QGIS Application - Bug report #15272
QgsVectorLayer's attributeValueChanged SIGNAL emitted when it should not

2016-07-13 09:17 AM - gcarrillo -

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	Vectors		
Affected QGIS version:	2.14.2	**Regression?:**	No
Operating System:	GNU/Linux	**Easy fix?:**	No
Pull Request or Patch supplied:		**Resolution:**	end of life
Crashes QGIS or corrupts data:		**Copied to github as #:**	23208

Description

I'm working on a plugin and trying to connect a SLOT to layer.attributeValueChanged SIGNAL. However, I found that such SIGNAL is (weirdly) emitted in the following scenario:

1. Load a vector file (I've tested this with a Shapefile and an SQLITE file) with a numeric (int or real) field, whose value for the first feature is NULL.
2. Open the attribute table of the vector layer.
3. Start an edit session cliking on "Toggle Edit Mode" button from the Attribute table window.

As soon as the edit session starts, the SIGNAL is emitted. Now, if I click on the "Toggle Edit Mode" button from the Attribute table window, I have to choose among Save, Cancel, or Close Without Saving options, even without changing any attribute value at all. Once I choose Close Without Saving, the layer.attributeValueChanged SIGNAL is emitted once more.

####################################
Note: I use this code snippet in the QGIS Python console to be notified when layer.attributeValueChanged is emitted:

```
##
def f(fid, idx, value):
  print fid,idx,value

l = iface.activeLayer()
l.attributeValueChanged.connect( f )##
```

I'm getting these messages printed (4 is the index of the numeric field whose value for the first feature is NULL):

0 4 NULL # When I open the edit session.
0 4 NULL # When I close the edit session.

If you want to reproduce the problem, you can use a sample Shapefile from here: http://downloads.tuxfamily.org/tuxgis/tmp/test.zip

####################################

Is this a bug? Do you know why layer.attributeValueChanged is being emitted in such scenario? I've noticed that sometimes a blockSignals() method is used in QGIS source code; couldn't we use blockSignals() to avoid emitting the SIGNAL in this case?

History

#1 - 2017-05-01 01:05 AM - Giovanni Manghi

- Easy fix? set to No

- Regression? set to No

#2 - 2019-03-09 03:08 PM - Giovanni Manghi

- Resolution set to end of life

- Status changed from Open to Closed

End of life notice: QGIS 2.18 LTR

Source:

http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/

QGIS 3.4 has recently become our new Long Term Release (LTR) version. This is a major step in our history – a long term release version based on the massive updates, library upgrades and improvements that we carried out in the course of the 2.x to 3x upgrade cycle.

We strongly encourage all users who are currently using QGIS 2.18 LTR as their preferred QGIS release to migrate to QGIS 3.4. This new LTR version will receive regular bugfixes for at least one year. It also includes hundreds of new functions, usability improvements, bugfixes, and other goodies. See the relevant changelogs for a good sampling of all the new features that have gone into version 3.4

Most plugins have been either migrated or incorporated into the core QGIS code base.

We strongly discourage the continued use of QGIS 2.18 LTR as it is now officially unsupported, which means we'll not provide any bug fix releases for it.

You should also note that we intend to close all bug tickets referring to the now obsolete LTR version. Original reporters will receive a notification of the ticket closure and are encouraged to check whether the issue persists in the new LTR, **in which case they should reopen the ticket**.

If you would like to better understand the QGIS release roadmap, check out our roadmap page! It outlines the schedule for upcoming releases and will help you plan your deployment of QGIS into an operational environment.

The development of QGIS 3.4 LTR has been made possible by the work of hundreds of volunteers, by the investments of companies, professionals, and administrations, and by continuous donations and financial support from many of you. We sincerely thank you all and encourage you to collaborate and support the project even more, for the long term improvement and sustainability of the QGIS project.