

QGIS Application - Bug report #12406

QgsDistanceArea.measure(geometry) - for Polygons in WGS84

2015-03-19 04:04 AM - Jens Deutschmann

Status:	Closed	
Priority:	Normal	
Assignee:		
Category:		
Affected QGIS version:	2.8.0	Regression?: No
Operating System:	Windows	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: fixed/implemented
Crashes QGIS or corrupts data:	No	Copied to github as #: 20579
Description This is since Version 2.8 (newer i dont know). The Method measures something but the Nummber is extremly tiny. And there for wrong! If you project it to a CoorDinateSystem somewhere in the world it would measure something nearly correctly (because of the incorrect projection). But this could not be a workaround... if you need the whole world.		
Related issues: Related to QGIS Application - Bug report # 12057: Computed area is wrong when... Closed 2015-01-26		

Associated revisions

Revision 19c1dc69 - 2015-05-21 01:48 PM - Martin Dobias

Fix #12406 (measured area is wrong when computed on ellipsoid)

Introduced in e568493

Revision 1949fe9c - 2015-05-21 02:24 PM - Martin Dobias

Fix #12406 (measured area is wrong when computed on ellipsoid)

Introduced in e568493

History

#1 - 2015-03-19 04:16 AM - Jens Deutschmann

I forgot to highlight that this occurs in my Python-Script (Plugin), dont know if its happaning in the FieldCalculator too. Its only a question of time to figure it out, sry.

#2 - 2015-03-19 09:04 PM - Martin Dobias

Could you paste a snipped of code where you use QgsDistanceArea? I assume you didn't enable ellipsoid mode, so the measurements you get are using Euclidean distance on your coordinates. You probably want to do something like this:

```
d = QgsDistanceArea()
d.setEllipsoidalMode(True)
d.setEllipsoid("WGS84")
d.measure(QgsGeometry.fromPolyline([QgsPoint(0,0),QgsPoint(1,0)]))
```

#3 - 2015-03-20 02:27 AM - Jens Deutschmann

Here is a snippet, but I don't think it's that problem because in QGIS 2.6 it all works well! And I think I enabled it this way....

```
def analyzeGeometry(self, geometry, layer, info):
    crs = QgsCoordinateReferenceSystem(layer.dataProvider().crs())
    calculator = QgsDistanceArea()
    calculator.setSourceCrs(crs)
    calculator.setEllipsoid(crs.ellipsoidAcronym())
    calculator.setEllipsoidalMode(crs.geographicFlag())
    ....
    ....
    elif geometry.type() == Qgs.Polygon:
        self.add(info, 'num_polygons', 1)
        self.add(info, 'tot_poly_area', int(calculator.measure(geometry)/1000000))
        self.add(info, 'tot_poly_perimeter', int(calculator.measurePerimeter(geometry)/1000))
```

#4 - 2015-03-25 06:24 AM - Martin Dobias

The code snippet above unfortunately does not help much without knowing the parameters/outputs.

Please try to provide a sample of CRS configuration + geometry + expected vs actual results.

#5 - 2015-05-10 01:03 AM - Giovanni Manghi

- Target version changed from Version 2.8 to Version 2.8.2

#6 - 2015-05-11 12:53 AM - Giovanni Manghi

- Status changed from Open to Feedback

#7 - 2015-05-21 04:52 AM - Martin Dobias

- Status changed from Feedback to Closed

Fixed in changeset commit:"19c1dc69e9ff09cbb4323cb73a64c7cd805dc63f".

#8 - 2015-05-21 05:34 AM - Martin Dobias

Backported to 2.8 in commit:1949fe9c86215e7818c9b21e14c80d4e417c2d7a

#9 - 2015-05-22 01:46 AM - Giovanni Manghi

- Resolution set to fixed/implemented