

# QGIS Application - Bug report #13973

## Crash when switching from custom renderer to another

2015-12-10 10:27 AM - Alexander Bruy

<b>Status:</b> Closed	
<b>Priority:</b> High	
<b>Assignee:</b> Martin Dobias	
<b>Category:</b> Python plugins	
<b>Affected QGIS version:</b> master	<b>Regression?:</b> No
<b>Operating System:</b> all	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b> No	<b>Resolution:</b> invalid
<b>Crashes QGIS or corrupts data:</b> Yes	<b>Copied to github as #:</b> 21987
<b>Description</b>	
Subject says it all. To reproduce:	
<ol style="list-style-type: none"><li>1. add custom renderer for vector layer (e.g. one from <a href="#">VectorFieldRenderer</a> plugin or from attached example plugin)</li><li>2. load point layer and select custom renderer for it from layer properties dialog</li><li>3. apply changes and close layer properties dialog</li><li>4. open layer properties dialog again and try to select standard renderer, e.g. Single Symbol</li><li>5. QGIS crashes</li></ol>	
<b>Related issues:</b>	
Related to QGIS Application - Bug report # 14025: Crash when loading project ...	<b>Closed</b> <b>2015-12-22</b>

### Associated revisions

#### Revision 492c1a9f - 2016-01-25 06:16 PM - Martin Dobias

Improve documentation for renderer registry metadata (refs #13973)

### History

#### #1 - 2015-12-22 06:21 AM - Alexander Bruy

- File *renderertest.tar.bz2* added

Ok, I solved issue with crash when switching from custom renderer to standard. But there still get Python error when switching

```
TypeError: invalid result type from MyTestRendererWidget.renderer()
```

I checked, `MyTestRendererWidget.renderer()` returns subclass of the `QgsFeatureRendererV2`, as required by API.

Test plugin updated

#### #2 - 2015-12-22 06:21 AM - Alexander Bruy

- File *deleted (renderertest.tar.bz2)*

#### #3 - 2016-01-19 03:59 AM - Alexander Bruy

- Assignee set to *Martin Dobias*

#### #4 - 2016-01-25 09:20 AM - Martin Dobias

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

The main problem was that the old renderer was used as-is in the test renderer implementation, while it is just a temporary pointer. See the fixed implementation I have attached in #14025 (mytestrenderer.py).

Improved the documentation and closing as invalid...

#### Files

---

renderertest.tar.bz2	4.12 KB	2015-12-22	Alexander Bruy
----------------------	---------	------------	----------------