

QGIS Application - Feature request #16418

Geometry validity inconsistency between tools

2017-04-06 11:52 PM - Giovanni Allegri

Status:	Closed	Resolution: Copied to github as #: 24327
Priority:	High	
Assignee:		
Category:	Geometry	
Pull Request or Patch supplied:	No	
Easy fix?:		
Description		
<p>The Topology Validation tool and Check Validity algorithm return different results for self-touching polygons which define a (false) hole. The difference is happening because Check Validity uses QgsGeometryValidator which calls GEOSisValidDetail_r with the GEOS GEOSVALID_ALLOW_SELFTOUCHING_RING_FORMING_HOLE flag.</p> <p>This kind of geometries are invalid for OGC, but they're ok for others (e.g. for Esri). I think QGIS should adhere to OGC strict rules to be consistent with other OGC compliant sw like Spatialite, PostGIS, etc.</p> <p>The following polygon is an example where Topology Validator, Spatialite and PostGIS agree to be invalid, while it's ok for QgsGeometryValidator:</p> <p>POLYGON ((200 400, 400 400, 400 200, 300 200, 350 250, 250 250, 300 200, 200 200, 200 400))</p> <p>QgsGeometryValidator should not set the flag by default and leave it as an option to relax validation.</p>		

Associated revisions

Revision 4e04d022 - 2019-02-26 09:05 PM - Nyal Dawson

By default, validity check should treat ring self intersections as invalid

We use the OGC definition of validity to ensure consistent results with PostGIS, GDAL, etc

Fixes #16418, fixes #21336

Revision 49742c30 - 2019-02-26 09:05 PM - Nyal Dawson

[needs-docs][processing] Add option to check validity alg to ignore self-intersection causing rings errors

By default the algorithm now uses the strict OGC definition of polygon validity, where a polygon is marked as invalid if a self-intersecting ring causes an interior hole.

If the "Ignore ring self intersections" option is checked, then this rule will be ignored and a more lenient validity check will be performed.

Refs #16418, refs #21336

Revision 25a42a9f - 2019-03-05 02:08 AM - Nyal Dawson

By default, validity check should treat ring self intersections as invalid

We use the OGC definition of validity to ensure consistent results
with PostGIS, GDAL, etc

Fixes #16418, fixes #21336

(cherry picked from commit 4e04d02293637923e065afba3f3b962249a9ca78)

Revision 850905e7 - 2019-03-05 02:08 AM - Nyal Dawson

[needs-docs][processing] Add option to check validity alg to ignore self-intersection
causing rings errors

By default the algorithm now uses the strict OGC definition of polygon validity, where
a polygon is marked as invalid if a self-intersecting ring causes an interior hole.
If the "Ignore ring self intersections" option is checked, then this rule will be
ignored and a more lenient validity check will be performed.

Refs #16418, refs #21336

(cherry picked from commit 49742c302afb97e6272ce8b6b3a806014cdbdb5b)

Revision 49798918 - 2019-03-07 03:02 AM - Nyal Dawson

By default, validity check should treat ring self intersections as invalid

We use the OGC definition of validity to ensure consistent results
with PostGIS, GDAL, etc

Fixes #16418, fixes #21336

(cherry picked from commit 4e04d02293637923e065afba3f3b962249a9ca78)

Revision 4cd9c110 - 2019-03-07 03:02 AM - Nyal Dawson

[needs-docs][processing] Add option to check validity alg to ignore self-intersection
causing rings errors

By default the algorithm now uses the strict OGC definition of polygon validity, where
a polygon is marked as invalid if a self-intersecting ring causes an interior hole.
If the "Ignore ring self intersections" option is checked, then this rule will be
ignored and a more lenient validity check will be performed.

Refs #16418, refs #21336

(cherry picked from commit 49742c302afb97e6272ce8b6b3a806014cdbdb5b)

(cherry picked from commit 66cd7e93024f432a9ebd98258dee74ac49834d1f)

History

#1 - 2017-05-01 12:44 AM - Giovanni Manghi

- *Easy fix?* set to *No*

#2 - 2019-02-26 09:04 PM - Nyal Dawson

- *% Done* changed from *0* to *100*

- *Status* changed from *Open* to *Closed*

Applied in changeset commit:qgis|4e04d02293637923e065afba3f3b962249a9ca78.