

QGIS Application - Bug report #11726

Assertion failed in QgsSingleton

2014-11-25 04:48 AM - Martin Dobias

Status:	Closed	
Priority:	Normal	
Assignee:	Matthias Kuhn	
Category:	Web Services clients/WMS	
Affected QGIS version:	master	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:		Resolution:
Crashes QGIS or corrupts data:		Copied to github as #: 19962
Description		
If I add a WMS layer on my machine, it crashes on the following assertion failure: Fatal: ASSERT: "sInstance == 0" in file /home/martin/qgis/git-master/src/core/qgssingleton.h, line 43 Sample code to be pasted in python console to replicate the problem: <pre>l = QgsRasterLayer("crs=EPSG:3857&dpiMode=all&format=image/png&layers=Baker&styles=&url=http://wms.lizardtech.com/lizardtech/iserv/ows", "x", "wms") QgsMapLayerRegistry.instance().addMapLayer(l)</pre> Backtrace: <pre>#0 0x00007ffffefa5ebb9 in __GI_raise (sig=sig@entry=6) at ../nptl/sysdeps/unix/sysv/linux/raise.c:56 #1 0x00007fffffa61fc8 in __GI_abort () at abort.c:89 #2 0x00000000005759b7 in qgisCrash (signal=-1) at /home/martin/qgis/git-master/src/app/main.cpp:306 #3 0x0000000000575b7a in myMessageOutput (type=QtFatalMsg, msg=0x7fff581b06d8 "ASSERT: \"sInstance == 0\" in file /home/martin/qgis/git-master/src/core/qgssingleton.h, line 43") at /home/martin/qgis/git-master/src/app/main.cpp:359 #4 0x00007ffff2aebbb1 in qt_message_output (msgType=msgType@entry=QtFatalMsg, buf=0x7fff581b06d8 "ASSERT: \"sInstance == 0\" in file /home/martin/qgis/git-master/src/core/qgssingleton.h, line 43") at global/qglobal.cpp:2308 #5 0x00007ffff2aef9 in qt_message(QtMsgType, const char *, typedef __va_list_tag __va_list_tag *) (msgType=msgType@entry=QtFatalMsg, msg=0x7ffff2c5ba20 "ASSERT: \"%s\" in file %s, line %d", ap=ap@entry=0x7fff63ffccf8) at global/qglobal.cpp:2429 #6 0x00007ffff2aee804 in qFatal (msg=<optimized out>) at global/qglobal.cpp:2612 #7 0x00007ffff4180dc9 in QgsSingleton<QgsNetworkAccessManager>::QgsSingleton (this=0x7fff5800b190) at /home/martin/qgis/git-master/src/core/qgssingleton.h:43 #8 0x00007ffff417d784 in QgsNetworkAccessManager::QgsNetworkAccessManager (this=0x7fff5800b180, parent=0x0) at /home/martin/qgis/git-master/src/core/qgsnetworkaccessmanager.cpp:91 #9 0x00007ffff809700ed in QgsWmsImageDownloadHandler::QgsWmsImageDownloadHandler (this=0x7fff63ffdd50, providerUri=..., url=..., auth=..., image=0x7fff580ddb80) at /home/martin/qgis/git-master/src/providers/wms/qgswmsprovider.cpp:3165 #10 0x00007ffff8095064f in QgsWmsProvider::draw (this=0x18f7d90, viewExtent=..., pixelWidth=785, pixelHeight=274) at /home/martin/qgis/git-master/src/providers/wms/qgswmsprovider.cpp:558 #11 0x00007ffff809560b7 in QgsWmsProvider::readBlock (this=0x18f7d90, bandNo=1, viewExtent=..., pixelWidth=785, pixelHeight=274, block=0x7fff5800b670) at /home/martin/qgis/git-master/src/providers/wms/qgswmsprovider.cpp:846 #12 0x00007ffff436d2ba in QgsRasterDataProvider::block (this=0x18f7d90, theBandNo=1, theExtent=..., theWidth=785, theHeight=274) at /home/martin/qgis/git-master/src/core/raster/qgsrasterdataprovder.cpp:208 #13 0x00007ffff43b4518 in QgsSingleBandColorDataRenderer::block (this=0x103ef70, bandNo=1, extent=..., width=785,</pre>		

```
height=274)
  at /home/martin/qgis/git-master/src/core/raster/qgssinglebandcolordatarenderer.cpp:67
#14 0x00007ffff43b79a5 in QgsBrightnessContrastFilter::block (this=0x198f730, bandNo=1, extent=..., width=785, height=274)
  at /home/martin/qgis/git-master/src/core/raster/qgsbrightnesscontrastfilter.cpp:125
#15 0x00007ffff43b8c41 in QgsHueSaturationFilter::block (this=0x131cf00, bandNo=1, extent=..., width=785, height=274)
  at /home/martin/qgis/git-master/src/core/raster/qgshuesaturationfilter.cpp:131
#16 0x00007ffff43a7073 in QgsRasterResampleFilter::block (this=0x12373b0, bandNo=1, extent=..., width=785, height=274)
  at /home/martin/qgis/git-master/src/core/raster/qgsrasterresamplefilter.cpp:170
#17 0x00007ffff43769c4 in QgsRasterProjector::block (this=0x122d9c0, bandNo=1, extent=..., width=785, height=274) at
/home/martin/qgis/git-master/src/core/raster/qgsrasterprojector.cpp:780
#18 0x00007ffff4383fc3 in QgsRasterIterator::readNextRasterPart (this=0x7fff63ffead0, bandNumber=1,
nCols=@0x7fff63ffe9bc: 785, nRows=@0x7fff63ffe9c0: 274, block=0x7fff63ffe9f0,
  topLeftCol=@0x7fff63ffe9c4: 0, topLeftRow=@0x7fff63ffe9c8: 0) at
/home/martin/qgis/git-master/src/core/raster/qgsrasteriterator.cpp:96
#19 0x00007ffff439cbdb in QgsRasterDrawer::draw (this=0x7fff63ffea0, p=0x1fdb4d0, viewPort=0x4877450,
theQgsMapToPixel=0x182cee0)
  at /home/martin/qgis/git-master/src/core/raster/qgsrasterdrawer.cpp:59
#20 0x00007ffff4392c82 in QgsRasterLayerRenderer::render (this=0x14814e0) at
/home/martin/qgis/git-master/src/core/raster/qgsrasterlayerrenderer.cpp:199
#21 0x00007ffff4171b36 in QgsMapRendererParallelJob::renderLayerStatic (job=...) at
/home/martin/qgis/git-master/src/core/qgsmaprendererparalleljob.cpp:215
#22 0x00007ffff41731ec in QtConcurrent::FunctionWrapper1<void, LayerRenderJob&>::operator() (this=0x11b5a38, u=...) at
/usr/include/qt4/QtCore/qtconcurrentfunctionwrappers.h:86
#23 0x00007ffff4172f54 in QtConcurrent::MapKernel<QList<LayerRenderJob>::iterator, QtConcurrent::FunctionWrapper1<void,
LayerRenderJob&> >::runIteration (this=0x11b5a00, it=...)
  at /usr/include/qt4/QtCore/qtconcurrentmapkernel.h:73
#24 0x00007ffff4172fd9 in QtConcurrent::MapKernel<QList<LayerRenderJob>::iterator, QtConcurrent::FunctionWrapper1<void,
LayerRenderJob&> >::runIterations (this=0x11b5a00,
  sequenceBeginIterator=..., beginIndex=0, endIndex=1) at /usr/include/qt4/QtCore/qtconcurrentmapkernel.h:82
#25 0x00007ffff417344b in QtConcurrent::IterateKernel<QList<LayerRenderJob>::iterator, void>::forThreadFunction
(this=0x11b5a00) at /usr/include/qt4/QtCore/qtconcurrentiteratekernel.h:263
#26 0x00007ffff4173156 in QtConcurrent::IterateKernel<QList<LayerRenderJob>::iterator, void>::threadFunction
(this=0x11b5a00) at /usr/include/qt4/QtCore/qtconcurrentiteratekernel.h:225
#27 0x00007ffff2aea005 in QtConcurrent::ThreadEngineBase::run (this=0x11b5a40) at
concurrent/qtconcurrentthreadengine.cpp:264
#28 0x00007ffff2aebfee in QThreadPoolThread::run (this=0x7fff6400b180) at concurrent/qthreadpool.cpp:108
#29 0x00007ffff2af832f in QThreadPrivate::start (arg=0x7fff6400b180) at thread/qthread_unix.cpp:349
#30 0x00007fffeceb9182 in start_thread (arg=0x7fff63fff700) at pthread_create.c:312
#31 0x00007fffebf22fbd in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:111
```

Associated revisions

Revision f2332258 - 2014-11-25 02:30 PM - Matthias Kuhn

Don't enforce singleton behavior with Q_ASSERT

Fix #11726

History

#1 - 2014-11-25 04:53 AM - Matthias Kuhn

- *Status changed from Open to Feedback*

The assert could be removed to allow also the multiple instantiation of objects that support the singleton pattern. And I think I'll remove it because for certain classes that seems to be required (and e.g. helps in testing).

But is it not possible to use the singleton instance for the wms download handler?

#2 - 2014-11-25 05:12 AM - Martin Dobias

The WMS download handler cannot use the default QgsNetworkAccessManager because that will not work across threads (the singleton can handle only requests from the main thread). My bad for changing NAM from singleton to not-just-singleton without documenting that properly in doxygen :-/

#3 - 2014-11-25 05:24 AM - Matthias Kuhn

Hmmm... quick local tests suggest that it works...

#4 - 2014-11-25 05:29 AM - Matthias Kuhn

But documentation says its dangerous...

#5 - 2014-11-25 05:30 AM - Matthias Kuhn

- *Status changed from Feedback to Closed*

Fixed in changeset commit:"f2332258c923a43f29993c9ae3d4ef77aa4a8b8f".