QGIS Application - Bug report #12206
processing: apply a difference geoalgorithm using two memory layers not working anymore

2015-02-14 09:18 PM - Mathieu Pellerin - nIRV

Status:	Closed		
Priority:	Severe/Regression		
Assignee:	Victor Olaya		
Category:	Processing/Core		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	fixed/implemented
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	20395

Description

I believe that's a pretty recent regression. When using two memory layers to apply a difference geoalgorithm through processing, it fails and returns no feature. Going via fTools works fine.

Steps to reproduce:
1. Create a new project
2. Create a new memory layer (memoryA), and add one polygon
3. Create another new memory layer (memoryB), and add a polygon that will cover part of the one you added above
4. Open the QGIS geoalgorithm's difference window in processing, and set memoryA as input layer, and memoryB as difference layer
5. Run the algorithm, notice the absence of a resulting polygon

Associated revisions

Revision f64730e8 - 2015-02-17 02:07 PM - Jürgen Fischer

workaround to create unique memory layer data sources (fixes #12206)

History

#1 - 2015-02-17 02:53 AM - Jürgen Fischer

Is it a regression? Your memory layers probably have the same data source uri (polygon?crs=EPSG:SOMECRS), that's what processing uses to identify the layers, but with layers having the same data source uri it just picks the first layer and intersects it with itself.

#2 - 2015-02-17 03:19 AM - Mathieu Pellerin - nIRV

Jef, right, that might be it (why it never affected me before is pure mystery). I'll double check with a QGIS 2.6 build to see whether processing fails to distinguish two layers having same source uri, or whether that has become an issue during the development cycle for 2.8.

That said, this does raise a problem that needs addressing by processing. If I choose two different layers having two different layer names, processing should be able to pick the right layer, irrespective of having the same source uri. fTools is able to do so just fine.

Pasting to new memory layer creates a data source uri that simply states the data type (i.e. Point, Line, Polygon); so say someone is pasting two polygons onto two temporary memory layers to do a geotransformation (union, division, etc.), it would also fail.

#3 - 2015-02-17 03:22 AM - Mathieu Pellerin - nIRV

Ie, right now, processing assumes that a source uri is unique to a layer name, which is wrong. The code should handle the scenario raised in the steps to

reproduce, that is:

Layer: MemoryA (data source uri:polygon?crs=something)
Layer: MemoryB (data source uri:polygon?crs=something)

#4 - 2015-02-17 05:07 AM - Jürgen Fischer
- Status changed from Open to Closed

Fixed in changeset commit:"f64730e8d196e40bfecd8bea67ae64b69fba0cb3".

#5 - 2015-02-17 05:12 AM - Jürgen Fischer
- Resolution set to fixed/implemented

Mathieu Pellerin - nIRV wrote:

> *That said, this does raise a problem that needs addressing by processing. If I choose two different layers having two different layer names, processing should be able to pick the right layer, irrespective of having the same source uri. fTools is able to do so just fine.*

The layer name shouldn't matter either - each layer has an unique id.

> *Ie, right now, processing assumes that a source uri is unique to a layer name, which is wrong. The code should handle the scenario raised in the steps to reproduce, that is:*

Yes, that's what I said. But I wonder when that was changed, dataobjects.getObjectFromUri is widely used all over processing and apparently it wasn't changed lately (the current version looks very similar to the initial version of commit:807ffaa).

So I guess it's not a regression, but tied I layers with the same data source - which is pretty rare for "normal" layers, but very common for memory layers.

commit:f64730e adds an additional parameter "memoryid" to the data source uri where QGIS creates memory layers (ie. "new scratch layer", "paste features as memory layer" and "shortest path" in the roadgraph plugin) so that the datasource uri stored in the layer is unique for memory layers. The provider ignores that parameter.

But it's a kludge - memory layers created by plugins will still trigger that problem in processing and there might be plugins that parse the layers dataSourceUri that cannot cope with this (although I'd consider those flawed).

#6 - 2015-06-07 04:33 AM - Giovanni Manghi
- Category changed from 94 to Processing/Core