QGIS Application - Bug report #14822
QGIS Server 2.14.2: segmentation fault in GetProjectSettings

2016-05-16 02:34 AM - luca76 -

Status:	Closed		
Priority:	High		
Assignee:			
Category:	QGIS Server		
Affected QGIS version:	2.14.2	**Regression?:**	Yes
Operating System:	Debian	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	not reproducable
Crashes QGIS or corrupts data:	Yes	**Copied to github as #:**	22777

Description

occasionally I have problems with QGIS Server 2.14.0-2 (but not in 2.8.x).

When I switch from a project to another from QGIS Web Client, when it calls the qgis server (WMS GetProjectSettings) I have a segmentation fault as you can see in the below GDB backtrace. It seems that it's a problem in QgsServerProjectParser that didn't exist in 2.8.x.

This problem occurs when I switch from a big project to another, with inherited layers from other projects.

```
Program received signal SIGSEGV, Segmentation fault.
0xb6f75ba1 in QBasicAtomicInt::ref (this=0x29) at
/usr/include/qt4/QtCore/qatomic_i386.h:120
120                 : "memory");

(gdb) bt

#0  0xb6f75ba1 in QBasicAtomicInt::ref (this=0x29) at
/usr/include/qt4/QtCore/qatomic_i386.h:120
#1  0xb6f77b74 in QString::QString (this=0x9c17c44, other=...) at
/usr/include/qt4/QtCore/qstring.h:726
#2  0xb722d753 in QgsMapLayer::id (this=0xcc483d8) at
/usr/local/src/qgis-2.14.2/src/core/qgsmaplayer.cpp:107
#3  0xb7088c81 in QgsLayerTreeLayer::QgsLayerTreeLayer
(this=0x9c17c28, layer=0xcc483d8)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreelayer.cpp:27
#4  0xb708932a in QgsLayerTreeLayer::readXML (element=...)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreelayer.cpp:111
#5  0xb709bf49 in QgsLayerTreeNode::readXML (element=...)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreenode.cpp:56
#6  0xb7086a81 in QgsLayerTreeGroup::readChildrenFromXML
(this=0x9874e18, element=...)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreegroup.cpp:293
#7  0xb70864bc in QgsLayerTreeGroup::readXML (element=...)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreegroup.cpp:259
#8  0xb709befc in QgsLayerTreeNode::readXML (element=...)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreenode.cpp:54
#9  0xb7086a81 in QgsLayerTreeGroup::readChildrenFromXML
(this=0x99d0320, element=...)
```

```
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreegroup.cpp:293
#10 0xb70864bc in QgsLayerTreeGroup::readXML (element=...)
    at /usr/local/src/qgis-2.14.2/src/core/layertree/qgslayertreegroup.cpp:259
#11 0x080f9cc2 in QgsServerProjectParser::findLegendGroupElements
(this=0xcbf39e8)
    at /usr/local/src/qgis-2.14.2/src/server/qgsserverprojectparser.cpp:1209
#12 0x080efacb in QgsServerProjectParser::QgsServerProjectParser
(this=0xcbf39e8, xmlDoc=0x8b5b138, filePath=...)
    at /usr/local/src/qgis-2.14.2/src/server/qgsserverprojectparser.cpp:59
#13 0x080603e4 in QgsConfigCache::serverConfiguration (this=0x8b7dad0,
filePath=...)
    at /usr/local/src/qgis-2.14.2/src/server/qgsconfigcache.cpp:55
#14 0x080f114c in QgsServerProjectParser::createLayerFromElement
(this=0xcbb01c0, elem=..., useCache=true)
    at /usr/local/src/qgis-2.14.2/src/server/qgsserverprojectparser.cpp:256
#15 0x080f01b2 in QgsServerProjectParser::projectLayerMap
(this=0xcbb01c0, layerMap=...)
    at /usr/local/src/qgis-2.14.2/src/server/qgsserverprojectparser.cpp:102
#16 0x080fadef in QgsServerProjectParser::wfsLayerNames (this=0xcbb01c0)
    at /usr/local/src/qgis-2.14.2/src/server/qgsserverprojectparser.cpp:1319
#17 0x080d79eb in QgsWMSProjectParser::wfsLayerNames (this=0x9d7ccc0)
#18 0x0807cec8 in QgsWMSServer::getCapabilities (
    this=<error reading variable: Cannot access memory at address 0xbfd6d5e4>,
    this@entry=<error reading variable: Cannot access memory at
address 0xbfd6d5dc>,
    version=<error reading variable: Cannot access memory at address
0xbfd6d5e8>,
    fullProjectInformation=<error reading variable: Cannot access
memory at address 0xbfd6d31c>)
    at /usr/local/src/qgis-2.14.2/src/server/qgswmsserver.cpp:589
```

Associated revisions

Revision 231f6af0 - 2017-01-14 06:53 AM - Nyall Dawson

Don't return const references to implicitly shared Qt classes

Instead return the inexpensive copies. Should provide extra safety
against issues like #14822

(refs #14822)

History

#1 - 2016-06-10 08:47 AM - Sandro Santilli

Luca what do you mean by switching from a project to another ? Could you provide a script to reproduce the crash ?

#2 - 2016-06-20 09:40 AM - Jürgen Fischer

- Status changed from Open to Feedback

#3 - 2016-06-21 06:04 AM - Giovanni Manghi

Sandro Santilli wrote:

> *Luca what do you mean by switching from a project to another ? Could you provide a script to reproduce the crash ?*

QGIS web client has "project switcher" functionality, I think that he refers to that.

#4 - 2016-06-21 06:05 AM - Giovanni Manghi

- Status changed from Feedback to Open

#5 - 2016-06-22 05:48 AM - michele zanolli

Giovanni Manghi wrote:

> *QGIS web client has "project switcher" functionality, I think that he refers to that.*

Exactly. When you change project with "project switcher", Qgis-Web-Client opens a GetProjectSettings request. If we do more GetProjectSettings requests on different projects (two or three times), the memory on the server grows and then we have a segmentation fault.
So we can suppose that the bug could be a memory leak problem.

Pay attention that we have projects that embed layers from other projects.

#6 - 2016-06-28 03:44 AM - Martin Dobias

- Status changed from Open to Feedback

Could you please verify that the problem is caused by memory usage on server growing and getting completely exhausted in the end, causing that crash?

Do embedded layers need to be involved in order to get the crash?

#7 - 2016-06-28 03:50 AM - luca76 -

Martin Dobias wrote:

> *Could you please verify that the problem is caused by memory usage on server growing and getting completely exhausted in the end, causing that crash?*

No, The QGIS Server memory around 250-300 MB and then crashes. System has 2GB of RAM.

> *Do embedded layers need to be involved in order to get the crash*

It's only an hypothesis. I'll do further tests about this.

#8 - 2016-06-30 11:22 PM - Andreas Neumann

Hi Luca,

I think the devs really need more information or access to your server to examine this case.

At my last workplace in Uster there are a lot of different projects on the same QGIS server machine. At the public server there are 28 projects, in the internal one there are some more, maybe 40. Maybe half of them have embedded layers. We did not experience the issues you describe.

That doesn't mean your problem does not exist with your particular version. But just to say that is not a general, widely known issue. Otherwise, more users would complain. Meanwhile there are quite a lot of QGIS server users around, and I guess that many of them have more than one project.

However, I would assume that not so many users/organizations use embedded layers. So maybe it is an issue around the embedded layers.

#9 - 2016-07-03 02:58 PM - Matthias Kuhn

Hi Luca,

I have a strong suspicion about what's going on. Are you able to run a self-compiled server there to get a bit of missing information?

#10 - 2016-07-03 11:54 PM - luca76 -

Matthias Kuhn wrote:

> *I have a strong suspicion about what's going on. Are you able to run a self-compiled server there to get a bit of missing information?*

First, we used the "official" debian packages from qgis.org and they generate the same error. Then we tested it with a DEBUG enabled compiled source code of qgis.

#11 - 2016-07-04 12:54 AM - Matthias Kuhn

> *Then we tested it with a DEBUG enabled compiled source code of qgis.*

So you are able to run your custom built sources, good :) Could you check with the latest master version (from this morning) and look for this debug message

"Map layer deleted without unregistering!"

#12 - 2016-07-05 08:14 AM - luca76 -

Matthias Kuhn wrote:

> *So you are able to run your custom built sources, good :) Could you check with the latest master version (from this morning) and look for this debug message*
>
> *"Map layer deleted without unregistering!"*

We compiled the last master version and all problems are gone! We are looking forward to test the 2.14.4 version what will be out this friday (according to the qgis.org homepage).

#13 - 2016-07-08 02:53 AM - michele zanolli

Thanks Matthias,

As Luca says, with the latest master version we do not have segmentation fault problems.
We do not see any kind of "Map layer deleted without unregistering!" messages in the qgis server log.

Is this fix backported to the upcoming LTR 2.14.4?

#14 - 2016-07-09 07:27 AM - Matthias Kuhn

Thanks for the feedback.

I'm not sure what exactly ends up in the server logs. It's a debug message. I think to get them you need a build with debug enabled and set some environment variables (https://lists.osgeo.org/pipermail/qgis-user/2014-June/028066.html)

If this solved the crashes there were layers deleted unexpectedly. I don't know why that happens and if these layers are now missing from your maps. If you find layers to be suddenly missing, please check the debug logs and open an issue with a link to this one.

#15 - 2016-07-09 07:27 AM - Matthias Kuhn

- Status changed from Feedback to Closed

#16 - 2016-07-11 01:47 AM - michele zanolli

- Status changed from Closed to Reopened

We did try with the latest 2.14.4, compiled with Debug active.

Here the gdb dump:

```
Program received signal SIGSEGV, Segmentation fault.
0xb5217abb in QString::append(QString const&) () from /usr/lib/i386-linux-gnu/libQtCore.so.4
(gdb) bt
#0  0xb5217abb in QString::append(QString const&) () from /usr/lib/i386-linux-gnu/libQtCore.so.4
#1  0xb6ee7814 in QString::operator+= (this=0xbfe7174c, s=...) at /usr/include/qt4/QtCore/qstring.h:274
#2  0xb6ee7928 in operator+ (s1=0xb74e483e "returning name '", s2=...) at /usr/include/qt4/QtCore/qstring.h:1031
#3  0xb719ab96 in QgsMapLayer::name (this=0xa859100) at /usr/local/src/qgis-2.14.4/src/core/qgsmaplayer.cpp:126
#4  0x080e08da in QgsWMSProjectParser::addLayers (this=0xa504a28, doc=..., parentLayer=..., legendElem=..., layerTreeGroup=0xbe6dff0,
    layerMap=..., nonIdentifiableLayers=..., version=..., fullProjectSettings=true, idNameMap=..., layerIDList=...)
    at /usr/local/src/qgis-2.14.4/src/server/qgswmsprojectparser.cpp:1252
#5  0x080e05b8 in QgsWMSProjectParser::addLayers (this=0xa504a28, doc=..., parentLayer=..., legendElem=..., layerTreeGroup=0xac24de8,
    layerMap=..., nonIdentifiableLayers=..., version=..., fullProjectSettings=true, idNameMap=..., layerIDList=...)
    at /usr/local/src/qgis-2.14.4/src/server/qgswmsprojectparser.cpp:1229
#6  0x080d5d26 in QgsWMSProjectParser::layersAndStylesCapabilities (this=0xa504a28, parentElement=..., doc=..., version=...,
    fullProjectSettings=true) at /usr/local/src/qgis-2.14.4/src/server/qgswmsprojectparser.cpp:130
#7  0x0807d237 in QgsWMSServer::getCapabilities (this=0xbfe72688, version=..., fullProjectInformation=true)
    at /usr/local/src/qgis-2.14.4/src/server/qgswmsserver.cpp:608
```

#8 0x080790c3 in QgsWMSServer::executeRequest (this=0xbfe72688) at /usr/local/src/qgis-2.14.4/src/server/qgswmsserver.cpp:174
#9 0x0810fb64 in QgsServer::handleRequest (this=0xbfe727d3, queryString=...)
    at /usr/local/src/qgis-2.14.4/src/server/qgsserver.cpp:628
#10 0x0805ea09 in main (argc=1, argv=0xbfe728a4) at /usr/local/src/qgis-2.14.4/src/server/qgis_map_serv.cpp:43

#17 - 2016-07-11 04:51 AM - luca76 -

Matthias Kuhn wrote:

> *Thanks for the feedback.*
>
> *I'm not sure what exactly ends up in the server logs. It's a debug message. I think to get them you need a build with debug enabled and set some environment variables (https://lists.osgeo.org/pipermail/qgis-user/2014-June/028066.html)*
>
> *If this solved the crashes there were layers deleted unexpectedly. I don't know why that happens and if these layers are now missing from your maps. If you find layers to be suddenly missing, please check the debug logs and open an issue with a link to this one.*

I've tested again with QGIS Server from master (2.15), debug messages enabled. I've set the QgsDebug*() to a log file, and I've never encountered the message "Map layer deleted without unregistering!".

Again, with the master version the server doesn't crash. So the bug could be in an another place.

#18 - 2016-07-11 05:04 AM - luca76 -

michele zanolli wrote:

> *We did try with the latest 2.14.4, compiled with Debug active.*
>
> *Here the gdb dump:*
>
> *> #3 0xb719ab96 in QgsMapLayer::name (this=0xa859100) at /usr/local/src/qgis-2.14.4/src/core/qgsmaplayer.cpp:126*

In the line 126 of the source code:
QgsDebugMsgLevel("returning name '" + mLayerName + '\\'', 4);

the mLayerName is NULL, this causes segmentation fault.

#19 - 2016-07-11 05:57 AM - Matthias Kuhn

mLayerName is not a pointer, it's a reference so it cannot be NULL.

The layer is deleted but a pointer to it is still in the map layer registry. So any access to one of its members causes a crash.

Basically, the stability fixes in 2.16 remove dead layers from the registry even if they are not properly unregistered. As a consequence, if someone tries to access the dead layer by id it will not receive an invalid pointer. But that leaves the question, how the layer got deleted and why something still wants to access the deleted layer.

Anyway, I will backport the fix in 2.16 to 2.14 and if you have any pointer on removed layers, or mentioned debug message, please open an issue (and ping me so I get a notification).

#20 - 2016-07-11 06:04 AM - luca76 -

Matthias Kuhn wrote:

> *Anyway, I will backport the fix in 2.16 to 2.14 and if you have any pointer on removed layers, or mentioned debug message, please open an issue (and ping me so I get a notification).*

Perfect, when you have backported the fix I'll try ASAP.

#21 - 2016-07-11 06:26 AM - Matthias Kuhn

Compiling a bunch of backports right now and will push right after if everything is alright

#22 - 2017-01-03 06:57 AM - Patrick Kirsch

- Assignee set to Matthias Kuhn

Was this issue resolved?
I tried following qgis versions on Debian GNU/Linux 8 (jessie):

- qgis-2.18.2+13jessie
- qgis-2.8.9
- qgis-2.14.0

I'm asking because I ran also probably in this issue.

This trace is from version qgis-2.16.3.

```
$:/usr/lib/cgi-bin# gdb -r qgis_mapserv.fcgi
GNU gdb (Debian 7.7.1+dfsg-5) 7.7.1
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from qgis_mapserv.fcgi...expanding to full symbols...done.
(gdb) set env QUERY_STRING =
SERVICE=WMS&VERSION=1.3&REQUEST=GetProjectSettings&map=/home/web/qgis-02-web-client/projects/immobilienverwaltung_fehler.qgs
(gdb) r
```

```
Starting program: /usr/lib/cgi-bin/qgis_mapserv.fcgi
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
QFSFileEngine::open: No file name specified
[New Thread 0x7fffd4323700 (LWP 7437)]
[New Thread 0x7fffc9b0f700 (LWP 7438)]
[New Thread 0x7fffc930e700 (LWP 7439)]
[New Thread 0x7fffc8b0d700 (LWP 7440)]
[New Thread 0x7fffbb6f8700 (LWP 7441)]
[New Thread 0x7fffbaef7700 (LWP 7442)]
ERROR 6: EPSG PCS/GCS code 900913 not found in EPSG support files.  Is this a valid
EPSG coordinate system?
ERROR 6: EPSG PCS/GCS code 900913 not found in EPSG support files.  Is this a valid
EPSG coordinate system?

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff3e2e029 in qHash(QString const&) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
(gdb) bt
#0  0x00007ffff3e2e029 in qHash(QString const&) () from /usr/lib/x86_64-linux-gnu/libQtCore.so.4
#1  0x00000000004989c8 in QHash<QString, QHashDummyValue>::findNode (this=0x430bbd0, akey=..., ahp=0x0) at
/usr/include/qt4/QtCore/qhash.h:882
#2  0x0000000000496264 in QHash<QString, QHashDummyValue>::contains (this=0x430bbd0, akey=...) at /usr/include/qt4/QtCore/qhash.h:874
#3  0x0000000000493027 in QSet<QString>::contains (this=0x430bbd0, value=...) at /usr/include/qt4/QtCore/qset.h:91
#4  0x00000000004e0ad1 in QgsWMSProjectParser::addLayers (this=0x430af40, doc=..., parentLayer=..., legendElem=...,
layerTreeGroup=0x461c420, layerMap=..., nonIdentifiableLayers=..., version=..., fullProjectSettings=true,
    idNameMap=..., layerIDList=...) at /source/qgis-2.16.3/src/server/qgswmsprojectparser.cpp:1262
#5  0x00000000004e06b0 in QgsWMSProjectParser::addLayers (this=0x430af40, doc=..., parentLayer=..., legendElem=...,
layerTreeGroup=0x461baa0, layerMap=..., nonIdentifiableLayers=..., version=..., fullProjectSettings=true,
    idNameMap=..., layerIDList=...) at /source/qgis-2.16.3/src/server/qgswmsprojectparser.cpp:1234
#6  0x00000000004e06b0 in QgsWMSProjectParser::addLayers (this=0x430af40, doc=..., parentLayer=..., legendElem=...,
layerTreeGroup=0x4718390, layerMap=..., nonIdentifiableLayers=..., version=..., fullProjectSettings=true,
    idNameMap=..., layerIDList=...) at /source/qgis-2.16.3/src/server/qgswmsprojectparser.cpp:1234
#7  0x00000000004d5d60 in QgsWMSProjectParser::layersAndStylesCapabilities (this=0x430af40, parentElement=..., doc=..., version=...,
fullProjectSettings=true) at /source/qgis-2.16.3/src/server/qgswmsprojectparser.cpp:130
#8  0x0000000000475344 in QgsWMSServer::getCapabilities (this=0x7fffffffe170, version=..., fullProjectInformation=true) at
/source/qgis-2.16.3/src/server/qgswmsserver.cpp:608
#9  0x0000000000470f7c in QgsWMSServer::executeRequest (this=0x7fffffffe170) at /source/qgis-2.16.3/src/server/qgswmsserver.cpp:174
#10 0x0000000000511564 in QgsServer::handleRequest (this=0x7fffffffe520, queryString=...) at /source/qgis-2.16.3/src/server/qgsserver.cpp:636
#11 0x000000000045419d in main (argc=1, argv=0x7fffffffe648) at /source/qgis-2.16.3/src/server/qgis_map_serv.cpp:43
(gdb) frame 4
#4  0x00000000004e0ad1 in QgsWMSProjectParser::addLayers (this=0x430af40, doc=..., parentLayer=..., legendElem=...,
layerTreeGroup=0x461c420, layerMap=..., nonIdentifiableLayers=..., version=..., fullProjectSettings=true,
    idNameMap=..., layerIDList=...) at /source/qgis-2.16.3/src/server/qgswmsprojectparser.cpp:1262
1262        if ( mProjectParser->restrictedLayers().contains( layerName ) ) //unpublished layer
(gdb) print /s layerName
$1 = {static null = {<No data fields>}, static shared_null = {ref = {_q_value = 474596}, alloc = 0, size = 0, data = 0x78f79a
<QString::shared_null+26>, clean = 0, simpletext = 0, righttoleft = 0, asciiCache = 0, capacity = 0,
    reserved = 0, array = {0}}, static shared_empty = {ref = {_q_value = 602}, alloc = 0, size = 0, data = 0x7ffff428293a
<QString::shared_empty+26>, clean = 0, simpletext = 0, righttoleft = 0, asciiCache = 0, capacity = 0,
    reserved = 0, array = {0}}, d = 0x4918c70, static codecForCStrings = 0x0}
```

Which further debug information should I provide?

#23 - 2017-01-03 07:48 AM - Matthias Kuhn

- Assignee deleted (Matthias Kuhn)

Just a random guess: try changing line 87 in qgsserverprojectparser.h

from

```
const QSet<QString>& restrictedLayers() const { return mRestrictedLayers; }
```

to

```
QSet<QString> restrictedLayers() const { return mRestrictedLayers; }
```

This seems to be a different issue than the one for which this bug was opened originally and I'm not familiar with this part of the code.

#24 - 2017-01-13 12:13 AM - luca76 -

Unfortunately this bug sometime still appear with QGIS Server LTS version.

#25 - 2017-01-14 04:09 AM - Patrick Kirsch

Just for the records, my currently workaround for the segfault situation is to disable caching:
in file src/server/qgswmsprojectparser.cpp, function:

```
QList<QgsMapLayer*> QgsWMSProjectParser::mapLayerFromStyle( const QString& lName, const QString& styleName, bool useCache ) const
```

If "useCache" is generally disabled (=set to false) the segfault does not happen.

#26 - 2017-02-28 02:09 PM - Giovanni Manghi

- Status changed from Reopened to Feedback

Is this issue still affecting the latest point releases of 2.14 and 2.18?

#27 - 2017-04-30 10:01 AM - Giovanni Manghi

- Description updated

Giovanni Manghi wrote:

> *Is this issue still affecting the latest point releases of 2.14 and 2.18?*

Calling for another update on this issue. Still true on 2.18/master?

#28 - 2017-04-30 05:06 PM - Giovanni Manghi

- Regression? set to Yes

#29 - 2017-04-30 05:08 PM - Giovanni Manghi

- Priority changed from Severe/Regression to High

#30 - 2017-05-01 01:10 AM - Giovanni Manghi

- Easy fix? set to No

#31 - 2017-05-26 05:31 PM - Giovanni Manghi

- Status changed from Feedback to Closed

- Resolution set to not reproducable

Closing for lack of feedback.