

QGIS Application - Bug report #13234

QgsMessageBar crash qgis in mutithread context

2015-08-19 10:37 AM - Luigi Pirelli

Status:	Closed	
Priority:	High	
Assignee:		
Category:		
Affected QGIS version:	2.8.3	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: invalid
Crashes QGIS or corrupts data:	Yes	Copied to github as #: 21295
Description		
<p>During test thread management in a plugin I created a test plugin to calibrate the user communication from the thread.</p> <p>I created a QgsMessageBarItem and set it's message, and when the thred finish I remove the QgsMessageBarItem from message bar and I push a new message notifying the result.</p> <p>During removing the item, the item it's still valid and will arrive also a message notify that does not find a valid object => qgis crash</p> <p>Attached you can find the simple test plugin that crash qgis... just run it press ok in the simple dialog, the thread will start notify message in the messageBar... then after 10 pings will crash (in my case)</p> <p>tested on 2.8.3 and qgisMaster... both cases compiled versions.</p> <p>QGIS version 2.8.3-Wien QGIS code revision 1207e38</p> <p>Compiled against Qt 4.8.6 Running against Qt 4.8.6</p> <p>Compiled against GDAL/OGR 1.11.2 Running against GDAL/OGR 1.11.2</p> <p>Compiled against GEOS 3.4.2-CAPI-1.8.2 Running against GEOS 3.4.2-CAPI-1.8.2 r3921</p> <p>PostgreSQL Client Version 9.4.4 SpatiaLite Version 4.1.1</p> <p>QWT Version 5.2.3 PROJ.4 Version 480</p> <p>QScintilla2 Version 2.8.3 This copy of QGIS writes debugging output.</p>		

History

#1 - 2015-08-19 07:13 PM - Nathan Woodrow

- Assignee deleted (Nathan Woodrow)

My guess is that you are calling the message bar directly from inside your thread? If so you can't alter GUI from a non UI thread. You will need to have a signal on your thread to notify the UI and show the message bar there.

#2 - 2015-08-19 07:15 PM - Nathan Woodrow

- Tag changed from QgsMessageBar crash thread to crash thread

- Resolution set to invalid

Here is an example:

```
from PyQt4.QtCore import QThread, QObject, pyqtSignal

class Worker(QObject):
    done = pyqtSignal()
```

```

finalResult = pyqtSignal(int)

def __init__(self, parent=None):
    super(Worker, self).__init__(parent)

def do_stuff(self):
    l = 0
    for i in range(100):
        l += i
    self.finalResult.emit(l)
    self.done.emit()

def cleanup(self):
    pass

def result(value):
    iface.messageBar().pushMessage("Done", str(value))

thread = QThread()
worker = Worker()
worker.moveToThread(thread)

worker.done.connect(thread.quit)
worker.finalResult.connect(result)
thread.started.connect(worker.do_stuff)
thread.finished.connect(worker.cleanup)

thread.start()

```

#3 - 2015-08-20 01:25 AM - Luigi Pirelli

in the proposed code the worker.run does not have gui interaction... it emits only signals. These are received by the plugin at the same instance level where the Thread is created and started and the worker (Algorithm) is passed.

I'll try more tests adding also thread.finished and movind thread release there

#4 - 2015-08-20 01:32 AM - Nathan Woodrow

That's right. You can't alter the UI in a non UI thread. So worker can't alter the UI it has to talk via signals.

#5 - 2015-08-20 02:58 AM - Luigi Pirelli

- Status changed from Open to Closed

crash seems not related with thread but:

```

iface.messageBar().popWidget(self.progressMessageBarItem)
-->> self.progressMessageBarItem.deleteLater()

```

I can't deleteLater QgsMessageBarItem because already removed by popWidget (http://qgis.org/api/qgsmessagebar_8cpp_source.html#l00174)

#6 - 2015-08-20 03:36 AM - Luigi Pirelli

I'm not a sip expert.. but could be a problem in the python binding specification for the QgsProgressBar items ownership?

<https://setanta.wordpress.com/binding-c/>

regards

Files

TestThreadWorker.zip	51.2 KB	2015-08-19	Luigi Pirelli
----------------------	---------	------------	---------------