QGIS Application - Bug report #12749
when using custom functions to style a layer, map canvas gives wrong colors (incoherent with legend)

2015-05-14 06:03 AM - Harrissou Santanna

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	Symbology		
Affected QGIS version:	master	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	end of life
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	20848

Description

Sorry, I don't know how i may title this bug... Hope that I'll explain it well.
I've set a custom function that I apply to a categorized style. The result seems and I set the color to apply to each classes. OK
But when looking to the map canvas, the colors applied to the layer are not coherent with the choice I made. these are inverted count on the layer.
The attached file may better explain. As you can see, there should be no red feature (count =0) but all the features are shown in red.
Tested on windows

History

#1 - 2015-06-03 01:43 PM - Jürgen Fischer

- Category set to Symbology

#2 - 2015-06-17 08:03 PM - Martin Dobias

- Status changed from Open to Feedback

Works fine for me with tests like "$id > 10".

Probably something specific to your custom function... could you please post the function source and the data you used for testing?

#3 - 2015-06-20 08:10 AM - Harrissou Santanna

Sorry, it's not that kind of functions. I wrote a custom function in the function editor. I was learning how this feature works so I wrote this function that retrieves for each feature the number of filled fields (NULL values are set to be shown NULL). Here it is. This code adapted to the console returns right nb for each feature.

```
@qgsfunction(args=0, group='Custom')
def compta(value1, feature, parent):

    nb = 0
    for idx in range(len(feature.attributes())):
        attr = feature.attributes()[idx]
        if not attr == NULL :
            nb = nb +1
    return nb
```

It can be applied on any data sample.
Then, Style menu > Categorized, fill the field box with **$compta**. Classify.

The classification shows the right classes. But when applied, it doesn't apply right colors although the "count features" is right as shown in previous attached file.
Labelling with this function doesn't seem to work properly.

#4 - 2015-12-20 10:15 AM - Giovanni Manghi

- Target version deleted (Future Release - High Priority)
- Status changed from Feedback to Open
- Priority changed from High to Normal

#5 - 2016-02-14 11:08 AM - Sebastian Dietrich

- Affected QGIS version changed from 2.8.2 to master

Confirmed on master (commit:25c289441ef2074826cf27aac6fb49667e8d9a23).

The behaviour is **provider-specific**. It does not occur when using a memory layer.

#6 - 2016-02-14 11:38 AM - Sebastian Dietrich

Seems like a caching issue. When classifying, the actual values of all attributes are fetched and the classification works correctly. However, the *feature* given to the user defined function *compta()* does not have the values filled in, so all features are counted as having no attributes set at all.

Probably an optimization to not fetch lots of stuff not needed for rendering anyway.

#7 - 2016-02-14 12:37 PM - Harrissou Santanna

Thanks Sebastian for digging this.
Could this be related to #14273

#8 - 2016-02-15 01:54 PM - Sebastian Dietrich

Another interesting observation:

If you pass an attribute as a parameter, that attribute's value is available within *feature*.

```
@qgsfunction(args='auto', group='Custom')
def firstAttr(dummy, feature, parent):
    return feature.attributes()[0]
```

Make the expression firstAttr(123) and it always returns NULL when drawing a feature.
Make the expression firstAttr("NameOfFirstAttribute") and it returns the correct value when drawing a feature.

#9 - 2016-02-15 08:59 PM - Nyall Dawson

Sebastian - does it help if you alter
https://github.com/qgis/QGIS/blob/e9ef51341c2cef016648be6c619a4d43ffe362ca/python/core/__init__.py#L43

from
QgsExpression.Function.__init__(self, name, args, group, helptext, usesgeometry)

to
QgsExpression.Function.__init__(self, name, args, group, helptext, usesgeometry, [QgsFeatureRequest.AllAttributes])

?

#10 - 2016-02-16 01:10 PM - Sebastian Dietrich
@Nyall

> *does it help ...*

Yes! The features are rendered correctly after that change.

I wonder if it could be an additional parameter to the @qgsfunction() what attributes the function expects to be filled. E.g.

```
@qgsfunction(args='auto', group='Custom', attrs=['attr1', 'attr2'])
```

would load the values of the attributes named attr1 and attr2.

```
@qgsfunction(args='auto', group='Custom', attrs=True)
```

would load the values of **all attributes**.

Unconditionally loading all attribute values when a user defined function is used in an expression might be a regression for some people with large layers, many attributes and a function that does not use the attribute values at all.

#11 - 2016-02-16 02:28 PM - Nyall Dawson

Sounds reasonable. I'd open a PR and get Nathan to take a look, this is his baby ;)

#12 - 2017-05-01 01:07 AM - Giovanni Manghi
- Easy fix? set to No
- Regression? set to No

#13 - 2019-03-09 04:09 PM - Giovanni Manghi
- Status changed from Open to Closed
- Resolution set to end of life

End of life notice: QGIS 2.18 LTR

Source:
http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/

Files

InvertedStylebetweenLegendandCanvasMap.png	50 KB	2015-05-14	Harrissou Santanna