

QGIS Application - Bug report #3517

new labeling engine fails to label self intersecting polygons

2011-02-23 01:33 AM - Mathieu Pellerin - nIRV

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Category:</b>	Labelling	
<b>Affected QGIS version:</b>	master	<b>Regression?:</b> No
<b>Operating System:</b>		<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b> fixed/implemented
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 13577
<b>Description</b>		
<p>The summary says it all. The new labeling engine fails to label self intersecting polygons.</p> <p>The old labeling engine is tolerant to these types of polygon and provides labeling.</p> <p>It actually took me some time to understand why qgis's new labeling engine was refusing to label some polygons from a shapefile I was given.</p>		

History

- #1 - 2011-12-16 09:47 AM - Paolo Cavallini
- Category changed from Symbolology to Labelling
  - Assignee deleted (nobody -)
  - Pull Request or Patch supplied set to No
- #2 - 2011-12-16 01:49 PM - Giovanni Manghi
- Target version changed from Version 1.7.0 to Version 1.7.4
- #3 - 2012-04-16 06:26 AM - Paolo Cavallini
- Crashes QGIS or corrupts data set to No
  - Target version changed from Version 1.7.4 to Version 1.8.0
  - Affected QGIS version set to master
- #4 - 2012-09-04 11:59 AM - Paolo Cavallini
- Target version changed from Version 1.8.0 to Version 2.0.0
- #5 - 2012-09-16 12:26 AM - Giovanni Manghi
- Operating System deleted (Debian)
  - Status info deleted (0)

at least I would consider it a good way to know that I have a self intersecting geometry :)

#6 - 2012-09-16 06:51 PM - Mathieu Pellerin - nIRV

Giovanni, a silent unexplained failure is *not* a good UX in any kind of way :)

I think there could be great benefit in QGIS indicating, either in the labelling dialogue itself or elsewhere, that a layer contains intersecting geometry that will prevent some labels from showing up. That way, your not leaving the user with a "huh, label now showing, black voodoo" but instead a useful hint that

leads said user into fixing issue.

Btw Giovanni, thanks for the efforts your putting into keeping the hub's issues refreshed.

**#7 - 2012-09-17 12:00 AM - Giovanni Manghi**

nirvn - wrote:

| *Giovanni, a silent unexplained failure is not a good UX in any kind of way :)*

of course, I was just joking :) cheers!

**#8 - 2012-09-17 04:30 AM - Gerhard Spieles**

Hi,

see also #2949

It seems to be fixed?

Gerhard

**#9 - 2012-09-17 04:33 AM - Giovanni Manghi**

gespiel - wrote:

| *Hi,*  
  
| *see also #2949*  
| *It seems to be fixed?*  
  
| *Gerhard*

no, it works for lines, but it fails for polygons.

**#10 - 2012-09-17 03:36 PM - Larry Shaffer**

- *Priority changed from Low to Normal*

See also #4079-23

and

<http://osgeo-org.1560.n6.nabble.com/Add-ST-MakeValid-functionality-to-QgsGeometry-td5002449.html>

nirvn - wrote:

| *The old labeling engine is tolerant to these types of polygon and provides labeling.*

The old labeling engine essentially just labeled points. How a point was generated for a prospective label was determined by the user options for the given vector layer type. As noted in the above 'see also', creating a centroid now has the same apparent tolerance for some invalid geometries. So, I think the old and new labeling engines are now on par with each other in this regard.

However, any label where an invalid polygon (not as a point) has to go through PAL to produce some of the newer labeling effects (Free, Perimeter, etc ) will currently silently fail. Adding some means of fixing those geometries, before sending through PAL might alleviate the issue, and allow the feature to be labeled.

*I think there could be great benefit in QGIS indicating, either in the labelling dialogue itself or elsewhere, that a layer contains intersecting geometry that will prevent some labels from showing up. That way, your not leaving the user with a "huh, label now showing, black voodoo" but instead a useful hint that leads said user into fixing issue.*

This is a good idea. The simplest thing would be to add a button in the labeling dialog to run fTools 'Check geometry validity'. Having such a function run automatically on dialog open may cause a long delay (for no apparent reason to the user) if there is a lot of geometry to check, especially if the function is to check all features in a layer, not just those currently visible.

Another option might be to add a 'Notify about unlabeled invalid geometries' checkbox to the labeling dialog (unchecked by default). Then, when features for that layer are encountered that are invalid (at the current zoom level, within the extent and not as a result of extent clipping) the fTools 'Check geometry validity' dialog would pop up listing invalid features found during the current labeling operation. That fTools dialog already has the ability to pan to those features and display the red X. This way, if a user does not see an expected label show up, they have an option to turn on to help find the culprit. The notify dialog would be invalid geometries of the current subset of features being labeled, with possibly a shortcut button to run 'Check geometry validity' for the whole layer.

**#11 - 2013-11-06 09:47 PM - Alex Leith**

- File Capture.JPG added

I just noticed this bug is still about in version 2.0 64 bit on Windows.

It was difficult to discover that it was invalid geometry causing it, and I was just now able to reproduce the missing labels on invalid geometry by drawing a trivial polygon with on self-intersection.

I think that falling back on a 'dodgy' centroid is much preferred over silently failing.

**#12 - 2014-01-30 11:43 PM - Paolo Cavallini**

- Target version changed from Version 2.0.0 to Future Release - High Priority

**#13 - 2014-04-02 03:28 PM - Alvaro Huarte**

I have implemented the **ST\_MakeValid** function for QgsGeometry class to fix the related issues #9861 and #9777. The code is a fork from the original code in postgres implementation with slight changes:

[https://github.com/postgis/postgis/blob/svn-trunk/liblwgeom/lwgeom\\_geos\\_clean.c](https://github.com/postgis/postgis/blob/svn-trunk/liblwgeom/lwgeom_geos_clean.c)

Also, the user can configure automatic validation of the fetched geometries of a layer and convert them to valid when needed.

<https://github.com/qgis/QGIS/pull/1241>

Best regards

Alvaro

**#14 - 2015-11-14 05:04 AM - Giovanni Manghi**

- Status changed from Open to Closed
- Resolution set to fixed/implemented

**#15 - 2015-11-15 12:02 PM - Nyall Dawson**

- Status changed from Closed to Reopened

Giovanni - that PR was never merged, and this is still an issue in master

**#16 - 2015-11-16 04:41 AM - Giovanni Manghi**

- Resolution deleted (fixed/implemented)

Nyall Dawson wrote:

| Giovanni - that PR was never merged, and this is still an issue in master

Hi Nyall, yes I have seen that the PR was not committed, but labelling self intersecting polygons works for me. Maybe there is a better solution (the label do not seems to be over the centroid of the whole geometry) but is better than before (no label) :)

**#17 - 2015-11-17 04:51 AM - Alvaro Huarte**

Hi, I would like to propose an idea in order to avoid these problems derivated from on-the-fly-simplification.

Now, the geometries are simplified just when the data provider fetchs the geometry (<https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrfeatureiterator.cpp#L322>). These simplified geometries can be invalid and there are some problems with it in labeling engine, or expression evaluations that use its geometric attributes, and so on.

For all that, as I said, I would like to propose move the simplification to just where the geometry is converted to be drawn (<https://github.com/qgis/QGIS/blob/master/src/core/symbology-ng/qgsrendererv2.cpp#L317>). I think this is slower but safer than current status.

What do you think?

Best regards

Alvaro

**#18 - 2015-11-17 05:02 AM - Alvaro Huarte**

Sorry, I forgot to say that the simplification executed in provider side remains unchanged. For postgis it executes "ST\_snapToGrid" or "st\_removeRepeatedpoints" functions.

**#19 - 2015-11-17 05:54 AM - Regis Haubourg**

Alvaro Huarte wrote:

*Hi, I would like to propose an idea in order to avoid these problems derivated from on-the-fly-simplification.*

*Now, the geometries are simplified just when the data provider fetchs the geometry ( <https://github.com/qgis/QGIS/blob/master/src/providers/ogr/qgsogrfeatureiterator.cpp#L286>). These simplified geometries can be invalid and there are some problems with it in labeling engine, or expression evaluations that use its geometric attributes, and so on.*

*For all that, as I said, I would like to propose move the simplification to just where the geometry is converted to be drawn ( <https://github.com/qgis/QGIS/blob/master/src/core/symbology-ng/qgsrendererv2.cpp#L317>). I think this is slower but safer than current status.*

*What do you think?*

*Best regards*

*Alvaro*

Hi, that sounds reasonable. Nyall, Martin, opinions?

**#20 - 2015-11-17 11:05 AM - Nyall Dawson**

*For all that, as I said, I would like to propose move the simplification to just where the geometry is converted to be drawn*

I'm not a fan of this approach. I think it's great how simplification also aids labelling at the moment. For instance, without simplification the "free" and "horizontal" label placements are so slow as to be nearly unusable. With simplification, they are fast enough to be used interactively.

I'd rather see the approach where liblwgeom is used to repair broken geometries explored.

**#21 - 2015-11-17 11:25 PM - Alvaro Huarte**

Nyall Dawson wrote:

*I'm not a fan of this approach. I think it's great how simplification also aids labelling at the moment. For instance, without simplification the "free" and "horizontal" label placements are so slow as to be nearly unusable. With simplification, they are fast enough to be used interactively.*

Hi, another alternative could be to add the "SnapToGrid" algorithm (See liblwgeom: <https://github.com/postgis/postgis/blob/svn-trunk/liblwgeom/ptarray.c#L1798>) and check results. It is pretty simple and feasible integrate it in <https://github.com/qgis/QGIS/blob/master/src/core/qgsmaptopixelgeometrysimplifier.cpp#L254> defining a new "algorithm" enum (possibly configurable).

**#22 - 2015-11-18 12:29 AM - Nyall Dawson**

That sounds like a better approach. I also think snap to grid simplification could be useful for composer exports... Eg, geometries could be simplified to a grid which is twice dpi, to remove redunant points from the resultant svg/PDF files.

**#23 - 2015-11-18 01:00 AM - Alvaro Huarte**

I think I might try adding this new algorithm to test results (I will try to get time). It is complementary to the ST\_makeValid implementation commented in other dedates.

#24 - 2015-11-18 04:11 AM - Alvaro Huarte

Hi, I have created a new pull <https://github.com/qgis/QGIS/pull/2483> to test results using the "SnapToGrid" algorithm by default.

It is a try to resolve the issue.

Alvaro

#25 - 2016-06-16 02:29 AM - Mathieu Pellerin - nIRV

- File *evidence.png* added
- Status changed from *Reopened* to *Closed*
- Resolution set to *fixed/implemented*

I've tried labelling an simple intersecting polygon today, it worked (to my surprise actually). Closing, fixed.

Files

qgis-labeling-missing.jpg	74.4 KB	2011-02-23	Mathieu Pellerin - nIRV
qgis-labeling-ok.jpg	75.1 KB	2011-02-23	Mathieu Pellerin - nIRV
Capture.JPG	30.3 KB	2013-11-06	Alex Leith
evidence.png	26.4 KB	2016-06-16	Mathieu Pellerin - nIRV