QGIS Application - Bug report #15683
python/sip: qgsvectorlayer instance improperly turns into qgsmaplayer when reference variable deleted

2016-10-09 04:22 AM - Mathieu Pellerin - nIRV

Status:	Closed		
Priority:	Severe/Regression		
Assignee:			
Category:	Python plugins		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	
Crashes QGIS or corrupts data:	Yes	**Copied to github as #:**	23606

Description

I've noticed a serious regression under processing, which results in processing being unusable after a first algorithm is run and file-based output layers are added. Those output layers, when added automatically by processing, are "missing" a dataProvider() object when trying to access it through python. Hopefully the following steps will make things clearer.

Steps to reproduce

1. Open QGIS, create a new project
2. Add a polygon layer (the input source type doesn't matter)
3. Open processing, and select the "polygons to line" algorithm (under QGIS)
4. In the algorithm's setting dialog, change the output to be saved as a shapefile, on your desktop (or any other location, it doesn't matter)
5. Run the algorithm, and take note of the output layer added onto your project (it should be the top item in your layer tree)
6. Open QGIS' python console, and execute the following code: iface.legendInterface().layers()[0].name()
7. The output layer name, "Lines from polygon" should be printed
8. Now in the console, execute the following code: iface.legendInterface().layers()[0].dataProvider().name()
9. An error will be printed, saying that the QgsMapLayer object has no attribute dataProvider (which is wrong, it should print 'ogr').

As a result of whatever is causing this, subsequent use of processing algorithms is broken, as the processing code assumes a dataProvider() object will always be attached to vector layers (which it should).

Note: while this issue was opened on the assumption that it was a processing-specific issue, it turned out something more fundamental is broken; see comments below

Associated revisions

Revision 03f08a6c - 2016-10-17 10:47 AM - Matthias Kuhn

Fix QgsMapLayer ConvertToSubClassCode (#3611)

Fix #15683

Revision 73afe217 - 2016-10-18 12:05 PM - Matthias Kuhn

Fix QgsMapLayer ConvertToSubClassCode (#3611)

Fix #15683

History

#1 - 2016-10-09 04:48 AM - Victor Olaya

It's an important issue, no doubt, but i dont understand how a layer object can be missing that method... It's puzzling...

If that's the case, i guess models wont work, right? HAve you tested?

#2 - 2016-10-09 08:16 PM - Mathieu Pellerin - nIRV

- File qgsvectorlayer_error.py added

- Category changed from Processing/Core to Python plugins

Ok, this goes beyond processing, something is broken at the sip level. Here's a greatly narrowed steps to reproduce the issue:

Steps to reproduce

1. Launch QGIS and create a blank project
2. Toggle the QGIS python console on, and load the attached script (qgsvectorlayer_error.py) in the editor
3. Replace the file path in the script (i.e. '/media/webmaster/Data/Gis/Math/ELCs 2012/ELCs2012.shp') with a shapefile on your local machine
4. Execute the script
5. The last line, iface.legendInterface().layers()[0].dataProvider().name(), should print 'ogr' in the python console, but it'll crash and throw an error

If you check iface.legendInterface().layers()[0], you'll notice it is considered to be a QgsMapLayer, which is the wrong cast, and explains why dataProvider() isn't accessible.

#3 - 2016-10-12 08:45 PM - Martin Dobias

There is indeed some change in the internals, not sure yet if it is related to SIP or to the switch to Python3.

It can be replicated even just in the console. It seems that when the temporary python wrapper of QgsMapLayerRegistry instance is deleted, things go wrong in QGIS 3. Also worth noting that for some reason the layer initially has 4 refs in QGIS 2, while only 3 refs in QGIS 3 - the missing reference may be the reason why things go wrong (QgsMapLayerRegistry has the same number of refs in both QGIS 2 and QGIS 3).

I made a series of dumps with SIP:

1. after being added to registry (registry referenced in console)

```
>>> r = QgsMapLayerRegistry.instance()
>>> QgsMapLayerRegistry.instance().addMapLayer(QgsVectorLayer('my_layer.shp', 'testing', 'ogr'))
>>> QgsMapLayerRegistry.instance().mapLayers()
>>> sip.dump(iface.legendInterface().layers()[0])
```

2. after garbage collection (registry still referenced in console)

```
>>> gc.collect()
>>> sip.dump(iface.legendInterface().layers()[0])
```

3. after getting rid of registry reference in console

```
>>> del r
>>> gc.collect()
>>> sip.dump(iface.legendInterface().layers()[0])
```

Output from QGIS 3.0

<qgis._core.QgsVectorLayer object at 0x7f59f53420d8>

Reference count: 3

Address of wrapped object: 0x5045080

Created by: Python

To be destroyed by: C/C++

Parent wrapper: <qgis._core.QgsMapLayerRegistry object at 0x7f59f5342168>

Next sibling wrapper: NULL

Previous sibling wrapper: NULL

First child wrapper: NULL

<qgis._core.QgsVectorLayer object at 0x7f59f53420d8>

Reference count: 2

Address of wrapped object: 0x5045080

Created by: Python

To be destroyed by: C/C++

Parent wrapper: <qgis._core.QgsMapLayerRegistry object at 0x7f59f5342168>

Next sibling wrapper: NULL

Previous sibling wrapper: NULL

First child wrapper: NULL

<qgis._core.QgsMapLayer object at 0x7f59f53420d8>

Reference count: 1

Address of wrapped object: 0x5045080

Created by: C/C++

To be destroyed by: C/C++

Parent wrapper: NULL

Next sibling wrapper: NULL

Previous sibling wrapper: NULL

First child wrapper: NULL

Output from QGIS 2.18

<qgis._core.QgsVectorLayer object at 0x7efcb04f0b98>

Reference count: 4

Address of wrapped object: 0x639d000

Created by: Python

To be destroyed by: C/C++

Parent wrapper: <qgis._core.QgsMapLayerRegistry object at 0x7efcb04f0b00>

Next sibling wrapper: NULL

Previous sibling wrapper: NULL

First child wrapper: NULL

<qgis._core.QgsVectorLayer object at 0x7efcb04f0b98>

Reference count: 3

Address of wrapped object: 0x639d000

Created by: Python

To be destroyed by: C/C++

Parent wrapper: <qgis._core.QgsMapLayerRegistry object at 0x7efcb04f0b00>

Next sibling wrapper: NULL

Previous sibling wrapper: NULL

First child wrapper: NULL

<qgis._core.QgsVectorLayer object at 0x7efcb04f0b98>

Reference count: 2

Address of wrapped object: 0x639d000

Created by: Python

To be destroyed by: C/C++

Parent wrapper: NULL

Next sibling wrapper: NULL

Previous sibling wrapper: NULL

First child wrapper: NULL

#4 - 2016-10-12 10:48 PM - Mathieu Pellerin - nIRV

Sharing some more experiment.

The following code **does not** work:

```
def load(fileName):
    qgslayer = QgsVectorLayer(fileName, 'testing', 'ogr')
    QgsMapLayerRegistry.instance().addMapLayers([qgslayer])

load('/media/webmaster/Data/Gis/Math/ELCs 2012/ELCs2012.shp')
iface.legendInterface().layers()[0].dataProvider().name()
```

The following code **does** work:

```
qgslayer = None

def load(fileName):
    global qgslayer
    qgslayer = QgsVectorLayer(fileName, 'testing', 'ogr')
    QgsMapLayerRegistry.instance().addMapLayers([qgslayer])

load('/media/webmaster/Data/Gis/Math/ELCs 2012/ELCs2012.shp')
iface.legendInterface().layers()[0].dataProvider().name()
```

#5 - 2016-10-12 11:23 PM - Victor Olaya

IF it's not a Processing bug, shouldn't we reopen it with another name to give it more visibility?

I think some devs might overlook it thinking it's only related to Processing...

#6 - 2016-10-14 12:45 AM - Mathieu Pellerin - nIRV

*- Subject changed from processing: file-based layers added as a result of an algorithm are missing a dataProvider() object, breaking subsequent
algorithms to python/sip: qgsvectorlayer instance improperly turns into qgsmaplayer when reference variable deleted
- Assignee deleted (Victor Olaya)*

Victor, I've renamed the title of this issue and have removed the assigned to value to indicate the need to address this severe regression.

#7 - 2016-10-16 11:42 PM - Matthias Kuhn

Can you check the PR 3611?

https://github.com/qgis/QGIS/pull/3611

I don't really have an explanation why that should fix anything but cleaned some sip conversion code and it seems at least no longer to return any "QgsMapLayer" typed objects.

#8 - 2016-10-17 01:45 AM - Mathieu Pellerin - nIRV

Matthias, it **did** fix this issue, I've got a working processing tool (db manager, etc.) again! :)

#9 - 2016-10-17 01:46 AM - Anonymous

- Status changed from Open to Closed

Fixed in changeset commit:"03f08a6c79fc4016d7ed0b89262d916cb3b70bab".

Files

qgsvectorlayer_error.py	258 Bytes	2016-10-09	Mathieu Pellerin - nIRV