# QGIS Application - Bug report #19560
# Qgis crashes when QgsExtentGroupBox.setOriginalExtent() is called

2018-08-07 03:53 PM - Wondimagegn Beshah

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | High | | |
| **Assignee:** | | | |
| **Category:** | GUI | | |
| **Affected QGIS version:** | 3.2.1 | **Regression?:** | Yes |
| **Operating System:** | Windows 10 | **Easy fix?:** | No |
| **Pull Request or Patch supplied:** | No | **Resolution:** | invalid |
| **Crashes QGIS or corrupts data:** | Yes | **Copied to github as #:** | 27387 |

**Description**

Below is the relevant code. I am using QgsProjectionSelectionWidget to get the CRS.

```
canvas_extent = self.canvas.extent()

transformer = QgsCoordinateTransform(
   self.canvas.mapSettings().destinationCrs(),
   self.input_projection_cbo.crs(),
   QgsProject.instance()
)

transformer.setDestinationCrs(
   self.input_projection_cbo.crs()
)
transformed_extent = transformer.transform(canvas_extent)

self.extent_box.setOriginalExtent(
   transformed_extent,
   QgsCoordinateReferenceSystem(
      self.input_projection_cbo.crs().authid()
   )
)

self.extent_box.setOutputCrs(
   QgsCoordinateReferenceSystem(
      self.input_projection_cbo.crs().authid()
   )
)
self.extent_box.setOutputExtentFromOriginal()
self.extent_box.setCurrentExtent(
   transformed_extent,
   QgsCoordinateReferenceSystem(
      self.input_projection_cbo.crs().authid()
   )

)
```

Below is the entire report.

h2. User Feedback

h2. Report Details

*Crash ID*: 8e827b9e58264385e7a945fbe0ac6df5b4daf9a5

*Stack Trace*
<pre>
QgsExtentGroupBox::setOriginalExtent :
PyInit__gui :
PyCFunction_FastCallDict :
PyObject_GenericGetAttr :
PyEval_EvalFrameDefault :
PyFunction_FastCallDict :
PyObject_CallFunctionObjArgs :
PyObject_Call :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
PyInit_QtCore :
QMetaObject::activate :
QGroupBox::childEvent :
QGroupBox::mouseReleaseEvent :
QgsPresetSchemeColorRamp::clone :
QWidget::event :
QGroupBox::event :
QgsPresetSchemeColorRamp::clone :
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 :
</pre>

*QGIS Info*
QGIS Version: 3.2.1-Bonn
QGIS code revision: commit:1edf372fb8
Compiled against Qt: 5.9.2
Running against Qt: 5.9.2
Compiled against GDAL: 2.2.4
Running against GDAL: 2.2.4

*System Info*
CPU Type: x86_64
Kernel Type: winnt
Kernel Version: 10.0.17134

### History

**#1 - 2018-08-07 03:55 PM - Wondimagegn Beshah**

The crash also happens on QGIS 3.3 master.

**#2 - 2018-09-18 08:16 AM - Nyall Dawson**

*- Status changed from Open to Feedback*

I can't reproduce. Can you share a complete self-contained code sample?

**#3 - 2018-09-21 05:53 AM - Wondimagegn Beshah**

I found out that the problem is not on the above code. The crash is most likely related to me hiding the extent groupbox buttons using Qt methods. I had to delete them because hiding did not work as they still reappeared. The code below used to work on QGIS 2.18 but was not working on QGIS 3.2 and later. Anyways I decided not to hide it due to the crash.

```
for button in self.extent_box.findChildren(QPushButton):
    button.setHidden(True)
    button.parent().setHidden(True)
    button.deleteLater()
    button.parent().deleteLater()
```

**#4 - 2018-09-21 06:44 AM - Nyall Dawson**

*- Resolution set to invalid*
*- Status changed from Feedback to Closed*

Yes - that kind of scripting is always going to be very fragile, and while it may work in certain versions is not part of the stable API contract.