

QGIS Application - Bug report #4180

rulebasedrendererV2 does not implement symbolForFeature (yet)

2011-08-16 02:20 AM - Richard Duivenvoorde

Status:	Closed	
Priority:	Normal	
Assignee:	Martin Dobias	
Category:	Symbology	
Affected QGIS version:		Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:		Resolution: invalid
Crashes QGIS or corrupts data:		Copied to github as #: 14156
Description		
Currently it is not possible to get the symbol for a feature via the interface 'symbolForFeature(feature)' in rulebasedrendererV2. The other V2 renderers do implement this interface, making it easily possible in python plugins to run through features and ask for it's symbology... it now shows: <pre>QgsSymbolV2* QgsRuleBasedRendererV2::symbolForFeature(QgsFeature& feature) { Q_UNUSED(feature); return mCurrentSymbol; }</pre> not sure if this is by design or just a 'Not Implemented Yet'? For me it would be great to have this implemented...		

History

#1 - 2011-08-16 02:33 AM - Martin Dobias

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

This is because there may be several symbols per one feature and therefore in general one symbol cannot be returned. What you should do instead is to go through the rules of the rule-based renderer and check which rules are valid (and thus which symbols will be used).

There may be also some custom renderers which do not return a valid symbol from symbolForFeature() method and their rendering logic is in renderFeature() method.

(the documentation could be better, I know)

#2 - 2011-08-16 06:33 AM - Richard Duivenvoorde

No sure if I understand correct:

- one feature has one symbol (or not?)
- one symbol can have more symbol layers (each consisting out of a symbol)

- in rulebasedrenderer, if I define two rules for one feature, only the first one is being used??

Please correct me if I'm wrong

#3 - 2011-08-16 07:00 AM - Martin Dobias

One symbol has one or more symbol layers - e.g. railroad symbol may consist of two symbol layers: 1. simple line 2. marker line (creating perpendicular bars)

The basic renderers (single/categorized/graduated symbol renderer) use for drawing one symbol per feature. But in general a custom renderer may render the data in arbitrary way. For example there is a plugin which renders each point as an arrow - the appearance and orientation is determined by attributes - no QgsSymbolV2 classes are employed for drawing, it is all done directly by the renderer.

In rule-based renderer if more rules match one feature then symbols for all matching rules are drawn. Typically there is just one matching rule, but there may be more. The symbols are drawn in the order of their rules.

So in general there is no direct connection between a feature and a symbol. However in common cases with the three basic renderers there is a connection and can be queried using the `symbolForFeature()` method.