=...)
```

```
at /usr/src/qgis/qgis-master/src/core/qgscredentials.cpp:118
#11 0x00007ff8bfd15ef9 in QgsCredentials::get (this=0x18252c0, realm=..., username=..., password=..., message=...)
at /usr/src/qgis/qgis-master/src/core/qgscredentials.cpp:62
#12 0x00007ff881a3f470 in QgsPostgresConn::QgsPostgresConn (this=0x1825380, conninfo=..., readOnly=true, shared=true, transaction=false)
at /usr/src/qgis/qgis-master/src/providers/postgres/qgspostgresconn.cpp:241
```

Nyall, ideas ?

#2 - 2016-02-10 07:36 AM - Sandro Santilli

For batch-processes (like in this case) we should avoid ever prompting the user.

The psql script has a switch for this, don't know how it translates down to libpq and up to whatever the python test uses:

```
-w, --no-password
```

Never issue a password prompt. If the server requires password authentication and a password is not available by other means such as a .pgpass file, the connection attempt will fail. This option can be useful in batch jobs and scripts where no user is present to enter a password.

Note that this option will remain set for the entire session, and so it affects uses of the meta-command \connect as well as the initial connection attempt.

#3 - 2016-02-10 08:00 AM - Sandro Santilli

Creating the database still hangs. What fixes the hang is tweaking the "dbconn" to only specify database name:

```
- cls.dbconn = u'dbname=\\'qgis_test\\' host=localhost port=5432 user=\\'postgres\\' password=\\'postgres\\'
+ cls.dbconn = u'dbname=\\'qgis_test\\'
```

Then I can configure all parameters via env variables, which works fine.

What would you think about such change Matthias ?

#4 - 2016-02-10 08:07 AM - Sandro Santilli

A PR to avoid hard-coding the connection parameters is here:

<https://github.com/qgis/QGIS/pull/2779>

#5 - 2016-02-15 09:50 AM - Sandro Santilli

I found another reason for hanging: database being down. In that case, running ctest with -V shows:

```
195: User DB Path: /usr/src/qgis/qgis-master/resources/qgis.db
195: Auth DB Path: /home/strk/.qgis2/qgis-auth.db
195:
195: credentials for dbname='qgis_test' sslmode=disable
195: message: could not connect to server: No such file or directory
```

195: Is the server running locally and accepting
195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:

And then it hangs there forever, rather than reporting the error and exiting.

#6 - 2016-02-15 09:56 AM - Sandro Santilli

The connection parameters default have been pushed with commit:ab580fbd0e0f238160b2cdee3de47b9a412eb6e5 but the new blockage (on unavailable database) is due to the following line never returning in setUpClass class method of TestPyQgsPostgresProvider:

```
cls.vl = QgsVectorLayer(cls.dbconn + ' sslmode=disable key=\\'pk\\' srid=4326 type=POINT table="qgis_test"."someData" (geom) sql=', 'test',  
'postgres')
```

#7 - 2016-02-15 10:58 AM - Sandro Santilli

To reproduce the blockage:

1. Shut down the database server
2. Run ctest -V -R Postgres

Note that adding self.assertEqual(1, 2) *before* the QgsVectorLayer call in setUpClass makes the test correctly exit with an error while adding it *after* it never reaches it.

#8 - 2016-02-15 11:04 AM - Sandro Santilli

I suspect this has to do with the new authentication mechanism, can it be ? Like, qgis trying to prompt user via host provided callback which isn't provided during testing ? Nyall, can it be ?

#9 - 2016-02-15 11:09 AM - Sandro Santilli

strace shows the process is looping here:

```
select(9, [4 6 8], NULL, NULL, {0, 87262}) = 0 (Timeout)
```

#10 - 2016-02-15 12:41 PM - Sandro Santilli

gdb backtrace of the select situation:

```
(gdb) bt  
#0 0x00007fa5c2095d83 in __select_nocancel () at ../sysdeps/unix/syscall-template.S:81  
#1 0x000000000078ec72 in cmsysProcess_WaitForData ()  
#2 0x0000000000574ac8 in cmProcess::GetNextOutputLine(std::string&, double) ()  
#3 0x000000000056948c in cmCTestRunTest::CheckOutput() ()
```

#4 0x00000000055f028 in cmCTestMultiProcessHandler::CheckOutput() ()
#5 0x000000000564778 in cmCTestMultiProcessHandler::RunTests() ()

#11 - 2016-02-15 12:51 PM - Sandro Santilli

The following debug lines are printed, do they ring any bell ?

195: src/core/qgsvectorlayer.cpp: 584: (geometryType) [0ms] invalid layer or pointer to mDataProvider is null
195: src/core/qgsvectorlayer.cpp: 592: (geometryType) [0ms] WARNING: This code should never be reached. Problems may occur...

#12 - 2016-02-16 12:57 AM - Sandro Santilli

- Assignee changed from Matthias Kuhn to Larry Shaffer

Larry, assigning to you as a trick to mention you here. Instructions are needed about how to avoid interactive credentials fetching for batch processing (testsuite being one case)

#13 - 2016-02-16 01:05 AM - Sandro Santilli

- Assignee changed from Larry Shaffer to Jürgen Fischer

As of commit:f8d2091255316aca11ae40eb0f6273bb2003cf5a (Thanks Juergen) the test exists with a failure after a few attempts:

195: credentials for dbname='qgis_test' sslmode=disable
195: message: could not connect to server: No such file or directory
195: Is the server running locally and accepting
195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:
195: username: password: credentials for dbname='qgis_test' sslmode=disable
195: message: could not connect to server: No such file or directory
195: Is the server running locally and accepting
195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:
195: src/core/qgscredentials.cpp: 64: (get) [1ms] requested realm:dbname='qgis_test' sslmode=disable username: password:
195: src/providers/postgres/qgspostgresconn.cpp: 253: (QgsPostgresConn) [0ms] Connecting to dbname='qgis_test' sslmode=disable
195: src/core/qgscredentials.cpp: 64: (get) [0ms] requested realm:dbname='qgis_test' sslmode=disable username: password:
195: src/providers/postgres/qgspostgresconn.cpp: 253: (QgsPostgresConn) [0ms] Connecting to dbname='qgis_test' sslmode=disable
195: username: password: credentials for dbname='qgis_test' sslmode=disable
195: message: could not connect to server: No such file or directory
195: Is the server running locally and accepting
195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:
195: src/core/qgscredentials.cpp: 64: (get) [0ms] requested realm:dbname='qgis_test' sslmode=disable username: password:
195: src/providers/postgres/qgspostgresconn.cpp: 253: (QgsPostgresConn) [0ms] Connecting to dbname='qgis_test' sslmode=disable
195: username: password: credentials for dbname='qgis_test' sslmode=disable
195: message: could not connect to server: No such file or directory
195: Is the server running locally and accepting

```

195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:
195: src/core/qgscredentials.cpp: 64: (get) [0ms] requested realm:dbname='qgis_test' sslmode=disable username: password:
195: src/providers/postgres/qgspostgresconn.cpp: 253: (QgsPostgresConn) [0ms] Connecting to dbname='qgis_test' sslmode=disable
195: username: password: credentials for dbname='qgis_test' sslmode=disable
195: message: could not connect to server: No such file or directory
195: Is the server running locally and accepting
195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:
195: src/core/qgscredentials.cpp: 64: (get) [0ms] requested realm:dbname='qgis_test' sslmode=disable username: password:
195: src/providers/postgres/qgspostgresconn.cpp: 253: (QgsPostgresConn) [0ms] Connecting to dbname='qgis_test' sslmode=disable
195: src/core/qgsmessagelog.cpp: 45: (logMessage) [0ms] 2016-02-16T10:01:46 PostGIS[1] Connection to database failed
195: could not connect to server: No such file or directory
195: Is the server running locally and accepting
195: connections on Unix domain socket "/tmp/.s.PGSQL.5493"?
195:
195: src/core/qgsproviderregistry.cpp: 402: (provider) [0ms] Instantiated the data provider plugin: postgres
195: src/core/qgsvectorlayer.cpp: 1534: (setDataProvider) [0ms] Instantiated the data provider plugin
195: src/core/qgsvectorlayer.cpp: 1539: (setDataProvider) [0ms] Invalid provider plugin dbname='qgis_test' sslmode=disable key='pk' srid=4326
type=POINT table="qgis_test"."someData" (geom) sql=
195: E
195: =====
195: ERROR: setUpClass (__main__.TestPyQgsPostgresProvider)
195: -----
195: Traceback (most recent call last):
195: File "/usr/src/qgis/qgis-master/tests/src/python/test_provider_postgres.py", line 43, in setUpClass
195:     assert(cls.vl.isValid())
195: AssertionError
195:
195: -----
195: Ran 0 tests in 0.003s
195:
195: FAILED (errors=1)
195: src/core/layertree/qgslayertreeregistrybridge.cpp: 78: (layersWillBeRemoved) [9ms] 0 layers will be removed, enabled:1
195: src/providers/postgres/qgspostgresconnpool.cpp: 33: (~QgsPostgresConnPool) [0ms] Entering.
195: src/providers/postgres/qgspostgresconnpool.cpp: 33: (~QgsPostgresConnPool) [1ms] Leaving.
195: CMake Error at /usr/src/qgis/build/master/tests/src/python/PyQgsPostgresProvider.cmake:22 (MESSAGE):
195: Test failed: 1
195:
195:
195: username: password:
1/1 Test #195: PyQgsPostgresProvider .....***Failed 0.36 sec

```

Jurgen: despite the commit log (avoid credentials requests in PyQgsPostgresProvider) the log still shows traces of QgsCredentials::get being called (which calls QgsCredentials::request) -- expected ?

#14 - 2016-02-16 01:06 AM - Sandro Santilli

- Resolution set to fixed/implemented
- Status changed from Open to Closed

in any case, this is now fixed.

#15 - 2016-02-24 11:15 AM - Larry Shaffer

Hi Sandro,

As far as the request for a 'master password' for the authentication system, one can predefine it using the environment variable QGIS_AUTH_PASSWORD_FILE. This assumes there is already one defined in the SQLite auth database. Generally, though, if there is not an authentication configuration used, no master password dialog would ever be shown. So, I think Jürgen's fix here addresses the source of the issue.