

Create or download a Spatialite test database

1. Download OSM data from <http://download.geofabrik.de/>, for example data from Hamburg <http://download.geofabrik.de/europe/germany/hamburg-latest.osm.pbf>
2. Acquire GDAL 1.10, for example from gisinternals as a zip file <http://gisinternals.com/sdk/PackageList.aspx?file=release-1600-gdal-mapserver.zip>
3. Open Windows command window and set the environment in GDAL home directory by running "SDKShell.bat".
4. Convert OSM data into Spatialite format with command (write it to the command line if the result with copy-paste is only 3.5 MB initialized but empty database file. Some invisible characters may cause troubles with copy-paste).

```
ogr2ogr -f SQLite -dsco spatialite=yes hamburg.sqlite hamburg-latest.osm.pbf -progress -gt 20000 --config OGR_SQLITE_SYNCHRONOUS OFF --config OSM_COMPRESS_NODES YES
```
5. Or just download a readymade Spatialite file (2013-04-08) from <http://latuviitta.org/documents/hamburg.zip>.

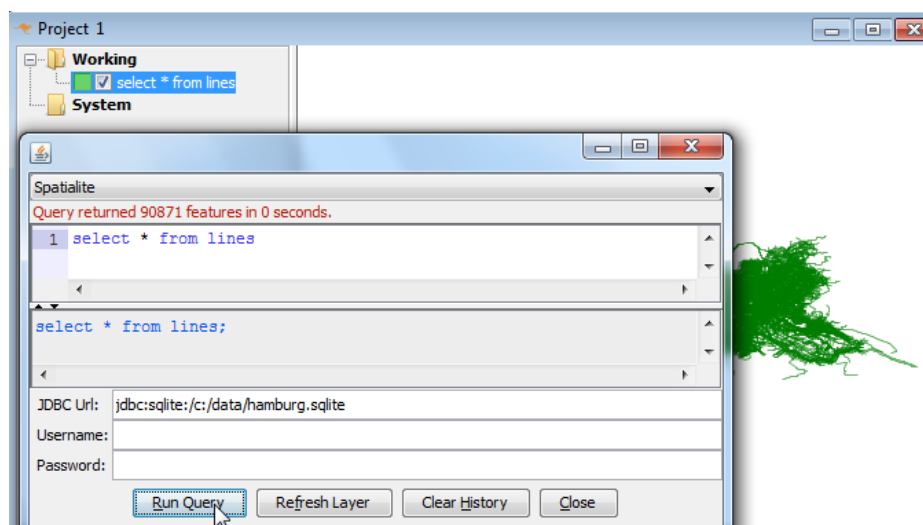
Install DB Query Plugin with Spatialite 4.0 support

1. Download the plugin from <http://netcologne.dl.sourceforge.net/project/jumpdbqplugin/jumpdbqplugin/jumpdbqplugin-0.8.1/jumpdbplugin-0.8.1.zip>
2. Download Spatialite dll files and jdbc driver from http://latuviitta.org/documents/Spatialite_4.0_test_with_jre_1.6.zip.
3. Unzip the plugin, 10 dll files and sqlite-jdbc-3.7.15-SNAPSHOT.jar into lib/ext
4. Start OpenJUMP

Test basic SQLite and advanced Spatialite functions

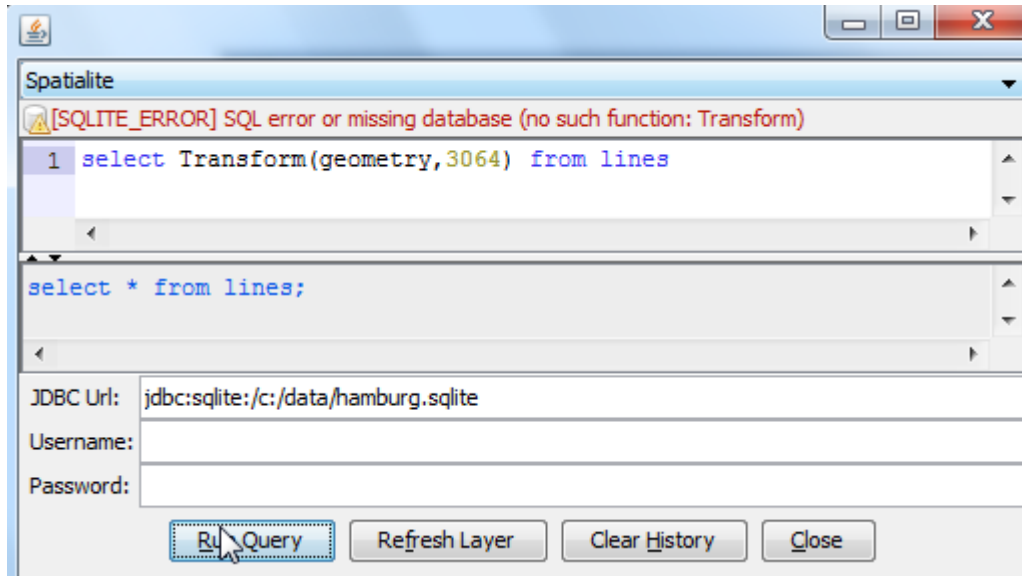
1. Select Tools – Database query, set jdbc connection string for the Spatialite1 connection and run SQL query

SELECT * from lines



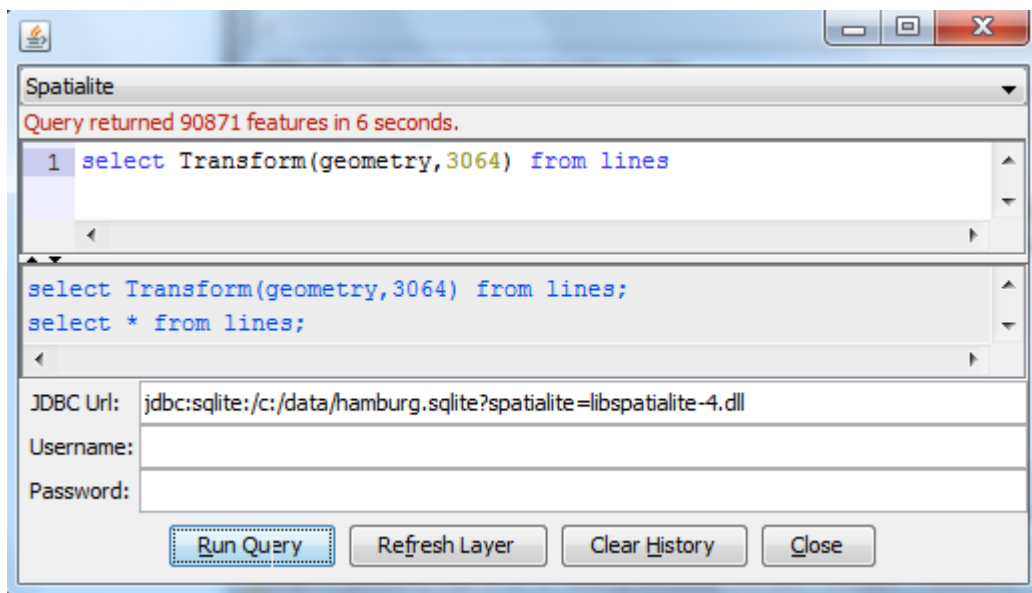
This means that basic jdbc reading works.

2. Try something that requires Spatialite, for example
SELECT Tranforms(geometry, 3064) from lines

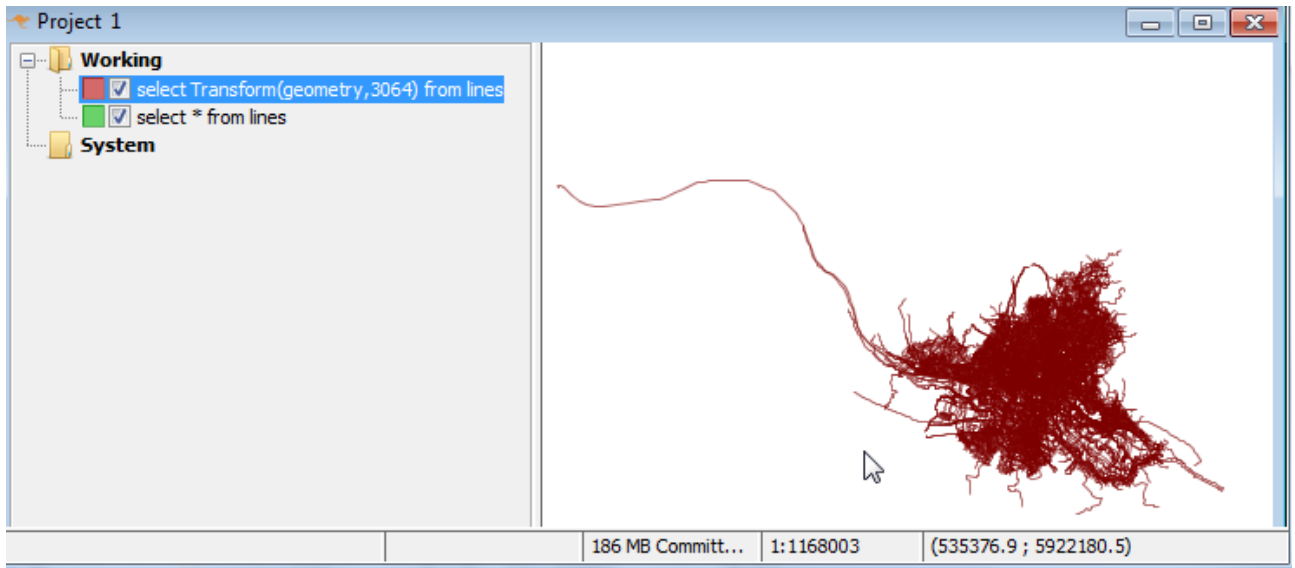


This fails because Transform is a Spatialite function and we do not have Spatialite.

3. Add instruction for loading Spatialite into the JDBC Url and repeat the previous test.
- 4.



5. Zoom to the new layer and observe that the coordinates are now UTM coordinates (but perhaps EPSG:3064 is not the most suitable for these data
<http://www.spatialreference.org/ref/epsg/3064/>)



Once we have Spatialite extension in use we have everything that is supported by Spatialite also available in OpenJUMP. The list of functions <http://www.gaia-gis.it/gaia-sins/spatialite-sql-4.0.0.html>

For example, compress linestring into points by selecting their centroids AND convert the result directly into EPSG:3064.

