

QGIS Application - Bug report #10739

Custom Forms with Python logic don't work anymore as the dialog closes even though the .accept() function is not run

2014-06-26 09:42 PM - Tudor Bărăscu

Status: Closed	
Priority: Normal	
Assignee:	
Category: Forms	
Affected QGIS version: master	Regression?: No
Operating System:	Easy fix?: No
Pull Request or Patch supplied:	Resolution:
Crashes QGIS or corrupts data:	Copied to github as #: 19123
Description	
<p>One wonderful QGIS feature is the ability to add Python logic to forms. Thank you Nathan and Tim for showing how.</p> <p>The dialog closes at the push of the ok button even though the validate method is not accessing the dialog.accept().</p> <p>I have made a small project that illustrates the new behavior:</p> <p>https://www.dropbox.com/s/itdpp097sm5btkd/form_problem.zip</p> <p>Is this a bug or there is something new that I haven't considered.</p> <pre>#!/usr/bin/env python 1. coding: utf-8 from PyQt4.QtCore import * from PyQt4.QtGui import * myDialog = None nameField = None def formTest(dialog, layerid, featureid): global myDialog myDialog = dialog global nameField nameField = dialog.findChild(QLineEdit, "name") buttonBox = dialog.findChild(QDialogButtonBox, "buttonBox") buttonBox.accepted.disconnect(myDialog.accept) buttonBox.accepted.connect(validate) buttonBox.rejected.connect(myDialog.reject) def validate(): if not len(nameField.text()) >= 4: msgBox = QMessageBox() msgBox.setText("Name field must have at least 4 letters.") msgBox.exec_() else: myDialog.accept()</pre>	
Related issues:	
Duplicated by QGIS Application - Bug report # 8885: Buttons (Ok, Cancel) disa...	Closed 2013-10-17

History

#1 - 2014-06-26 11:57 PM - Matthias Kuhn

- Target version changed from Version 2.4 to Future Release - High Priority

See also #8885 and [this gis.se](#) entry.

On a technical level, the problem is that the previous solution was really closely integrated into the QDialog system by rewiring very specific signals. This is an easy and straightforward way for when the form is shown on a dialog.

However, this way is not flexible to be used in other contexts like when the form is shown on the attribute table. Therefore a new and improved system will be implemented, but that needs to wait for [PR/1370](#) to be merged.

I tried to get the old behavior back by pretending to still support the signals. On current master (thus 2.4 release), the message in your project shows up when closing the dialog and the value is dismissed, but it won't prevent the dialog from closing. It also will not work on the form view in attribute table.

On a related note: constraints introduced by this kind of form logic have never been considered on the attribute table. A constraints (domains) system has been discussed on a previous hackfest. That would allow to proper checking at appropriate times throughout the code.

Currently there is another possibility which I think might even be better suited in terms of usability. The form variable passed to your python init code will emit a signal `attributeChanged` whenever the user types something. So you can use it for checks and show warnings before the user clicks "Ok". This also works on the form view on the attribute table. Modifying your example a bit it then looks like this:

```
def formTest(form, layerid, featureid):
    ....
    form.attributeChanged.connect(validate)

def validate( attr, value ):
    label=myDialog.findChild(QLabel, "label")

    if attr == "name":
        if len( value ) < 4:
            label.setText( "Name is too short. Must have at least 4 characters." )
            nameField.setStyleSheet( "background-color: '#ff9999'" )
        else:
            label.setText( "Well done! Everything is ok." )
            nameField.setStyleSheet( "" )
```

#2 - 2017-05-01 01:08 AM - Giovanni Manghi

- Easy fix? set to No

- Regression? set to No

#3 - 2018-10-15 08:38 AM - Tudor Bărăscu

- Status changed from Open to Closed

Not relevant anymore (at least for me).

Files

