QGIS Application - Bug report #11395
QgsAttributeForm not entering edit mode

2014-10-13 02:25 AM - Andrew McClure

Status:	Closed		
Priority:	Normal		
Assignee:	Matthias Kuhn		
Category:	Attribute table		
Affected QGIS version:	2.4.0	**Regression?:**	No
Operating System:	OSX	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	end of life
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	19678

Description

Use case:

Pre conditions:

A valid UI file and Init function have been specified via layer.setEditForm and layer.setEditFormInit

Steps:

1. User enters Attribute Table
2. Selects edit mode
3. Switches to Open form

Form opens but is not editable

Alternative flow 1.

2. Switch to open Form
3. Select edit mode

Form is still not in edit mode

Alternative flow 2.

1. Select layer
2. Select feature
3. Select "Toggle Editing" to enable editing
3. Click 'i' Identify Features
4. Edit Form opens (if configured to open directly) but dialog is not editable.

Finally, to force the form to open as editable it is necessary to programmatically start editing then directly call openFeatureForm:

```
layer.startEditing()
iface.openFeatureForm(layer, feature, True)
```

Example available using this plugin:

http://plugins.qgis.org/plugins/TelemetryLayer/

History

#1 - 2014-10-13 02:31 AM - Matthias Kuhn

- Status changed from Open to Feedback

In your screenshot you did not select a feature. We could think about pre-selecting the first feature to make it more intuitive.

I recently discussed with Alexandre about this (or a similar issue) and realized, that we were talking about two different things.
He wanted a feature form to edit feature attributes without saving them to the feature but just to get an edited feature back (in a python plugin). Is this your use-case as well?

#2 - 2014-10-16 03:50 PM - Andrew McClure

There are a number of use cases.

1. Be able to enter edit mode using a custom form from the Attribute Table dialog. Currently the custom feature form only opens as not editable regardless of layer state

2. Be able to enter an editable OR non editable version of the feature by selecting 'i' from the layer itself - and respecting the layer's edit value

3. Programmatically be able to edit the feature by calling openFeatureForm with an existing issue. I have found a work around for this and posted it to the mailing list and included it below:

- Use a feature saved say to a widget's data properties.
  - On a callback, copy the attributes the current layer.getFeatures() to ensure the old feature is in sync.
  - Make the layer editable and pass the "updated" feature to the openFeatureForm

#3 - 2014-10-16 11:36 PM - Matthias Kuhn

Is there something wrong with the first two scenarios? These were working nicely last time I looked.

Related to python code, there was a problem when directly using QgsAttributeDialog and toggling edit mode after opening the dialog. This has been solved in commit:a3dad24. openFeatureForm should have always worked.

Can you clarify what you are trying to do, what is not working, point to some sample code (exact locations, not a whole plugin. Most appreciated would be a minimal working example that can be run from the python console). It could also be related to the provider you use or if a subset string is set on your layer.

#4 - 2014-10-17 03:06 AM - Andrew McClure

Hi Matthias

I would certainly love this to be a problem with my own code!

Unfortunately, for you to view the flow/recreate it is not as trivial as running a few lines of code. However, I can point to the places in my code. And there is also the feature form available from the AttributeTable if you simply run the plugin TelemetryLayer and follow the steps to create a new Telemetry Layer.

As for what is not working/what we are trying to do, this should be clear from my last issue update. I simply want the edit mode of the layer to be respected with regards opening a custom feature form.

As to the code, here are the critical bits:

1. This is the custom edit form init:

https://github.com/nzfarmer1/telemetrylayer/blob/master/topicmanagers/editformfactory.py

2. Here is the class that gets invoked and performs the dialog logic. You can see the debug statement showing the editable state.

https://github.com/nzfarmer1/telemetrylayer/blob/master/tltopicmanager.py : Line 107

class tlSysFeatureDialog(tlFeatureDialog):

3. Here is the code that programmatically opens a featureForm using a feature stored in a QWidget which has been refreshed.
(Note by changing the feature that is passed to the one that is returned from Layer.getFeature, the dialog fails to open in edit mode regardless of the layer's state.)

https://github.com/nzfarmer1/telemetrylayer/blob/master/tlbrokerconfig.py : Line 177+

#5 - 2015-12-20 08:02 AM - Giovanni Manghi
- Status changed from Feedback to Open

#6 - 2017-05-01 01:08 AM - Giovanni Manghi
- Regression? set to No
- Easy fix? set to No

#7 - 2019-03-09 03:12 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/

Files

screenshot.pdf	91.7 KB	2014-10-13	Andrew McClure