

QGIS Application - Feature request #9907

Add a "stop at first feature" mode to the identify tool

2014-03-27 07:33 AM - Olivier Dalang

Status:	Open	Resolution: Copied to github as #: 18421
Priority:	Normal	
Assignee:		
Category:	Map Canvas	
Pull Request or Patch supplied:		
Easy fix?:	No	
Description		
Hi !		
When in "stop at first" mode, the identify tool stops at first layer, but not at first feature.		
This is annoying when using the identify tool with the "open feature form when a single feature is identified", since when clicking on overlapping features, two features will be identified which make that "identify window" popup appear instead of the clicked feature's form.		
The identify tool should be able to work exactly as the "select single feature" tool.		
Proposed solution :		
<ul style="list-style-type: none">- Add a "stop at first feature" mode to the identify tool- Rename "stop at first" mode to "stop at first layer".		

History

#1 - 2014-03-28 02:03 AM - Matthias Kuhn

The problem with stop at first feature (in contrast to stop at first layer) is, that it is non-deterministic in which order features are returned for a query. As such you might end up with the feature form of the wrong feature.

Instead I would propose to open the attribute table in form view mode with a filter applied to only show the identified features. This way you get the form but are able to switch between multiple identified features. Then we could rename the "open feature form if single feature is identified" to just "open feature form".

#2 - 2014-03-28 02:30 AM - Olivier Dalang

Hi !

Are you sure it's non deterministic ? I get consistent results with the "select single feature" tool. It seems it takes the display order into account (ie a feature that's shown in front of another will be selected).

But in any case, I like your suggestion. The attribute table in form view is indeed much more usable than the "identify window" (at least in my use cases).

One point : it may make lead to some (more) confusion between "identified" features and "selected" features, at such a point one could ask if "identify feature" should not simply work on a feature selection.

The attribute table (when filtered by selection) already reacts in real time to selections, it's just a matter of making the "identify features" window do the same. The only difference between the select tool and the identify tool would then be that the identify tool opens a form window.

#3 - 2014-04-02 05:09 AM - Matthias Kuhn

Olivier Dalang wrote:

Are you sure it's non deterministic ? I get consistent results with the "select single feature" tool. It seems it takes the display order into account (ie a feature that's shown in front of another will be selected).

I guess that's the way a certain provider is implemented. But as long as there is no explicit z-index and order-by there is no guarantee and therefore this may change in a future release/other provider/spatial index boundaries...

But in any case, I like your suggestion. The attribute table in form view is indeed much more usable than the "identify window" (at least in my use cases).

Cool

One point : it may make lead to some (more) confusion between "identified" features and "selected" features, at such a point one could ask if "identify feature" should not simply work on a feature selection.

The attribute table (when filtered by selection) already reacts in real time to selections, it's just a matter of making the "identify features" window do the same. The only difference between the select tool and the identify tool would then be that the identify tool opens a form window.

That's a valid conceptual question and requires careful consideration.

I am a bit afraid, that people will loose their selection (which consists of features carefully selected each on it's own) when they want to check if they should add another feature based on an attribute and therefore use the identify tool.

If we apply a filter to such a view which looks like "object_id in (1,25,37,52)", this would be shown at the bottom of the dialog and could be sufficient to communicate the filtering process to the user.

#4 - 2014-04-02 02:14 PM - Olivier Dalang

Matthias Kuhn wrote:

That's a valid conceptual question and requires careful consideration.

I agree this goes (by far) beyond the scope of this ticket. Still, the more I'm thinking of it, the more I find it makes sense. I'll open a thread about this on the dev ML..

And about your point, it's right, but I think we can overcome this, maybe with a "store selections" feature (I was suggesting this in the Legend interface thread on the mailing list too) or by storing (grouped) selection changes in the undo stack.

#5 - 2017-05-01 12:47 AM - Giovanni Manghi

- Easy fix? set to No