

QGIS Application - Bug report #14805

"On map identification" option of relation reference widget not working properly

2016-05-11 10:04 PM - zicke -

Status:	Closed	
Priority:	Normal	
Assignee:	Matthias Kuhn	
Category:	Relations	
Affected QGIS version:	master	Regression?: No
Operating System:	Ubuntu	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: end of life
Crashes QGIS or corrupts data:	No	Copied to github as #: 22762
Description		
The "On map identification" option does not work if "Add feature" dialog is open. The identification works if you first add the (geometryless) feature and close the dialog (saving feature is not needed) and then identify the geometry.		

History

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

- *Regression? set to No*

- *Easy fix? set to No*

#2 - 2018-08-19 01:05 AM - Simon South

This problem still occurs when trying to add related features (through a many-to-many relation) via the attribute table. It happens because the "Add Feature" dialog is [shown by calling its exec\(\) method](#), effectively [making the dialog application-modal](#) and preventing the user from interacting with the main window while it is open (so the feature-identification tool cannot function). Issue #11300 addressed the same problem occurring outside this specific context.

For the courageous, a simple workaround is to apply a similar fix by changing [line 40 of qgsguivectorlayertools.cpp](#) from

```
bool added = a.addFeature( defaultValues );
```

to

```
bool added = a.addFeature( defaultValues, false );
```

which effectively makes the dialog non-modal and allows the feature-identification tool to function normally.

This isn't really a solution, though:

- There's no parent-child relationship between the attribute table and the "Add Feature" dialog (nor any obvious way of specifying one presently), so one window or the other tends to get lost behind the application.

- Along the same lines, there's no way to specify [window modality](#) for the dialog, which is probably what's really needed here.

- The user is no longer blocked from clicking on the attribute table's controls while the dialog is open, so it may be possible to get the application or the layer in an inconsistent state.

I think a "proper" fix would involve refactoring the "Add Feature" dialog's logic, currently spread across [QgsFeatureAction::addFeature\(\)](#) and [QgsFeatureAction::newDialog\(\)](#), into a separate class that inherits from QDialog and simply returns the user's description of a new feature (or perhaps the feature itself) to its caller. That would provide the flexibility to use the dialog in multiple contexts and a straightforward way of specifying things like the

dialog's parent and modality when important.

#3 - 2018-08-19 10:49 AM - Giovanni Manghi

It seems you could provide a patch via a pull request on github?

Simon South wrote:

This problem still occurs when trying to add related features (through a many-to-many relation) via the attribute table. It happens because the "Add Feature" dialog is [shown by calling its exec\(\) method](#), effectively [making the dialog application-modal](#) and preventing the user from interacting with the main window while it is open (so the feature-identification tool cannot function). Issue #11300 addressed the same problem occurring outside this specific context.

For the courageous, a simple workaround is to apply a similar fix by changing [line 40 of qgsguivectorlayertools.cpp](#) from

[...]

to

[...]

which effectively makes the dialog non-modal and allows the feature-identification tool to function normally.

This isn't really a solution, though:

- There's no parent-child relationship between the attribute table and the "Add Feature" dialog (nor any obvious way of specifying one presently), so one window or the other tends to get lost behind the application.

- Along the same lines, there's no way to specify [window modality](#) for the dialog, which is probably what's really needed here.

- The user is no longer blocked from clicking on the attribute table's controls while the dialog is open, so it may be possible to get the application or the layer in an inconsistent state.

I think a "proper" fix would involve refactoring the "Add Feature" dialog's logic, currently spread across [QgsFeatureAction::addFeature\(\)](#) and [QgsFeatureAction::newDialog\(\)](#), into a separate class that inherits from QDialog and simply returns the user's description of a new feature (or perhaps the feature itself) to its caller. That would provide the flexibility to use the dialog in multiple contexts and a straightforward way of specifying things like the dialog's parent and modality when important.

#4 - 2018-08-19 01:20 PM - Simon South

Giovanni Manghi wrote:

| It seems you could provide a patch via a pull request on github?

Sure, but I don't consider the workaround I posted to be a suitable fix. The code needs to be refactored and this could be a significant undertaking, considering the "Add Feature" dialog is used in multiple places.

Does the idea of pulling out the UI-related portions of QgsFeatureAction into a separate QDialog subclass sound reasonable to you? If I could demonstrate this working, is it something the QGIS team might accept?

#5 - 2018-08-20 10:45 AM - Giovanni Manghi

Simon South wrote:

Giovanni Manghi wrote:

It seems you could provide a patch via a pull request on github?

Sure, but I don't consider the workaround I posted to be a suitable fix. The code needs to be refactored and this could be a significant undertaking, considering the "Add Feature" dialog is used in multiple places.

Does the idea of pulling out the UI-related portions of QgsFeatureAction into a separate QDialog subclass sound reasonable to you? If I could demonstrate this working, is it something the QGIS team might accept?

your suggestion are valuable, but to make them gain traction you should make a pull request or eventually comment the existing code, but things better done on github. Thanks!

#6 - 2019-03-09 04:10 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/>