

## QGIS Application - Bug report #20661

### QgsPoint.clone() not fully reliable? It's original QgsFeature may get corrupted?

2018-11-28 03:49 PM - Juan Manuel Perez

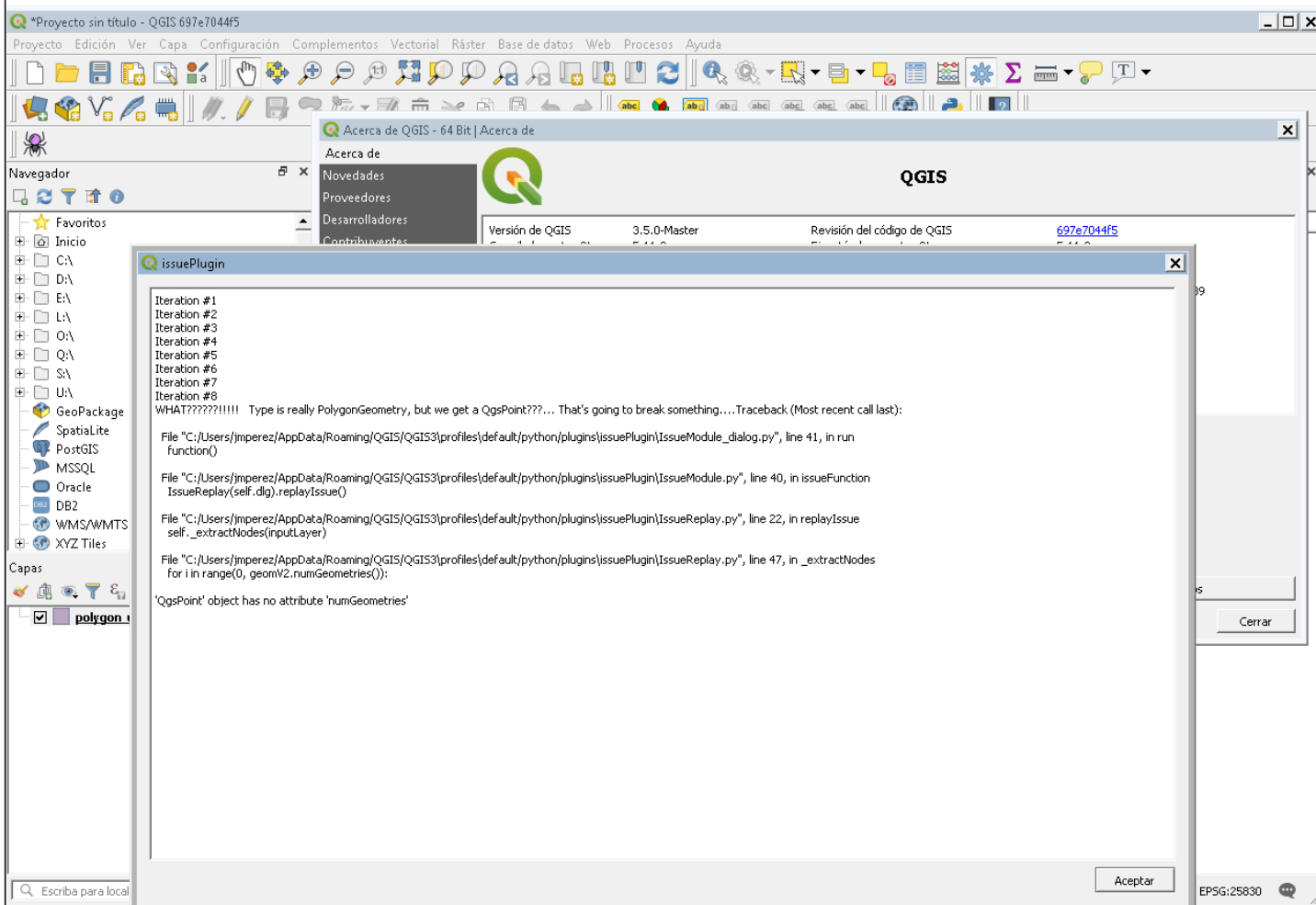
<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Category:</b>	Python bindings / sipify	
<b>Affected QGIS version:</b>	3.5(master)	<b>Regression?:</b> Yes
<b>Operating System:</b>	Windows 7 64 bits	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b> up/downstream
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 28481

#### Description

We have stumbled upon an strange problem and reproducible problem. We attach a plugin to reproduce it.

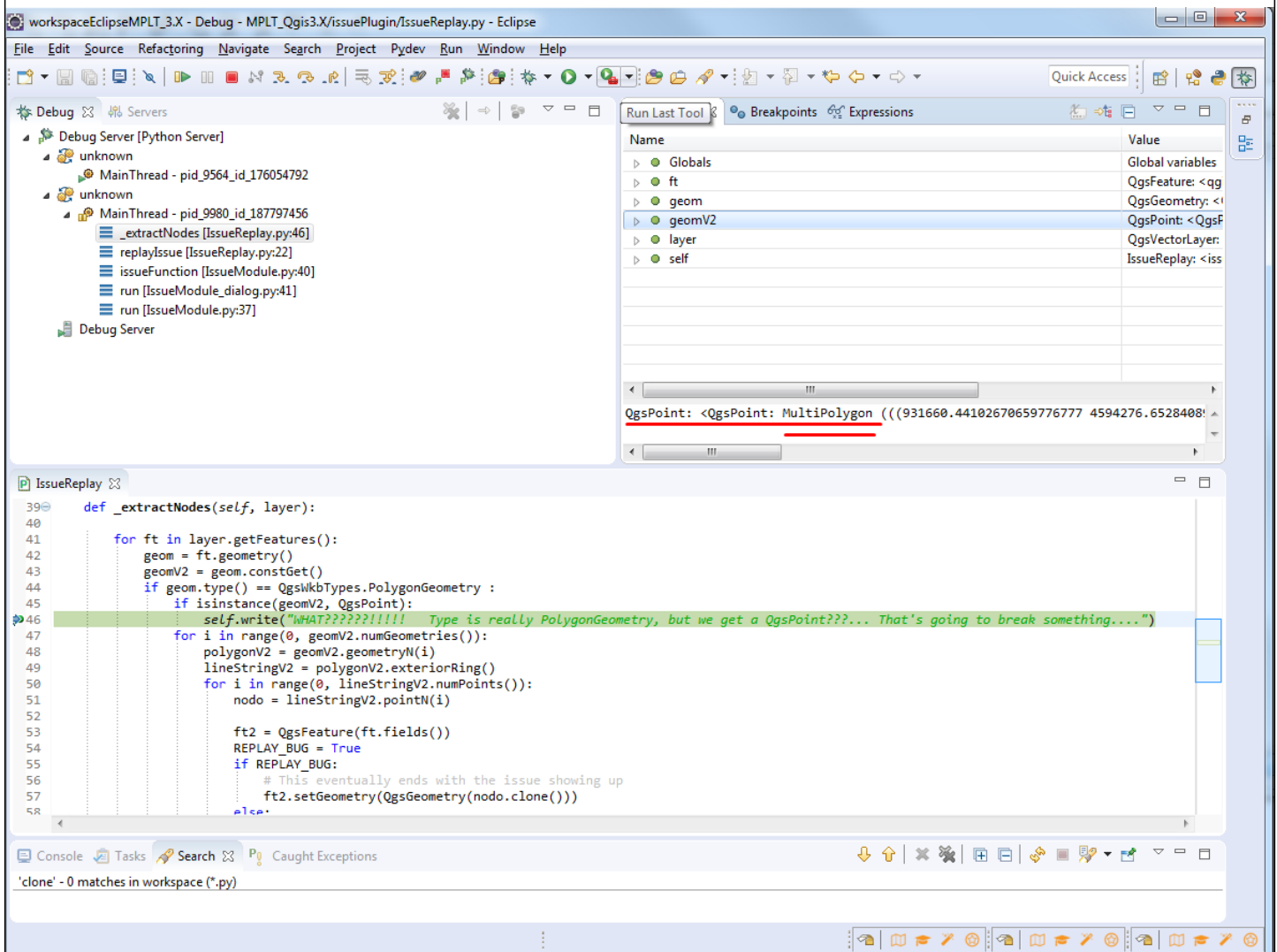
The problem arises in this circumstances:

- We have a layer. We iterate over all its features, then over all their geometries, then all of their nodes (QgsPoints). To reproduce the problem, we use a layer with only a single polygon.
- For each node, we create a new QgsFeature (in memory), and we assign it a new geometry initialized with a clone of the node (QgsPoint)
- The original layer is not modified in any way, nor its features.
- We then repeat the above steps. We may repeat it successfully but only for a limited and variable number of iterations.
- Eventually, the process will break with an unexpected exception.



- More precisely, the exception raises because eventually we get a PolygonGeometry QgsGeometry (as expected), but its actual

geometry (geom.constGet()) is reported as a QgsPoint. Debugging, it looks as "<QgsPoint: MultiPolygon (...)>", as shown in the attached screenshots.



## How to reproduce it

We have prepared a plugin to easily reproduce the issue. We have attached as 'issuePlugin.zip' to this bug report. You only need to install it in QGIS. Then, a toolbar with a spider will appear (see above screenshots). You only need to click this icon. A dialog will open, showing how the test iterates until the unexpected exception is raised.

We have tested it in Qgis 3.5 MASTER and 3.4.1.

## History

**#1 - 2018-11-28 03:51 PM - Juan Manuel Perez**

For now, we use a simple workaround: not using `QgsPoint.clone()`, but creating a brand new `QgsPoint` (`QgsPoint(nodo.x(), nodo.y())`). But we think such a strange behavior should need be reviewed, as it may lead to unexpected or unreliable results...

**#2 - 2018-11-29 11:04 AM - Martin Dobias**

I can replicate on linux too.

Simplified code to reproduce the issue:

```
import sip
for i in range(50):
    geom = QgsGeometry.fromWkt("MULTIPOLYGON(((1 1, 1 2, 2 2, 2 1, 1 1)))")
    geomV2 = geom.constGet()
    if isinstance(geomV2, QgsPoint):
        print("WHAT?????!!!!!! " + hex(sip.unwrapinstance(geomV2)))

    pt = QgsGeometry.fromWkt("POINT( 3 4 )")
    pt_g = pt.constGet()
    pt_clone = pt_g.clone()
    pt_x = QgsGeometry(pt_clone)
    print("new pt " + hex(sip.unwrapinstance(pt_clone)))
print("-----")
```

#3 - 2019-02-05 10:39 PM - Matthias Kuhn

- Resolution set to up/downstream
- Status changed from Open to Closed

Issue fixed upstream with upcoming sip version 4.19.14.

Files			
iteration_screenshot.png	82.5 KB	2018-11-28	Juan Manuel Perez
debugging_screenshot.png	116 KB	2018-11-28	Juan Manuel Perez
issuePlugin.zip	487 KB	2018-11-28	Juan Manuel Perez