

QGIS Application - Bug report #21376

QGIS crashes when working with QgsLayerTreeLayers in a processing script

2019-02-25 02:07 PM - Jochen Schwarze

Status: Closed	
Priority: High	
Assignee:	
Category: Processing/QGIS	
Affected QGIS version: 3.6.0	Regression?: No
Operating System:	Easy fix?: No
Pull Request or Patch supplied: No	Resolution: invalid
Crashes QGIS or corrupts data: Yes	Copied to github as #: 29193

Description

User Feedback

Trying to reorder layers from a processing script causes QGIS to crash. The following code works from the python console, but causes QGIS to crash when used within a processing script:

```
layer_order = [u'formelsammlung',
               'kabeld',
               'KabelD nicht zuordbar',
               'Erschlossene Flaechen',
               'ax_flurstueck',
               'Flurstuecke erschlossen (KabelD)',
               'Flurstuecke n. eind. erschl.',
               'ax_adressen',
               'Adressen erschlossen (KabelD)',
               'Adressen auf n. eind. erschl. Flst.']]

root = QgsProject.instance().layerTreeRoot()

for lyr_name in layer_order:
    lyr = QgsProject.instance().mapLayersByName(lyr_name)
    if len(lyr) > 0:
        lyr = lyr[0]
        my_lyr = root.findLayer(lyr.id())
        my_clone = my_lyr.clone()
        root.insertChildNode(0, my_clone)
        root.removeChildNode(my_lyr)
```

This is the script that causes the script:

```
from PyQt5.QtCore import QApplication
from qgis.core import QgsProject, QgsProcessingAlgorithm, QgsProcessingParameterFile,
QgsProcessingParameterVectorLayer, QgsProcessing
from qgis.utils import iface

import processing

class BBBBLayerStylerProcessingAlgorithm(QgsProcessingAlgorithm):
    STYLE_FOLDER = 'STYLE_FOLDER'
    INPUT_ADR = 'INPUT_ADR'
    INPUT_FLST = 'INPUT_FLST'

    def tr(self, string):
        return QApplication.translate('Processing', string)

    def createInstance(self):
        return BBBBLayerStylerProcessingAlgorithm()

    def name(self):
```

```

return 'bbbblayerstylercrashtest'

def displayName(self):
    return self.tr('BBBB Layer Styler /Crashtest')

def group(self):
    return self.tr('giswg')

def groupId(self):
    return 'giswg'

def shortHelpString(self):
    return self.tr("Apply styles and order layers from BBBB analysis")

def initAlgorithm(self, config=None):
    self.addParameter(
        QgsProcessingParameterFile(
            self.STYLE_FOLDER,
            self.tr('Style folder'),
            QgsProcessingParameterFile.Folder,
            "",
            'H:\\BBBB\\_styles_test_etc'
        )
    )

    self.addParameter(
        QgsProcessingParameterVectorLayer(
            self.INPUT_ADR,
            self.tr('Input address layer'),
            [QgsProcessing.TypeVectorPoint]
        )
    )

    self.addParameter(
        QgsProcessingParameterVectorLayer(
            self.INPUT_FLST,
            self.tr('Input parcels layer'),
            [QgsProcessing.TypeVectorPolygon]
        )
    )

def processAlgorithm(self, parameters, context, feedback):
    folder = self.parameterAsFile(
        parameters,
        self.STYLE_FOLDER,
        context
    )

    input_adr_lyr = self.parameterAsVectorLayer(
        parameters,
        self.INPUT_ADR,
        context
    )

    input_flst_lyr = self.parameterAsVectorLayer(
        parameters,
        self.INPUT_FLST,
        context
    )

    layer_order = [u'formelsammlung',
        'kabelD',
        'KabelD nicht zuordbar',
        'Erschlossene Flaechen',
        'ax_flurstueck',
        'Flurstuecke erschlossen (KabelD)',
        'Flurstuecke n. eind. erschl.',
    ]

```

```

'ax_adressen',
'Adressen erschlossen (KabelD)',
'Adressen auf n. eind. erschl. Flst.'])

root = QgsProject.instance().layerTreeRoot()

feedback.setProgressText('\norder layers ...')

for lyr_name in layer_order:
    lyr = QgsProject.instance().mapLayersByName(lyr_name)
    if len(lyr) > 0:
        lyr = lyr[0]
        my_lyr = root.findLayer(lyr.id())
        my_clone = my_lyr.clone()
        root.insertChildNode(0, my_clone)
        root.removeChildNode(my_lyr)

feedback.setProgressText('\n----- ALL DONE -----')

return {}

```

Report Details

Crash ID: 7972d0aa46f3682d49822a6e9e6d72944a22cffd

Stack Trace

```

QgsTransformEffect::readProperties :
QgsLayerTreeNode::children :
QgsLayerCapabilitiesModel::trUtf8 :
QMetaCallEvent::placeMetaCall :
QObject::event :
QApplicationPrivate::notify_helper :
QApplication::notify :
QgsApplication::notify :
QCoreApplication::notifyInternal2 :
QCoreApplicationPrivate::sendPostedEvents :
qt_plugin_query_metadata :
QEventDispatcherWin32::processEvents :
CallWindowProcW :
DispatchMessageW :
QEventDispatcherWin32::processEvents :
qt_plugin_query_metadata :
QEventLoop::exec :
QDialog::exec :
PyMethodDef_RawFastCallKeywords :
PyMethodDef_RawFastCallKeywords :
PyEval_EvalFrameDefault :
PyFunction_FastCallDict :
PyMethodDef_RawFastCallDict :
PyObject_Call :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
QMetaObject::activate :
QAbstractItemView::doubleClicked :
QTreeView::mouseDoubleClickEvent :
QWidget::event :
QFrame::event :
QAbstractItemView::viewportEvent :
QCoreApplicationPrivate::sendThroughObjectEventFilters :
QApplicationPrivate::notify_helper :
QApplication::notify :

```

```
QgsApplication::notify :
QCoreApplication::notifyInternal2 :
QApplicationPrivate::sendMouseEvent :
QSizePolicy::QSizePolicy :
QSizePolicy::QSizePolicy :
QApplicationPrivate::notify_helper :
QApplication::notify :
QgsApplication::notify :
QCoreApplication::notifyInternal2 :
QGuiApplicationPrivate::processMouseEvent :
QWindowSystemInterface::sendWindowSystemEvents :
QEventDispatcherWin32::processEvents :
CallWindowProcW :
DispatchMessageW :
QEventDispatcherWin32::processEvents :
qt_plugin_query_metadata :
QEventLoop::exec :
QCoreApplication::exec :
main :
BaseThreadInitThunk :
RtlUserThreadStart :
```

QGIS Info

```
QGIS Version: 3.6.0-Noosa
QGIS code revision: commit:58734527ab
Compiled against Qt: 5.11.2
Running against Qt: 5.11.2
Compiled against GDAL: 2.4.0
Running against GDAL: 2.4.0
```

System Info

```
CPU Type: x86_64
Kernel Type: winnt
Kernel Version: 10.0.17134
```

History

#1 - 2019-02-26 12:46 AM - Nyal Dawson

- Resolution set to invalid
- Status changed from Open to Closed

You need to indicate that your algorithm CANNOT be run in a background thread by adding:

```
def flags(self):
    return super().flags() | QgsProcessingAlgorithm.FlagNoThreading
```