

# QGIS Application - Bug report #19630

## crash when using QThreadPool + QRunnable

2018-08-15 05:01 PM - Min Min

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b> Python plugins	
<b>Affected QGIS version:</b> 3.0.3	<b>Regression?:</b> No
<b>Operating System:</b> Windows 10 x64	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b> No	<b>Resolution:</b>
<b>Crashes QGIS or corrupts data:</b> Yes	<b>Copied to github as #:</b> 27457

### Description

### User Feedback

crash when using QThreadPool + QRunnable in plugin

no GUI updating function is executed in QRunnable. QRunnable writes data to local file, create a QgsVectorLayer, getFeatures, fields, addFeatures, addAttributes (fields) to QgsVectorLayer/QgsDataProvider. When editing, QMutex and QMutexLocker is used.

tasks assigned to QThreadPool has finished successfully, QThreadPool.activeThreadCount() returns 0

After tasks finished around 30-60 sec, QGIS crash

### Report Details

**Crash ID:** a279b085a2ce54c5ff69e8b1e03c403266d4bf00

### Stack Trace

```
QCoreApplication::notifyInternal2 :
QEventDispatcherWin32Private::sendTimerEvent :
QEventDispatcherWin32::processEvents :
CallWindowProcW :
DispatchMessageW :
QEventDispatcherWin32::processEvents :
qt_plugin_query_metadata :
QEventLoop::exec :
QCoreApplication::exec :
main :
BaseThreadInitThunk :
RtlUserThreadStart :
```

### QGIS Info

QGIS Version: 3.0.3-Girona  
QGIS code revision: commit:8a899c8758  
Compiled against Qt: 5.9.2  
Running against Qt: 5.9.2  
Compiled against GDAL: 2.2.4  
Running against GDAL: 2.2.4

### System Info

CPU Type: x86\_64  
Kernel Type: winnt  
Kernel Version: 10.0.15063

### History

**#1 - 2018-08-22 11:42 AM - Min Min**

Error caused by reading feature from a QgsVectorLayer in a QThread. As QgsVectorLayer use QgsConnectionPool for fetching feature, and QgsConnectionPool is singleton and always run in main thread (not thread-safe)

timerEvent in Stack Trace is caused by this connection pool. Whether this is a bug, or by design is unclear.

**#2 - 2018-08-22 12:17 PM - Nyal Dawson**

It's not safe to fetch features directly from a thread - you need to first obtain the feature iterator in the main thread, and then interested over the features in the spawned thread.

**#3 - 2018-09-03 03:42 PM - Min Min**

- *Status changed from Open to Closed*

getting feature iterator in main thread and getFeatures in spawned thread works without crash.  
problem resolved !