

QGIS Application - Feature request #3222

Fast, easy and beautiful on the fly rule-based rendering of OSM maps

2010-11-16 02:15 PM - Mayeul Kauffmann

Status:	Closed	
Priority:	Low	
Assignee:	Tim Sutton	
Category:	Symbology	
Pull Request or Patch supplied:	No	Resolution: fixed
Easy fix?:	No	Copied to github as #: 13282
Description Currently, rule-based rendering of openstreetmap.org-based maps is pretty slow, not easy to achieve (many data-management operation, need to apply patch manually and recompile...) and sometimes give ugly results. It should be fast, easy and beautiful. Fast: to make navigation a fun experience; to allow doing this on old computers/computers without internet access (for users not having access to the slippy map). Easy: because everybody can create new tags of tags, there are thousands of different tags. They will never be displayed all on online-rendered map. There are as many different needs as users. Each user should be able to define what he/she wants to render, and how, based on personal needs and tastes. Beautiful: artefacts, unwanted graphic features and semantic needs that are impossible to meet make ugly maps; a beautiful map is easy to read for the ordinary user, rich and precise for the advanced user. Below are links to related discussion and tickets related to achieving this more global need. #3039 ticket:2832		

History

#1 - 2010-11-16 02:19 PM - Mayeul Kauffmann

I wrote in #2832#comment:8 that I'm working on rendering openstreetmap data in a customized way with filters based directly on the full-length OSM tags. Attached is a screenshot showing the work in progress (near Somma Lombardo, Lombardy, Italy). It is usable with quite large vector files. Symbology must be improved but it is already usable (at least in this area). See attached screenshot.

#2 - 2010-11-21 01:29 PM - Mayeul Kauffmann

Attached below a first release for a set of rules. In addition to visual quality control (on part of Northern Italy) for scales between 1:5,000 and 1:20,000,000, I wrote a script to make some statistical control with an entire country (Italy): For lines, 98% of Italian ways in italy.osm planet file would be symbolized (most of the lines not matched by any rule are unclosed polygons that should normally be rendered as polygons, for instance: "landuse"="farmyard", "natural"="land", etc.). There is still room for progress of course.

I have no statistics on polygons and points yet.

To use the rules, best if you have *four* layers:

1. points
2. polygons
3. lines

4. polygons

Layer 2 (top polygon layer) is used mainly for closed highways (such as roundabout) that the QGIS osm-to-layer converter converts to polygons. Rendering is ugly without this layer in many areas (fun example here: [roundabouts near Malpensa airport](#) I could not have a nice result with only 3 layers).

The lines layer uses symbol levels. It will look ugly without the full patch mentioned here: [#2832#comment:8](#)

Beware: you have to activate symbol levels BEFORE opening the highway_all_rules_v34.qml file. If you modify the rules and save them, you have to add ' symbollevels="1" ' (without the single quotes) after ' <renderer-v2 ' on the forth line of the qml file.

#3 - 2010-11-21 01:34 PM - Mayeul Kauffmann

Additional note: after opening .osm file with the osm-to-layer converter, I converted the layers to .sqlite files (and not to .shp files, because tags are truncated when saving as shp file). The disadvantage is that you do not have spatial index in standard sqlite3 files (I'm not aware of a format that would have this and would not truncate the tags; please comment if you know one).

The advantages are:

1. It works
2. sqlite is one of the fastest database backends for simple select queries.

#4 - 2010-12-09 02:19 PM - Mayeul Kauffmann

I improved somewhat the lines rendering and the points a lot, creating a new set of icons (some of them from scratch). About 95% of the features are rendered for the following areas (planet.osm extracts): Italy, Alps (I think some areas are slightly better rendered than with mapnik or osmarender). Below qml and snapshot.

#5 - 2010-12-19 02:10 PM - Mayeul Kauffmann

I am currently making substantial additions to the rule-based renderer to help achieve this goal.
Currently, I have made the following:

- added Label field, used in legend (and I think in composer)
- added Description field (filter rules are sometimes complex especially because OSM contributors do not always follow the same editing rules; it is helpful to have a field to put comments, etc.)
- make the list of rules tabular (like for categorized or graduated symbols)

I hope to submit a patch soon. Screenshot attached. This would be my first patch for qgis, so any help is welcome ;-)

Mayeul

#6 - 2010-12-22 05:11 PM - Mayeul Kauffmann

I have made extensive additions to many parts of the code to support some useful features to render map with complex symbology grammar. Still, I have problems with the symbol levels. I would like to enable symbol levels when several rules are matched. It would be useful when a feature (specially highways) should be represented with a combination of symbols. For instance, a road can be on a bridge, under a tunnel [with transparent tunnel symbol], in construction, bordered with trees, etc.

In many cases, it is possible to simply combine the symbols, e.g. one symbol for the road itself and another one for the bridge "around" it. It would be

possible to use only one rule for all highways (except motorways if shown with wider symbol).

If we have 30 types of ways and 5 types of additional symbols (bridge, etc), we could then do this with only $30+5=35$ rules instead of $30*5=150$. (Imagine if you add 10 scale-based rules: you got 1500 combinations!!).

For the implementation, I modified the symbolForFeature function (code below) in src/core/symbology-ng/qgsrulebasedrendererv2.cpp

Here is what I'm trying to do:

For the first matched rule for a given feature, the code makes a copy of the rule's symbol. If other rules are matched, it merges the layers of the additional matched rule with the symbol(s) of previously matched rule(s). It also merges the information on symbol levels.

Tests on rendering on small areas (with features matched by 2 rules) works partially:

- the symbols are correctly merged. A new symbol is created and displayed, based on the symbology and levels of the matching rules 8-)
- However, it modifies the symbol associated with the first matched rule and shown in the style sheet (if you open the layer properties, you see that the symbol has been permanently modified): the 1st matched rule is now represented by a combination of the 2 original symbols. This change is saved permanently in the Layer Properties. Similarly for symbol levels. 8-(
- it crashes very often 8-(

Below is the related draft code. I'm sure there are several incorrect thinks here. Anyone could help?

Thanks!

Mayuel

```
[[QgsSymbolV]]2* [[QgsRuleBasedRendererV]]2::symbolForFeature( [[QgsFeature]]& feature )
{
    // return mCurrentSymbol;
    // [[QgsSymbolV]]2* mergedSymbols = [[QgsSymbolV]]2::defaultSymbol( QGis::Line ); // tmp test
    [[QgsSymbolV]]2* mergedSymbols; //FIXME is this correct?
    int nSym = 0; // Number of symbols for a given feature

    for ( QList<Rule*>::iterator it = mCurrentRules.begin(); it != mCurrentRules.end(); ++it )
    {
        Rule* rule = *it;

        if ( rule->isFilterOK( mCurrentFields, feature ) )
        {
            if ( nSym == 0 ){
                ++nSym;
                [[QgsDebugMsg]]( "#####");
                [[QgsDebugMsg]]( "@@ 1st symbol to be merged:" + rule->dump());
                mergedSymbols = rule->symbol(); //FIXME is this correct? is it why the symbol in the rule itself get modified?
            }
            else
            {
                [[QgsDebugMsg]]( "@@ ADDING ADDITIONAL SYMBOL FOR GIVEN FEATURE");
                [[QgsSymbolV]]2* tmpSymbol = rule->symbol();
                // tmpSymbol = rule->symbol();
                [[QgsDebugMsg]]( "num layers tmpSymbol:" + QString::number( tmpSymbol->symbolLayerCount() ) );
                [[QgsDebugMsg]]( "tmpSymbol Rule:" + rule->dump());
                [[QgsDebugMsg]]( "tmpSymbol Pass0 :" + QString::number( tmpSymbol->symbolLayer( 0 )->renderingPass() ) );

                // Merge layers of additional matched rule with symbol(s) of previously matched rules
                [[QgsSymbolLayerV]]2* sl = tmpSymbol->symbolLayer( 0 );

                [[QgsDebugMsg]]( "Highest symbol level : " + QString::number ( (mergedSymbols->symbolLayer(
                (mergedSymbols->symbolLayerCount()) - 1 )->renderingPass() ) );
```

```

// insert symbol matching additional rule only if its first layer is strictly above the last layer of the already merged symbol:
// this is partly to avoid part of the catastrophic consequences of the bug with insertSymbolLayer a few lines below
// and might be changed when this bug is fixed
if(rule->symbol()->symbolLayer( 0 )->renderingPass() <= ((mergedSymbols->symbolLayer( (mergedSymbols->symbolLayerCount()) -1
))->renderingPass())) {QgsDebugMsg("Break"); break;}

[[QgsDebugMsg]]( QString::number(rule->symbol()->symbolLayer( 0 )->renderingPass()));
mergedSymbols->insertSymbolLayer( (mergedSymbols->symbolLayerCount()) , sl ); // probably buggy: modifies the symbol styles as
well!!
//      (mergedSymbols->symbolLayer(1))->setRenderingPass(sl->renderingPass() );
(mergedSymbols->symbolLayer(1))->setRenderingPass( rule->symbol()->symbolLayer( 0 )->renderingPass() );
[[QgsDebugMsg]]( "mergedSymbols layer: " + QString::number(1) + " Pass: " +
    QString::number(mergedSymbols->symbolLayer(1)->renderingPass()) );

// For testing, currently only one layer is added. Adding several layers could be done with some code more or less like:
//      for( int slid = 0 ; slid < tmpSymbol->symbolLayerCount() ; ++slid ){
//          [[QgsSymbolLayerV]]2* sl = tmpSymbol->symbolLayer( slid );
//          mergedSymbols->insertSymbolLayer( slid, sl );
//          (mergedSymbols->symbolLayer(slid))->setRenderingPass(sl->renderingPass() );
//          [[QgsDebugMsg]]( "mergedSymbols layer: " + QString::number(slid) + " Pass: " +
//              QString::number(mergedSymbols->symbolLayer(slid)->renderingPass()) );
//      }

    }

//      return rule->symbol(); //work with levels but takes only first rule
    }
}

[[QgsDebugMsg]]( "num layers mergedSymbols:" + QString::number( mergedSymbols->symbolLayerCount() ) );
[[QgsDebugMsg]]( " mergedSymbols Pass layer0:" + QString::number(mergedSymbols->symbolLayer( 0 )->renderingPass() ) );
[[QgsDebugMsg]]( "@ @ @ @ @ @ return mergedSymbols");
return mergedSymbols;
}

```

#7 - 2010-12-23 08:16 AM - Mayeul Kauffmann

I have just attached a draft patch:
rule_renderer_patch_on_r14936.diff

symbolForFeature is really buggy so it is more if someone wants to have a look at this work in progress. (Still, if s.o. wants to try the working part, it is possible by reverting changes in this function or applying the part of http://trac.osgeo.org/qgis/attachment/ticket/2832/rule_based_renderer.diff which is related to symbolForFeature).

I do not know how to solve the bug I mentioned above. When this will be solved, I plan to do so code-re-reading, documentation and cleanup.

As it is my first qgis patch (and a draft one!) be careful with that!

#8 - 2010-12-27 02:52 PM - Martin Dobias

Hi Mayeul,

thanks for the patch. I had a brief look at it and I like some parts of it :-)

Personally I think it would be good to separate the patch to "safe" and "unsafe" part. Things like support for rule label and description could be committed immediately, however the changes related to symbol levels are unstable and should be given more time to settle down.

Regarding the symbol levels, I do not like the approach with merging of the symbols. I have in mind some improvements to renderer interface that would solve it: instead of calling renderer's `symbolForFeature()` when rendering with symbol levels, the `draw()` method from `[[QgsVectorLayer]]` would let renderer to connect more symbols with a feature. I will try to work on this in the following days/weeks.

In the meanwhile, if you attach a newer version of the patch that leaves out the changes regarding symbol levels, I would be happy to apply it in case there are no problems.

Martin

#9 - 2010-12-28 12:38 PM - Mayeul Kauffmann

Thanks for your comments Martin. I am working on modifying the patch following your suggestion. Still, as a temporary workaround for the unavailability of symbol levels in rule-based renderer, I'm thinking of the following temporary fix:

- if symbol levels are disabled, use all matching rules
- if symbol levels are enabled, use only the first matching rule

Currently, symbol levels are disabled even if you check the symbol levels check box, so the current solution is temporary anyway and this proposed temporary fix will improve this. What do you think?

#10 - 2010-12-28 03:09 PM - Mayeul Kauffmann

I have reversed `symbolForFeature()` (see patch above). I also put comments next to 3 lines:

```
// return mCurrentSymbol; // uncomment this out to disable [[UsingSymbolLevels]]

if( !usingSymbolLevels() ) return mCurrentSymbol; // comment this out to disable [[UsingSymbolLevels]]

r->setUsingSymbolLevels( usingSymbolLevels() ); // comment this out to disable [[UsingSymbolLevels]]
```

Disabling symbollevels will simply draw not map if symbollevel checkbox is checked.

If those 3 lines are left unchanged, symbol levels work correctly (still, only the first matching rule provides the symbol). Rendering without symbol levels work as well.

Anyway, should you apply or not those 3 lines, the result is the same: rendering works nicely and is stable, but trying to turn on or off symbol levels generally crashes qgis:

Checking or unchecking the "Enable symbol levels" checkbox, then clicking "OK" then "Apply" generally crashes qgis.

A workaround is to disable rendering or disable the layer, open the "Layer Properties" Dialog, open the "Symbol Levels" dialog, switch (on or off) the "Enable symbol levels" box, press "OK", then press "OK", then re-enable rendering or re-enable the layer. In this case it seems to be stable. I have no idea how to change the code to avoid that crash without this workaround 8-(

Mayeul

#11 - 2011-01-10 03:41 PM - Mayeul Kauffmann

Hi Martin,

I've just uploaded a new patch: rule_renderer_patch_on_r15004-symbols_not_merged.diff

There are just a few changes on the previous one. It is stable :-))

I could not make qgis crash even with extensive use with a large project (17 layers, a large part of Northern Italy).

In order to let the testers understand why the support for symbol levels is very low, I've added a note to explain that work is in progress and changes will be made later on (as suggested in your comment:11).

Please have a look, thanks!

Mayeul

#12 - 2011-02-20 07:01 AM - Martin Dobias

Finally I have reviewed the patch and applied most of it in . I have modified some parts of code and removed few things which I did not like:

- handling of symbol levels
- last column (Id)
- dark background for some columns

#13 - 2011-02-21 02:11 PM - Mayeul Kauffmann

Hi,

Thanks for this Martin!

About symbol levels, I guess you prefer the more definitive fix you said (in comment:11) you would like to work on. This makes sense. The "last column (Id)" was necessary for my quick fix to sort the rules according to id (since with the quick fix only the first matched rule is rendered, id=priority).

here are a few comments:

1. Making these extensive tests on OSM data gave me the following idea: Sometimes, we need several rules to apply at the same time. There are at least about 15 cases like this with OSM data: attributes related to some of the [road properties](#) (bridge, tunnel, cutting, embankment...) and some of the [road restrictions](#) (access, vehicle, toll, disused...). In most of these cases, overlaying a primary/secondary/tertiary/unclassified road symbol with a specific symbol (bridge, tunnel, embankment,) will give good visual result, while necessitating only few more rules. The idea is, for each rule, to let the user say whether (if this rule is matched) it should be the last rule tested; there could be a check box next to each rule. Typically, the first 15 rules could be rules for road properties and restrictions (those could be combined with other rules, check box would be unchecked), while the next 100 rules would be for road/lines types (they would not be combined with additional rules: check box would be checked). If, say, the 5th rule (among the 100 mentioned above) is matched, then the 95 next rules are ignored [this improves visual quality and has the side effect of saving CPU time]. In the end, we only need 115 rules instead of 1500 rules (15x100, which is the optimistic case that assumes that the 15 first rules are never combined with each other; in fact, they are: there are toll roads with bridges, and very steep roads with embankments, so probably without a way to deal with multiple rules, thousands of combinations are necessary).

2. I guess not applying the few lines of the proposed patch related to symbol levels should have no other effect than disabling symbol levels (with my patch I can enable or disable symbol levels without problems). However, with , opening one of my projects or loading a style (attached [in this ticket here](#)) makes qgis crash (tested with line style). If the style format is incompatible, I would need to recreate my 325 rules/symbols (5 minutes each, that's 27 hours).

Often, it crashes with this message:

Fatal: ASSERT failure in QVector<T>::operator[]: "index out of range", file /usr/include/qt4/QtCore/qvector.h, line 343
Abandon (core dumped)

It even crashes (with the same message) when I open the 'ways' style file saved without the symbol levels (i.e. style saved after unchecking the "Use symbol levels" box).

3. "dark background for some columns" removed: Since the columns are often hidden (by the filter in "group by filter" window or by the scale in the "group by scale" window), I added the background on even columns (2nd and 4th) to help the user see where the columns are. There are grey lines in the "Fields" tab, maybe we can use them as well in the "Style" tab instead of the background?

#14 - 2011-03-21 02:59 PM - Mayeul Kauffmann

I proposed the following patch: patch_on_r15538-rbr_with_symbol_levels.diff

This is a patch against commit:b76c9620 (SVN r15539) adding symbol levels in rule-based renderer (on top of). It is a minimalist patch which adds a quick fix for symbol levels and a "Priority column" (instead of "ID" in previous patch) which is necessary to know what is the, er... priority of rules when symbol levels are used.

#15 - 2011-04-06 05:37 PM - Mayeul Kauffmann

This patch http://trac.osgeo.org/qgis/attachment/ticket/3222/patch_on_r15676-rbr_symbol-levels_reordering_1st-rule_buttons.diff implements most of wonder's suggestion made [here](#), namely "switching first rule/all rules, enabling symbol levels".

It does the following:

- adds a quick fix for symbol levels [use only first rule] and a "Priority column" which is necessary to know in which order the rules will be checked
- the ability to reorder the rules (otherwise, the only option to move a rule to first position is to delete and recreate all rules)
- allows to use only first matching rule (also without using symbol levels), with a new check box
- as it is, using symbol levels forces the use of the "use only first matching rule" behavior. This is reflected in the states of the checkboxes: if the user checks "enable symbol levels", then the "Use only first matched rule" is automatically checked, and grayed out so that it cannot be disabled. This provides feedback to the user that both checkboxes are linked; to make this feedback more clear, the "enable symbol levels" is currently duplicated (and kept synchronized with original checkbox) in the Layer properties of the rule-based renderer (this duplicate checkbox would be removed later, when continuing after first matched rule would be possible with symbol levels).

#16 - 2011-05-03 11:32 PM - Tim Sutton

Hi Mayeul

The patch (http://trac.osgeo.org/qgis/attachment/ticket/3222/patch_on_r15676-rbr_symbol-levels_reordering_1st-rule_buttons.diff) does not apply cleanly because it references paths in your local file system. Can you please generate it from inside the top level of the qgis checkout dir, and against the release branch.

Thanks

Tim

#17 - 2011-12-16 01:55 PM - Giovanni Manghi

- Target version changed from Version 1.7.0 to Version 1.7.4

#18 - 2012-04-15 10:10 AM - Giovanni Manghi

- Target version changed from Version 1.7.4 to Version 2.0.0

#19 - 2012-10-06 02:34 AM - Pirmin Kalberer

- Target version changed from Version 2.0.0 to Future Release - Nice to have

#20 - 2013-03-12 04:23 PM - Martin Dobias

- Resolution set to fixed

- Status changed from Open to Closed

- Pull Request or Patch supplied set to No

I believe this ticket is now obsolete - 1.7 has introduced reworked rule-based rendering with much more flexibility.

Files

OSM_A0_print_Somma_Lombardo_qgis_indexed-col_extract.png	148 KB	2010-11-16	Mayeul Kauffmann
osm_qgis-rules_r1.tar.gz	9.18 KB	2010-11-21	Mayeul Kauffmann
symbols_snapshot_r1.png	252 KB	2010-11-21	Mayeul Kauffmann
symbols_snapshot_r2.png	256 KB	2010-12-09	Mayeul Kauffmann
osm_qgis-rules_r2.tar.gz	13.6 KB	2010-12-09	Mayeul Kauffmann
screenshot_rules_labels_description.png	248 KB	2010-12-19	Mayeul Kauffmann
rule_renderer_patch_on_r14936.diff	32.2 KB	2010-12-23	Mayeul Kauffmann
rule_renderer_patch_on_r14936_symbols_not_merged.diff	28.9 KB	2010-12-28	Mayeul Kauffmann
rule_renderer_patch_on_r14936_symbols_not_merged.2.diff	28.9 KB	2011-01-10	Mayeul Kauffmann
rule_renderer_patch_on_r15004-symbols_not_merged.diff	31 KB	2011-01-10	Mayeul Kauffmann
osm_qgis-rules_r3.zip	17.6 KB	2011-02-21	Mayeul Kauffmann
patch_on_r15538-rbr_with_symbol_levels.diff	5.34 KB	2011-03-21	Mayeul Kauffmann
patch_on_r15676-rbr_symbol-levels_reordering_1st-rule_buttons.diff	18.3 KB	2011-04-06	Mayeul Kauffmann
patch_on_first_1_7_0_symbol-levels_rbr_jef.diff	16.9 KB	2011-05-07	Mayeul Kauffmann