QGIS Application - Bug report #11099
iface.openFeatureForm() does not update feature

2014-08-26 02:22 AM - Alexandre Neto

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	Forms		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	19431

Description

In QGIS 2.2, the iface.openFeatureForm(layer,feature) would change the features attributes but would not update the feature in the respective layer. In fact, It could be used in temporary (not yet commited) features. This allowed some feature attributes or geometry manipulation before layer.addFeature(feature) or layer.updateFeature(feature).

In 2.4, this was not replicable because of this bug:

#10865

In master (2.5 1b205be), after pressing the feature form ok button, a new feature is automaticly added to the layer (or updated in case of already existing features), and the temporary feature attributes are not changed.

Not sure if this is a intended behaviour (hope not), or a regression.

I though it might be something to do with updateFeatureOnly boolean value, but setting it to True does not seems to make any difference. The API says its UNUSED:

[[http://qgis.org/api/classQgisInterface.html#a11b90f38afd09ac5e9f363933ee4a509]]

The "issue" can be seen with the following script:

```
mc = iface.mapCanvas()
layer = mc.currentLayer()
temp_feature = QgsFeature()
attributes = []

# getting default values (for primary keys)
provider = layer.dataProvider()
for j in layer.pendingAllAttributesList():
    if provider.defaultValue(j):
        attributes.append(provider.defaultValue(j))
    else:
        attributes.append(None)
temp_feature.setAttributes(attributes)

iface.openFeatureForm(layer, temp_feature)

print temp_feature.attributes()
```

Associated revisions

Revision 4b8288c0 - 2014-08-26 04:07 PM - Matthias Kuhn

openFeatureForm: Update attributes on provided when used modal
Fix #11099

History

#1 - 2014-08-26 02:52 AM - Matthias Kuhn

The issue with the attribute dialog was, that it would always open in modal mode. That means, no map interaction is possible at the same time as the main thread waits for the form to be finished. That was preventing certain features like picking a related feature from the feature form (Denis is working on that). Therefore this was changed.
The behavior you describe is a by-product of this change.

Now there is a new parameter you can pass to openFeatureForm that will allow the attribute dialog to be shown modal or non-modal. I will check that if used in the previous mode (modal) it will return an updated feature. That means no change necessary for you.

#2 - 2014-08-26 03:48 AM - Alexandre Neto

The default for showModal is True, and it does not change the attributes. Setting it to False will not wait for ok, and therefore will carry with the script printing the not changed feature. After pressing Ok in the form the temp_features attributes don't change anyway.

#3 - 2014-08-26 07:07 AM - Matthias Kuhn

- Status changed from Open to Closed

Fixed in changeset commit:"4b8288c026e1cb7d6e1bd30eab6c09aec9b06404".

#4 - 2014-08-28 03:30 AM - Alexandre Neto

I have been testing the changes in master. openFeatureForm() now changes the input features attributes in modal mode, but not in non-modal. Is there a reason for that not to occur? Modal or not after exec() the input feature attributes could\\should always be changed for further use.

Besides that, in current master, openFeaterForm() always tries to update\\add the input feature in the layer (using "mLayer->addFeature(updatedFeature)" I think).

Back in 2.2 this would not happen, the feature add\\update to the layer probably happened outside the openFeatureForm(). This gave more flexibility, since one could intersect the attributes changes, do some checking or new automated changes and only then update\\add the feature (or features) to the layer.

I keep thinking about about the "updateFeatureOnly" attribute that is unused. Could this be use to prevent the feature to be added\\updated in the layer, by setting it to true?

Since I'm not sure of what really is the intended behavior or the implication around this, I will leave to you to reopen the issue in case you think you should.

Thanks for your help.

#5 - 2014-09-02 08:32 AM - Matthias Kuhn

If it's opened non-modal (that is not via exec() but via show()) control returns immediately to the caller. That means, if the feature is updated, it might happen anytime, unexpectedly. And it might even have been deleted meanwhile (resulting in a crash).

Therefore, you should rather just create your own QgsAttributeDialog or QgsFeatureForm directly and connect to the appropriate signals (attributeChanged(), featureSaved(feature)), that will give the plugin the possibility to properly handle things.

The last comment is something I have recently been thinking about. The attribute dialog should be able to work in different modes

- Directly change feature on layer
- Add feature
- Only change provided feature
- Change multiple features (that might be a subclass QgsMultiFeatureForm)