QGIS Application - Bug report #9655
Strange behavior of labeling for multipolygons, due to geometry simplification

2014-02-25 08:59 AM - Augustin Roche

Status:	Closed		
Priority:	High		
Assignee:	Alvaro Huarte		
Category:	Labelling		
Affected QGIS version:	2.2.0	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	Yes	**Resolution:**	fixed/implemented
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	18223

Description

For labeling multipolygons it seems that sometimes the label is placed on rather small parts of the shape, whereas there is plenty of room to place the label elsewhere. In the attached example, islands are labeled instead of mainland.
In some cases, the label is not displayed at all for the whole shape, whereas there is obviously enough room to place the label.
The behavior depends on the scale and what part of the feature is displayed.

Many Thanks

Related issues:

Related to QGIS Application - Bug report # 9673: some polygons not labelled w...	**Closed**	**2014-02-27**
Related to QGIS Application - Feature request # 9521: Introduce support for "...	**Closed**	**2014-02-09**

History

#1 - 2014-02-25 06:02 PM - Larry Shaffer

- Assignee set to Larry Shaffer

- Target version set to Version 2.4

Hi Augustin,

Can you please share the data and project?

Is this with **Label every part of multi-part features** checked (under Labels->Rendering)?

> *In some cases, the label is not displayed at all for the whole shape, whereas there is obviously enough room to place the label.*

Occasionally, when features are clipped to the extent (during the current labeling process), their geometries become invalid and are then not labeled.

#2 - 2014-02-26 03:03 AM - Augustin Roche

- File data_test.zip added

Here is sample data that shows the problem.

#3 - 2014-02-26 08:34 AM - Larry Shaffer

It seems to be an issue with the new geometry simplification. Under Vector Layer Properties -> Rendering, uncheck **Simplify geometry**. The layer is then

labeled correctly here, with the largest geometry getting the label (unless **Label every part of multi-part features** is checked, of course).

Please verify my results.

It seems that during on-the-fly geometry simplification your coastlines return invalid geometries, which will not be labeled.

While turning off geometry simplification avoids the issue, it is still a bug that needs addressed.

#4 - 2014-02-26 08:36 AM - Larry Shaffer

- Priority changed from Normal to High

- Subject changed from Strange behavior of labeling for multipolygons to Strange behavior of labeling for multipolygons, due to geometry simplification

#5 - 2014-02-26 08:39 AM - Larry Shaffer

Hi Alvaro,

I added you as a watcher to this issue, since I believe it is directly related to the new geometry simplification.

Regards,

Larry

#6 - 2014-02-26 08:51 AM - Augustin Roche

Indeed turning off on-the-fly simplification solves the labeling problems.
Many thanks,
Augustin

#7 - 2014-02-26 10:26 AM - Alvaro Huarte

Larry Shaffer wrote:

> *Hi Alvaro,*
>
> *I added you as a watcher to this issue, since I believe it is directly related to the new geometry simplification.*
>
> *Regards,*
>
> *Larry*

Hi Larry, I can try to solve this issue, are you agree ?

#8 - 2014-02-26 10:38 AM - Larry Shaffer

- Assignee changed from Larry Shaffer to Alvaro Huarte

Alvaro Huarte wrote:

> *Hi Larry, I can try to solve this issue, are you agree ?*

Yes, please do. I assigned the issue to you. I'm pretty busy with work on labeling unit tests, so don't really have time to review/learn your simplification code now.

Thanks!

Ultimately, and for 2.3, we should seriously look at including **liblwgeom** support, or something similar, to introduce on-the-fly fixing of invalid geometry. This has been discussed multiple times in the past.

There is also the prepair lib, as discussed previously. However, it requires an additional dependency on CGAL.

Such a feature will also help with labeling all invalid geometries, not just those possibly resulting from simplification, etc., as well as general fixing of invalid geometries.

#9 - 2014-02-26 10:45 AM - Alvaro Huarte

Larry Shaffer wrote:

> *Alvaro Huarte wrote:*
>
> > *Hi Larry, I can try to solve this issue, are you agree ?*
>
> *Yes, please do. I assigned the issue to you. I'm pretty busy with work on labeling unit tests, so don't really have time to review/learn your simplification code now.*
>
> *Thanks!*
>
> *Ultimately, and for 2.3, we should seriously look at including **liblwgeom** support, or something similar, to introduce on-the-fly fixing of invalid geometry. This has been discussed multiple times in the past.*
>
> *There is also the prepair lib, as discussed previously. However, it requires an additional dependency on CGAL.*
>
> *Such a feature will also help with labeling all invalid geometries, not just those possibly resulting from simplification, etc., as well as general fixing of invalid geometries.*

Ok, I study it!

#10 - 2014-02-26 01:14 PM - Larry Shaffer

Hi Alvaro,

Of note: liblwgeom is part of PostGIS, but is basically available as a separate package for the major QGIS platforms. It would be good to utilize it without incurring a additional dependency upon PostGIS/PostgreSQL.

OSGeo4W
Ubuntu
Mac Homebrew

It may be reasonable to just embed its source within QGIS as well, then just publicly expose its functionality via **QgsGeometry** functions.

#11 - 2014-02-26 03:24 PM - Alvaro Huarte

Hi Larry, I fixed an error in the simplification algorithm. It created invalid polygons with two points:
https://github.com/qgis/QGIS/pull/1219

It is not related with the bad calculation of label position.
This error is throwed in 'QgsPalLabeling::drawLabeling()' method, it is very complex and I need some extra time to fix!

Alvaro

#12 - 2014-02-26 03:33 PM - Alvaro Huarte

Alvaro Huarte wrote:

> *Larry Shaffer wrote:*
>
> > *Alvaro Huarte wrote:*
> >
> > > *Hi Larry, I can try to solve this issue, are you agree ?*
> >
> > *Yes, please do. I assigned the issue to you. I'm pretty busy with work on labeling unit tests, so don't really have time to review/learn your simplification code now.*
> >
> > *Thanks!*
> >
> > *Ultimately, and for 2.3, we should seriously look at including **liblwgeom** support, or something similar, to introduce on-the-fly fixing of invalid geometry. This has been discussed multiple times in the past.*
> >
> > *There is also the prepair lib, as discussed previously. However, it requires an additional dependency on CGAL.*
> >
> > *Such a feature will also help with labeling all invalid geometries, not just those possibly resulting from simplification, etc., as well as general fixing of invalid geometries.*
>
> *Ok, I study it!*

Hi Larry, I agree with append MakeValid-functionality-to-QgsGeometry but IMHO it should not be applied after simplification because of it will be very slow to use in a rendering task, I think better fix the calculation of centroid of these invalid geometries.

#13 - 2014-02-26 04:28 PM - Larry Shaffer

Alvaro Huarte wrote:

> *Hi Larry, I fixed an error in the simplification algorithm. It created invalid polygons with two points:*
> *https://github.com/qgis/QGIS/pull/1219*
>
> *It is not related with the bad calculation of label position.*
> *This error is throwed in 'QgsPalLabeling::drawLabeling()' method, it is very complex and I need some extra time to fix!*

What error? I know the workings of that class pretty well.

The calculation of the centroid is currently in the PAL lib. However, there is an issue ticket for adding a better centroid solution that ensures the "label is always on top of the feature":#9480.

There will be a lot of changes upcoming in the QgsPalLabeling class. Including an attempt to fix geometries on-the-fly. But, if you wish to tackle the "better centroid" (issue #9480) for PAL, that would be great. Realize however, if the geometry is invalid, then it will never be registered with PAL to begin with. This doesn't mean that invalid geometries can't be used to create valid centroids, though. That is one approach for polygons, *before* registering with PAL.

#14 - 2014-02-26 04:34 PM - Alvaro Huarte

Larry Shaffer wrote:

> *What error? I know the workings of that class pretty well.*

I meant that it miscalculates the centroid when the geometry is invalid.
Thanks for your notes, I am testing QgsPalLabeling class.

#15 - 2014-03-01 04:02 AM - Giovanni Manghi

> *There is also the prepair lib, as discussed previously. However, it requires an additional dependency on CGAL.*

see also #9521

#16 - 2014-03-01 07:18 AM - Alvaro Huarte

PAL library already has a centroid algorithm forced within of the polygon.

It is assigned when the 'placement' property of the label configuration is equals to 'FREE'.
https://github.com/qgis/QGIS/blob/master/src/core/pal/feature.cpp#L1034

I want find where the PAL library fails using polygons with self-intersects edges.
Alvaro

#17 - 2014-03-01 08:27 AM - Jürgen Fischer

- Target version changed from Version 2.4 to Future Release - High Priority

#18 - 2014-03-01 09:28 AM - Larry Shaffer

Giovanni Manghi wrote:

> > *There is also the prepair lib, as discussed previously. However, it requires an additional dependency on CGAL.*
>
> *see also #9521*

Hi, Giovanni. The 'prepair' external library (fixes geometry) is different than 'prepared' GEOS geometry (which appears to be caching optimization in nature). However, the latter looks very interesting, indeed.

#19 - 2014-03-02 09:39 PM - Alvaro Huarte

- Resolution set to fixed/implemented

- Pull Request or Patch supplied changed from No to Yes

I have changed how resolve the issue. I use 'buffer' method to convert the geometry to valid, and it works better

~~I added to pal::Layer a new 'ignoreInvalidGeometries' property to enable labeling of invalid geometries.~~
~~I works fine, but it requires modify the code of PAL Labeling library (http://pal.heig-vd.ch).~~

See:
~~https://github.com/ahuarte47/QGIS/commit/ad1e07c9e697db34c5ebc2b2c7c921a071a05dc1~~ in
https://github.com/qgis/QGIS/pull/1219

#20 - 2014-03-03 01:40 PM - Larry Shaffer

- Resolution deleted (fixed/implemented)

Hi Alvaro,

Your pull request doesn't fix the issue in my testing, using the provided sample project. (See other comment in request.)

With **Simplify geometry** ON, I do not get good results, e.g. if *any* extent clipping happens to the noted problematic simplified feature, it is never labeled. This includes only clipping off an island, which should leave a valid geometry. I think there is something else going on here, stemming from simplification of multi-geometry features (maybe?).

#21 - 2014-03-03 03:04 PM - Alvaro Huarte

Larry Shaffer wrote:

> *Hi Alvaro,*
>
> *Your pull request doesn't fix the issue in my testing, using the provided sample project. (See other comment in request.)*
>
> *With **Simplify geometry** ON, I do not get good results, e.g. if any* extent clipping happens to the noted problematic simplified feature, it is never labeled. This includes only clipping off an island, which should leave a valid geometry. I think there is something else going on here, stemming from simplification of multi-geometry features (maybe?).

Hi Larry, I have changed how resolve the issue. I use 'buffer' method to convert the geometry to valid, and it works better

#22 - 2014-03-08 03:13 PM - Larry Shaffer

- Status changed from Open to Closed

- Resolution set to fixed/implemented

Hi Augustin,

Should be fixed with commit commit:245422a. Tested here on Mac OS X 10.7.5 and 10.9.2 with your test data. If not please reopen issue.

Files

labelling_multipolygon_qgis2.2.0.jpg	45.3 KB	2014-02-25	Augustin Roche
data_test.zip	14.3 KB	2014-02-26	Augustin Roche