QGIS Application - Bug report #17866
Cannot load WMS capabilities from WMS provider in QGIS3 from standalone python

2018-01-16 10:05 AM - David Marteau

Status:	Closed		
Priority:	High		
Assignee:			
Category:	Data Provider		
Affected QGIS version:	master	**Regression?:**	Yes
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	no timely feedback
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	25761

Description

When trying to load projects with wms or wcs layer defined from python: loading capabilities fail **the first time** with timeout error, thus preventing creating any project or layer from python.

The problem has been verified in Debian, Ubuntu and fresh OSX build from master branch.

The effect is that you cannot instanciate properly a QgsProject with a WMS layer.

This problem has been verified on Debian stretch, Ubuntu Xenial, OXS 10.11 from fresh master.

This can be tested by simply instanciate a QgsProject in python (in a standalone python script) from a .qgs project having a single wms layer: it will fail to instanciate the layer.

We have somehow been able to reduce the problem to the loading of the wmsprovider. The following python code provide a minimal example for reproducing the problem:

Note that when you run this script, clear the Qgis network cache between two invocations.

```
import os
import logging

from qgis.core import QgsApplication, QgsMessageLog
from qgis.core import QgsProject, QgsProviderRegistry

logger = logging.getLogger()
logger.setLevel(logging.DEBUG)

# No Display
os.environ['QT_QPA_PLATFORM'] = 'offscreen'

qgis_application = QgsApplication([], False )
qgis_application.setPrefixPath('/usr/local', True)
qgis_application.initQgis()

# Log any messages
def writelogmessage(message, tag, level):
    arg = '{}: {}'.format( tag, message )
    if level == QgsMessageLog.WARNING:
        logger.warning(arg)
    elif level == QgsMessageLog.CRITICAL:
```

```
        logger.error(arg)
    else:
        logger.info(arg)

messageLog = QgsApplication.messageLog()
messageLog.messageReceived.connect( writelogmessage )

wmsuri = ("contextualWMSLegend=0&crs=EPSG:4326&dpiMode=7&featureCount=10&format=image/jpeg"
          "&layers=s2cloudless&styles&amp;tileMatrixSet=s2cloudless-wmsc-14"
          "&url=http://tiles.maps.eox.at/wms?" )

# GetCapabilities Fail with Request Timeout !!!
provider = QgsProviderRegistry.instance().createProvider( "wms", wmsuri )

# Run again => works !!!
provider = QgsProviderRegistry.instance().createProvider( "wms", wmsuri )

qgis_application.exitQgis()
del qgis_application
```

On failure, the logs will show the following errors

```
WMS: Download of capabilities failed: Operation canceled
Network: Network request https://tiles.maps.eox.at/wms?SERVICE=WMS&REQUEST=GetCapabilities timed out
```

After some investigation we have found that:

- Sending a request from python directly with the QgsNetworkAccessManager and QEventLoop works as expected.
- Calling from python a simple C++ method that do the same call fail: the QgsNetworkAccessManager never a send a finished signal from a successfully sent request
- When tracing the c++ execution we have checked that the event loop has the running status.

History

#1 - 2018-01-16 05:54 PM - Alessandro Pasotti

- File test_provider_wms.py added

- Status changed from Open to Feedback

- Assignee set to Alessandro Pasotti

I suspect that a main event loop is required for a proper execution of the signals and slots in the network access manager, please have a look to the attached working test case.

But I still don't know why :(

#2 - 2018-01-16 06:11 PM - David Marteau

I do not think this is totally right since after the first attempt (which fail) then things appears to be ok: IHMO there is some kind of init state condition concerning event loop which is not set on first run.

Anyway, runnig app.exec will defeat the purpose of embedding since it introduce a blocking state.

#3 - 2018-01-16 06:44 PM - Alessandro Pasotti

- Assignee deleted (Alessandro Pasotti)

Well, I don't really know what you're trying to do, but in QGIS application you do have a running main event loop and this is not an issue.

If you have a standalone non-GUI script that must run without an event loop you need a blocking condition or your script might terminate before the network reply comes back.

In any case you need an event loop (main or local).

Btw, in theory this whole thing should work without a main event loop because in the wms capabilities code there is a local event loop running, whose purpose is exactly to turn an asynchronous operation into a blocking one, what I don't understand is why it works the second time, I believe that the answer is in Qt QNetworkAccessManager core code, I don't see any obvious issue in QgsNetworkAccessManager or in the WMS code.

Just curious, was it working in 2.x ?

#4 - 2018-01-16 09:07 PM - David Marteau

Alessandro Pasotti wrote:

> *Well, I don't really know what you're trying to do, but in QGIS application you do have a running main event loop and this is not an issue.*

This is true for desktop. This is not true for server: in the fcgi implementation the main loop is a fcgi loop which is not related
to the QGIS application loop.

> *If you have a standalone non-GUI script that must run without an event loop you need a blocking condition or your script might terminate before the network reply comes back.*

Yes

> *In any case you need an event loop (main or local).*

Yes

> *Btw, in theory this whole thing should work without a main event loop because in the wms capabilities code there is a local event loop running, whose purpose is exactly to turn an asynchronous operation into a blocking one*

Exactly

> *, what I don't understand is why it works the second time,*

In fact, this is the way it should work in the normal way: as you said it, because of the the local event, it should not block the first time.

> *I believe that the answer is in Qt QNetworkAccessManager core code, I don't see any obvious issue in QgsNetworkAccessManager or in > the WMS code.*

Me neither, I have spent some time to check the c++ code and I did not see anything suspicious.
Curiously, when I wrote a test case in python directly with the QNetworkAccessManager (and a local QEventLoop) to mimic what's in the c++ code then it work flawlessly.

> *Just curious, was it working in 2.x ?*

This has to be checked, and it has to be checked also in the fcgi implementation of the Qgis3 server.

#5 - 2018-01-17 09:48 AM - Alessandro Pasotti

- File testqgswmsprovider17866.cpp added

I've translated the (failing) python test to C++ and it passes ... that's wierd.

```
# This is failing
def testWmsGetCapabilitiesLayer(self):
    self.layer = QgsRasterLayer(self.wmsuri, 'mywms', 'wms')
    self.assertTrue(self.layer.isValid())


// This is passing:
QgsRasterLayer layer( QStringLiteral( "contextualWMSLegend=0&crs=EPSG:4326&dpiMode=7&featureCount=10&format=image/jpeg"
                                      "&layers=s2cloudless&styles&amp;tileMatrixSet=s2cloudless-wmsc-14"
                                      "&url=http://tiles.maps.eox.at/wms?" ),
                      QStringLiteral( "mywms" ),
                      QStringLiteral( "wms" ));
QVERIFY(layer.isValid());
```

That's pointing us towards the initialization code in QGIS application: initQGIS doesn't do all the stuff...

#6 - 2018-01-17 03:48 PM - Alessandro Pasotti

- Regression? changed from No to Yes

The test works fine in 2.x so it's a regression

#7 - 2018-01-17 03:48 PM - Alessandro Pasotti

- Priority changed from Normal to High

#8 - 2018-01-17 06:11 PM - Alessandro Pasotti

I did some more research with different endpoints and I've discovered something really confusing, I now believe that we hit a Qt bug.
Here is my evidence:
I've tested several endpoints (qgis-server, geoserver and remote wms), this is the summary:

- qgis server on http://127.0.0.1 works
- qgis server (same as above) on http://localhost fails
- geoserver on a local docker with defined hostname fails
- geoserver on a local docker (same as above) with the docker IP works
- any remote hostname fails
- any remote IP address works

For example (same server):

```
    fails =
'contextualWMSLegend=0&crs=EPSG:4326&dpiMode=7&featureCount=10&format=image/png&layers=track_1&styles=&url=https://www.movimentolento.it/re
entolento.it/resource/wms/'
    works =
'contextualWMSLegend=0&crs=EPSG:4326&dpiMode=7&featureCount=10&format=image/png&layers=track_1&styles=&url=http://85.94.200.131/resource/wr
1/resource/wms/'
```

So, we can assume that there is something going bad with hostname resolution, every time I use the IP address it works and it fails when using the hostname (of course I verified with a browser that all the getcapabilities URLs were working).

#9 - 2018-01-18 11:33 AM - Alessandro Pasotti

Further info: a DNS look up prior to the connection works (Note: Since Qt 4.6.3 QHostInfo is using a small internal 60 second DNS cache for performance improvements.)

```
def testLookup(self):
    self.loop = QEventLoop()
    def _p(info):
        self.loop.quit()
        self.assertTrue(info.addresses()[0].toString() == '85.94.200.130')
    QHostInfo.lookupHost('www.movimentolento.it', _p)
    self.loop.exec_()
    del(self.loop)
    # Without the DNS lookup the following fails
    self.layer = QgsRasterLayer(self.wmsuri_remote_ml_hostname, 'mywms', 'wms')
    self.assertTrue(self.layer.isValid())
```

#10 - 2018-01-18 11:54 AM - David Marteau

Allessandro, thanks for investigating this.

So you think that it is related to DNS ?

For information, In my tests, wathever the url given in the first call but all subsequent call on différent hostname were successful: I have tested this by first calling with a 'localhost' - which has no wms service at all, but fail with timeout, then doing other calls to - say 'tiles.maps.eox.at' - are all successfull.

So could we have a scenario where the DNS resolver is not correctly available/initialized for network opération at the first call and available for all subsequent calls ?

#11 - 2018-01-18 12:36 PM - Alessandro Pasotti

> *So you think that it is related to DNS ?*

I suspect it is related with QHostInfo::lookupHost() which is called by QAbstractSocketPrivate, there are a lot of signal-slot connections that being in a separate thread are probably bound to the GUI thread loop (which isn't running in our test cases) for a correct dispatch and delivery. Still this doesn't explain why subsequent calls on different hostnames works after the first fails.

> *For information, In my tests, wathever the url given in the first call but all subsequent call on différent hostname were successful: I have tested this by first calling with a 'localhost' - which has no wms service at all, but fail with timeout, then doing other calls to - say 'tiles.maps.eox.at' - are all successfull.*
> *So could we have a scenario where the DNS resolver is not correctly available/initialized for network opération at the first call and available for all subsequent calls ?*

I admit that I don't have a clear picture but yes, I think that DNS lookup initialization might be involved.

Would you please confirm that using an IP address instead of an hostname works?

#12 - 2018-01-18 02:26 PM - Alessandro Pasotti

I understand that this is just a workaround, but calling

```
QHostInfo.fromName('') # Yes: it also works whith empty/invalid URL
```

initialize the system and then it works as expected, internally it dynamically load a bounch of system libraries (on unix).

#13 - 2018-01-24 09:59 AM - Alessandro Pasotti

- Priority changed from High to Normal

Lowering priority because we have a workaround.

#14 - 2018-02-22 12:03 PM - Giovanni Manghi

- Operating System deleted (All)
- Status changed from Feedback to Open
- Priority changed from Normal to High

Making it "high" as is a regression.

#15 - 2018-12-28 12:19 PM - Paolo Cavallini

Could you please check again on current release?
Thanks.

#16 - 2018-12-28 02:58 PM - Giovanni Manghi

- Status changed from Open to Feedback

Paolo Cavallini wrote:

> *Could you please check again on current release?*
> *Thanks.*

Please change status to "feedback" when needed.

#17 - 2019-01-14 11:57 PM - Nyall Dawson

- Status changed from Feedback to Closed

- Resolution set to no timely feedback

Files

test_provider_wms.py	1.85 KB	2018-01-16	Alessandro Pasotti
testqgswmsprovider17866.cpp	2.03 KB	2018-01-17	Alessandro Pasotti