CHAPTER 18

# Answer Sheet

## 18.1 Results for *Adding Your First Layer*

### 18.1.1 *Preparation*

You should see a lot of lines, symbolizing roads. All these lines are in the vector layer that you just loaded to create a basic map.

*Back to text*

## 18.2 Results for *An Overview of the Interface*

### 18.2.1 *Overview* (Part 1)

Refer back to the image showing the interface layout and check that you remember the names and functions of the screen elements.

*Back to text*

### 18.2.2 *Overview* (Part 2)

1. *Save as*
2. *Zoom to layer*
3. *Help*
4. *Rendering on/off*
5. *Measure line*

*Back to text*

## 18.3 Results for *Working with Vector Data*

### 18.3.1 *Shapefiles*

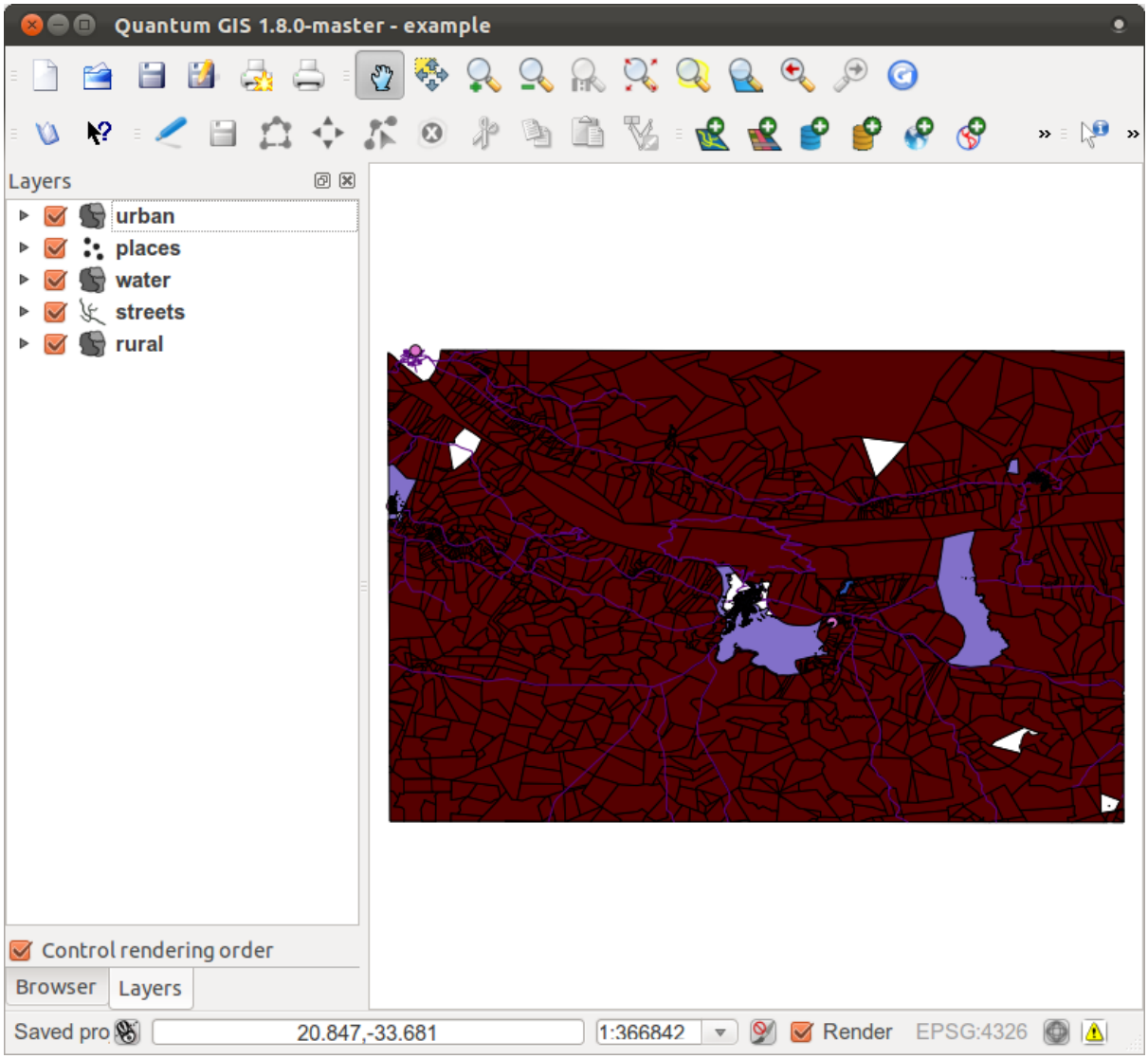
There should be two layers on your map:

- *places* and
- *streets*.

*Back to text*

### 18.3.2 *Databases*

All the vector layers should be loaded into the map. It probably won't look nice yet:



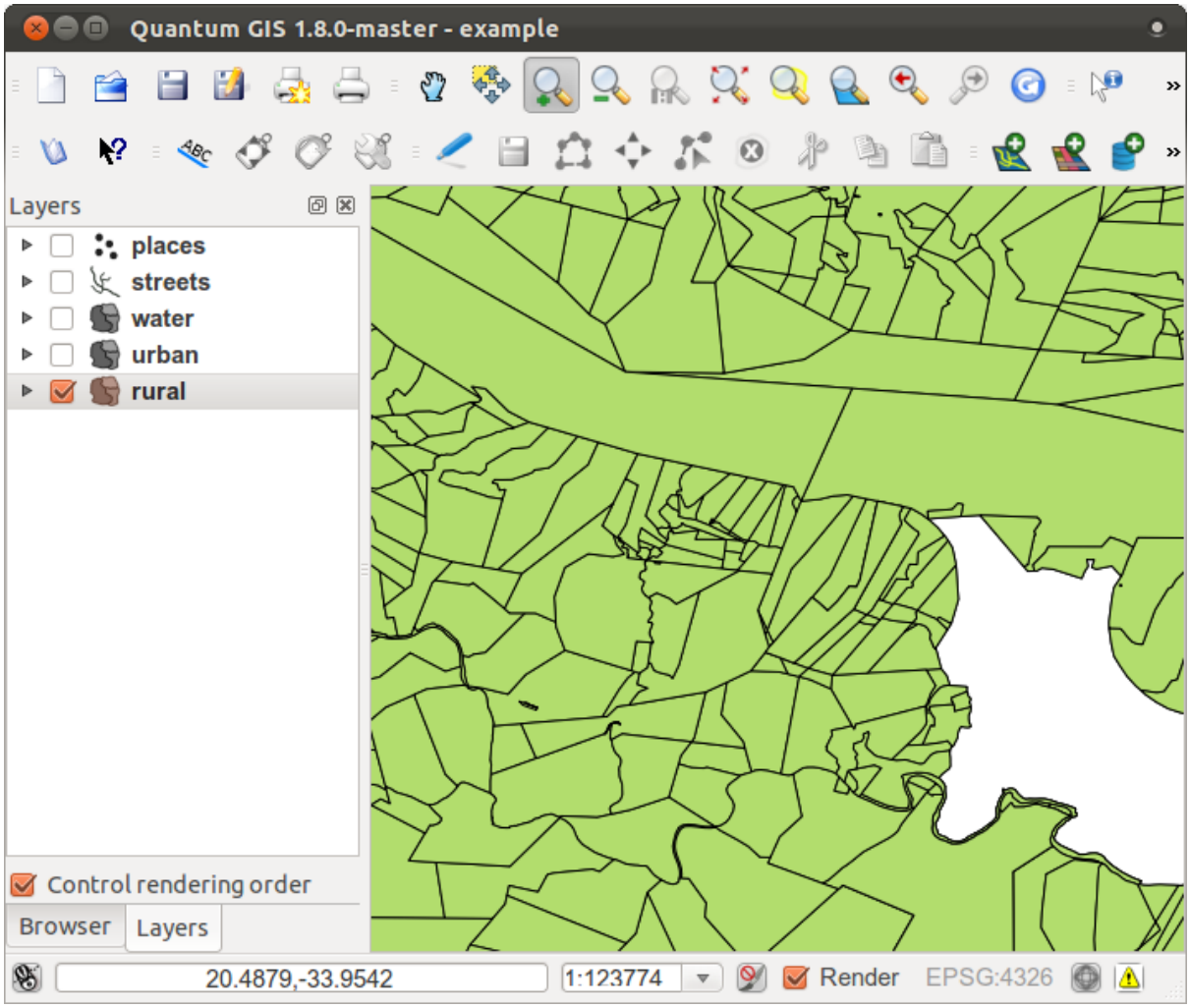

(We'll fix the ugly colors later.)

*Back to text*

## 18.4 Results for *Symbology*

### 18.4.1 *Colors*

- Verify that the colors are changing as you expect them to change.
- It is enough to change only the *rural* layer for now. An example is below, but may look different depending on the color you chose.

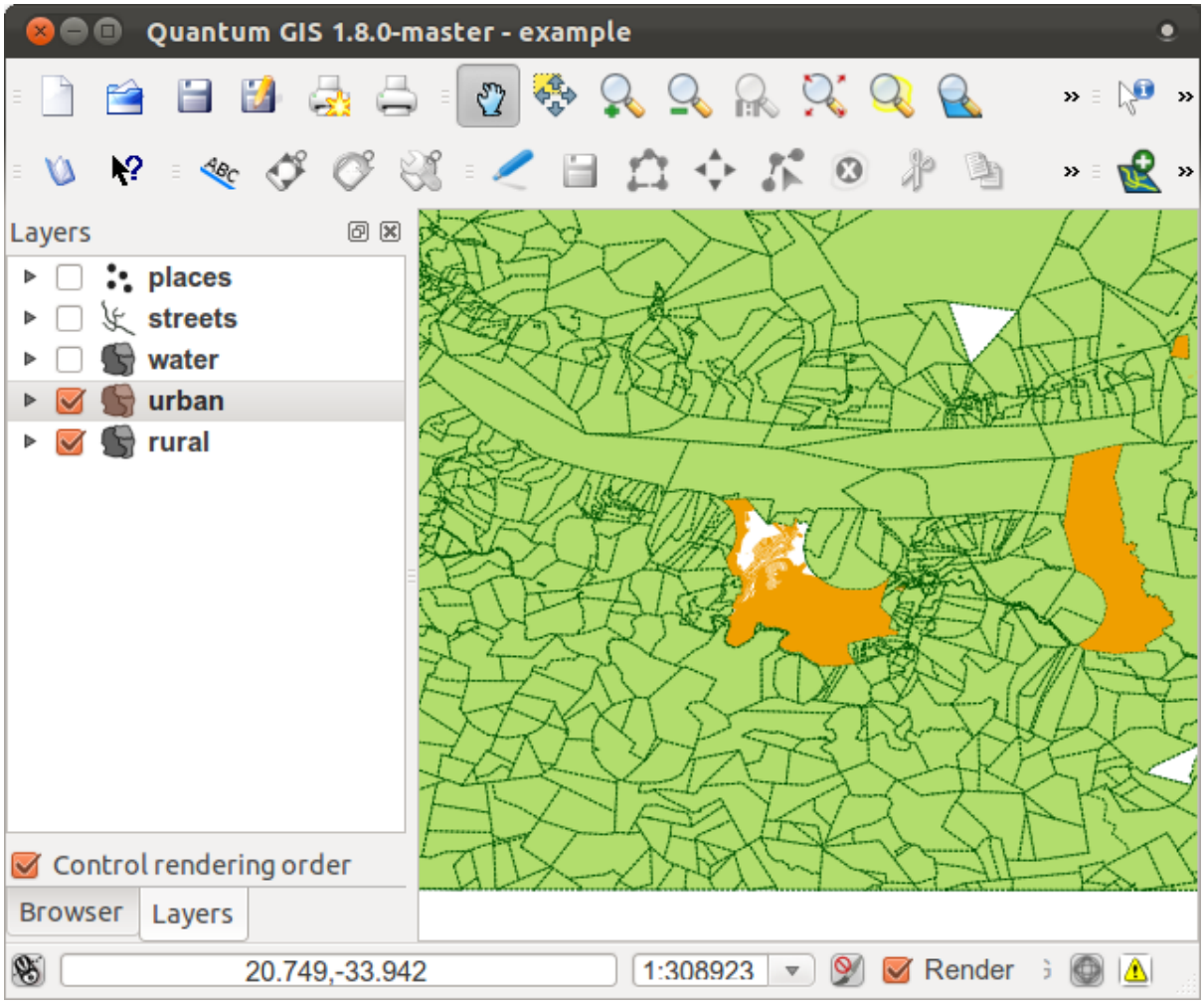



**Note:** If you want to work on only one layer at a time and don't want the other layers to distract you, you can hide a layer by clicking in the check box next to its name in the Layers list. If the box is blank, then the layer is hidden.

*Back to text*

### 18.4.2 Symbol Structure

Seen together, your rural and urban layers should look more or less like this:




If you are a Beginner-level user, you may stop here.

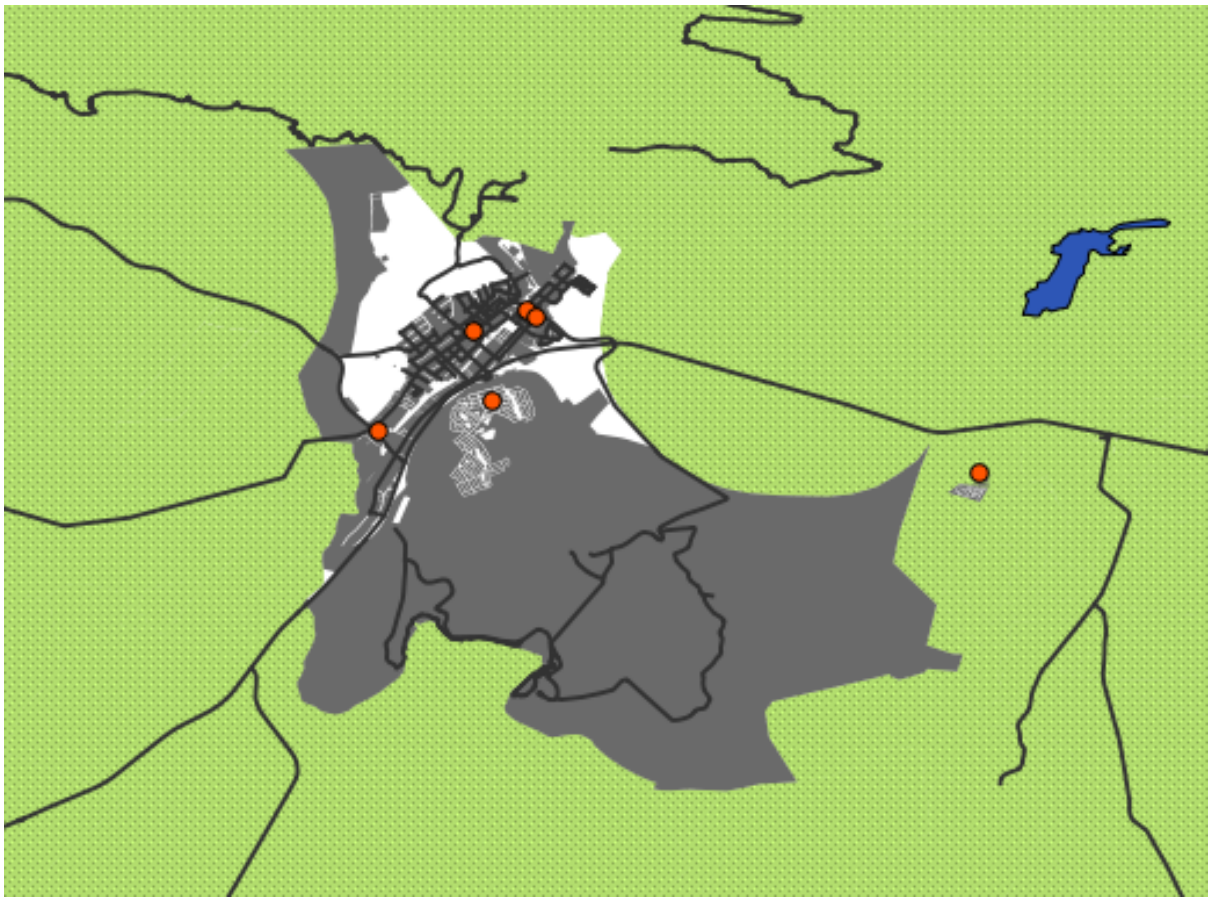
- Use the method above to change the colors and styles for all the remaining layers.
- Try using natural colors for the objects. For example, a road should not be red or blue, but can be gray or black.
- Also feel free to experiment with different *Fill Style* and *Border Style* settings for the polygons.

*Back to text*

### 18.4.3 Symbol Layers

- Customize your layers as you like, but remember that it has to be easy to tell different layers apart on the map.

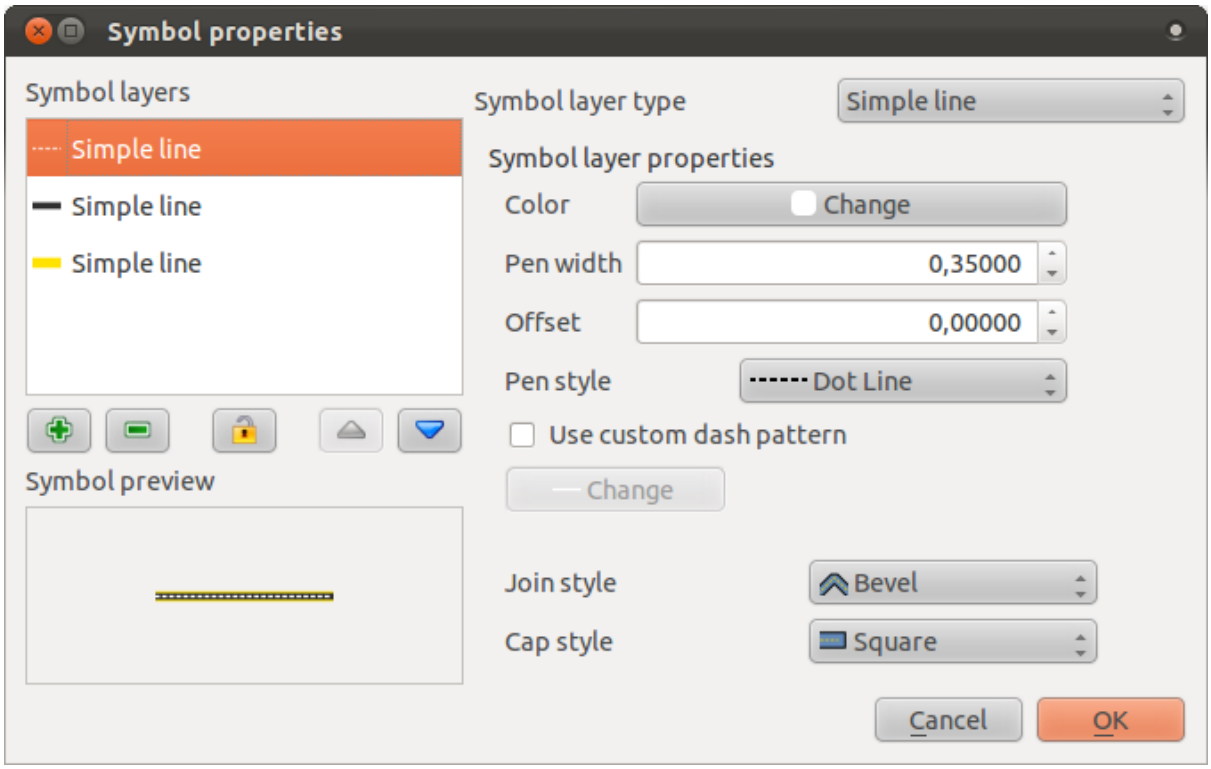
Here's an example:



*Back to text*

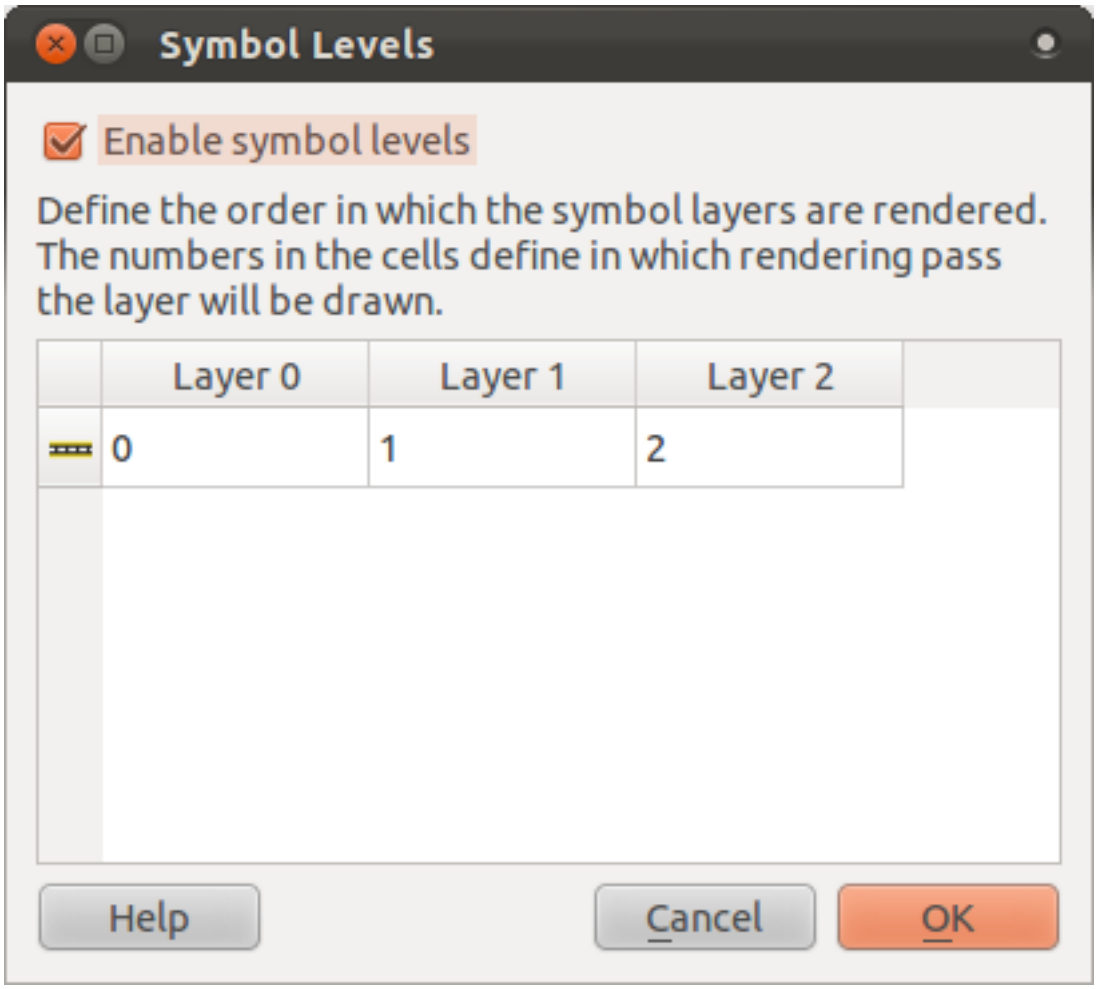
### 18.4.4 *Symbol Levels*

To make the required symbol, you need at least three symbol layers:

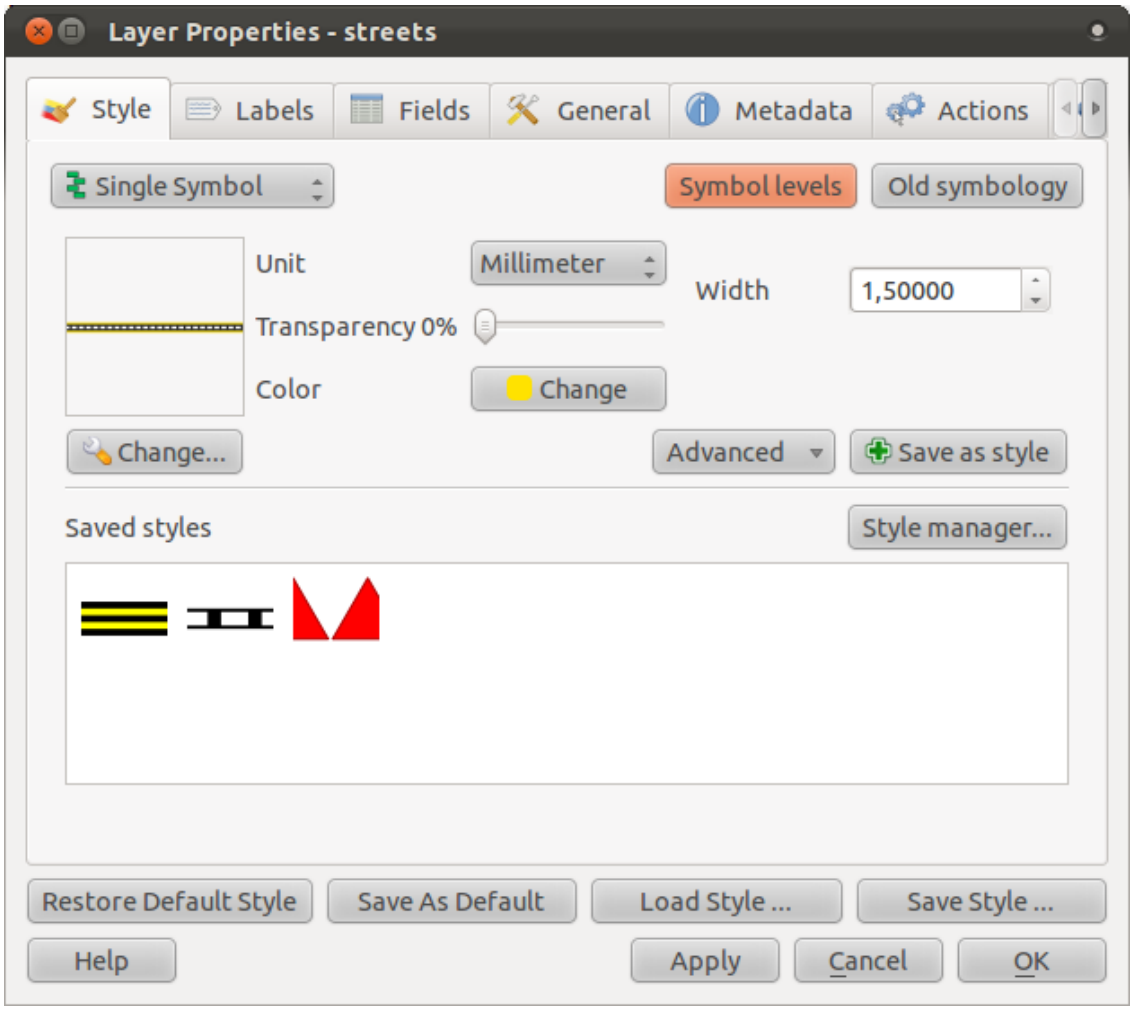



The lowest symbol layer is a broad, solid yellow line. On top of it there is a slightly thinner solid dark gray or black line. The symbol layer on top is a thin white dashed line.

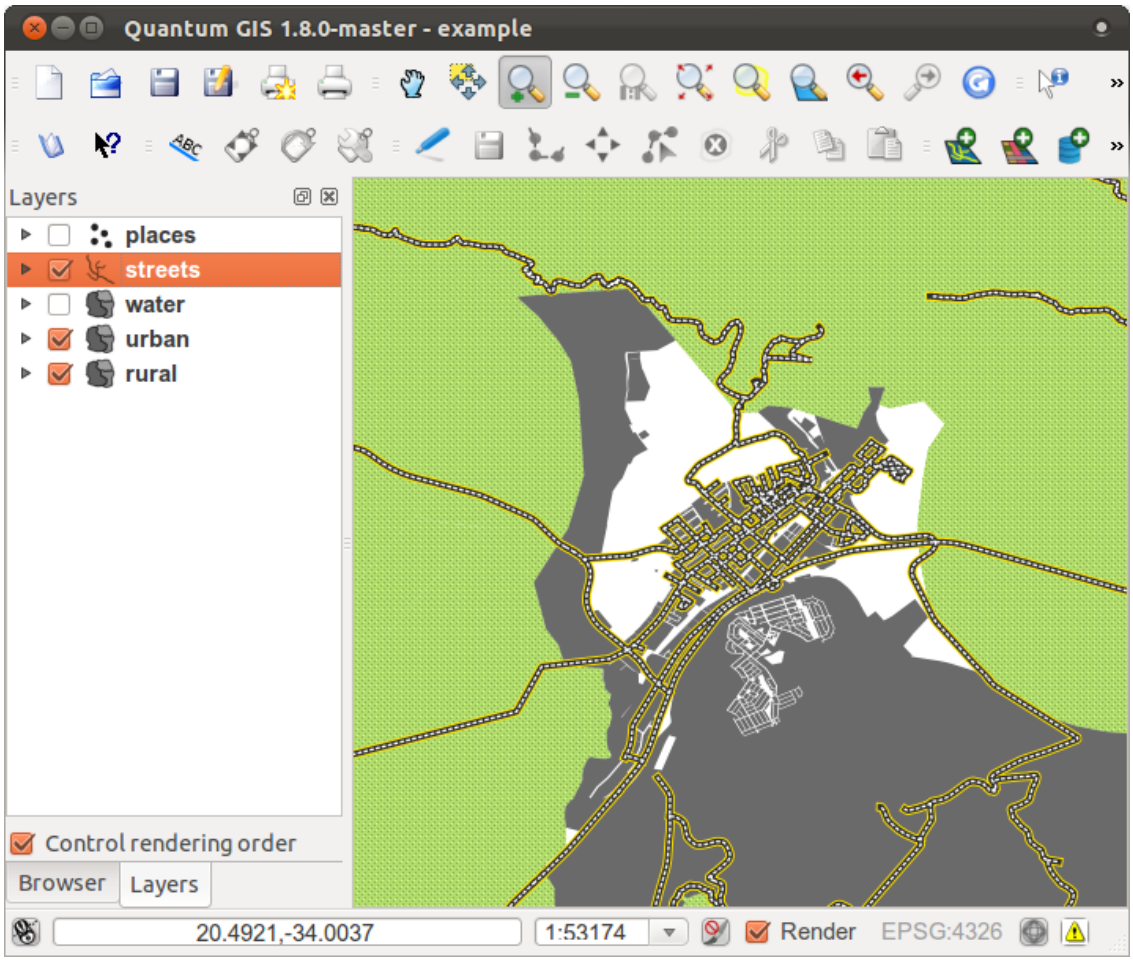
- If your symbol layers resemble the above but you're not getting the result you want, check that your symbol levels look something like this:




- Now your symbol should look like this:



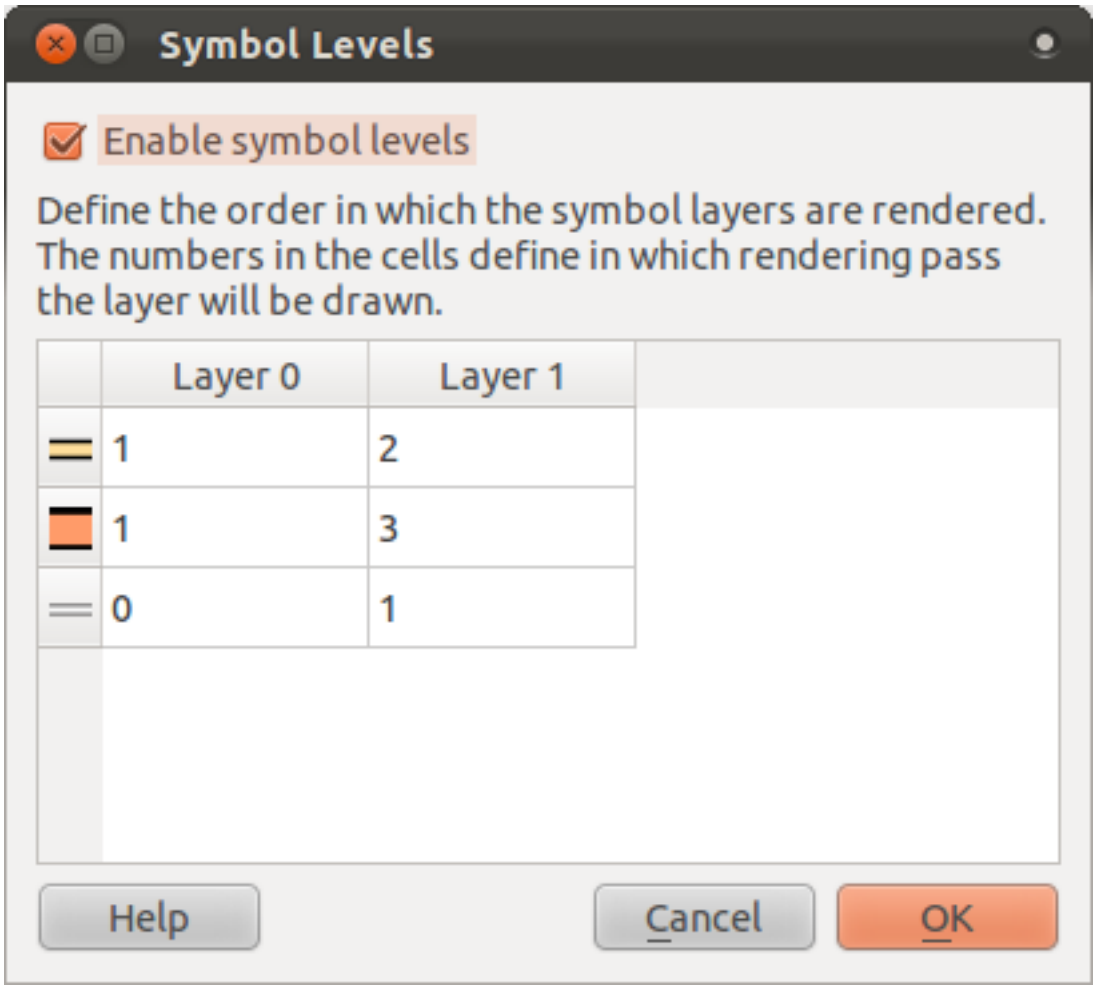

- Which, on the map, gives you this kind of effect:




*Back to text*

## 18.4.5 *Symbol Levels*

- Adjust your symbol levels to these values:




- Experiment with different values to get different results.
- Open your original map again before continuing with the next exercise.

*Back to text*

## 18.5 Results for *Attribute Data*

### 18.5.1 *Attribute Data*

The *NAME* field is the most useful to show as labels. This is because all its values are unique for every object, and it contains no "null" values (i.e., fields with no value).

*Back to text*

## 18.6 Results for *The Label Tool*

### 18.6.1 *Label Customization* (Part 1)

Your map should look like this:

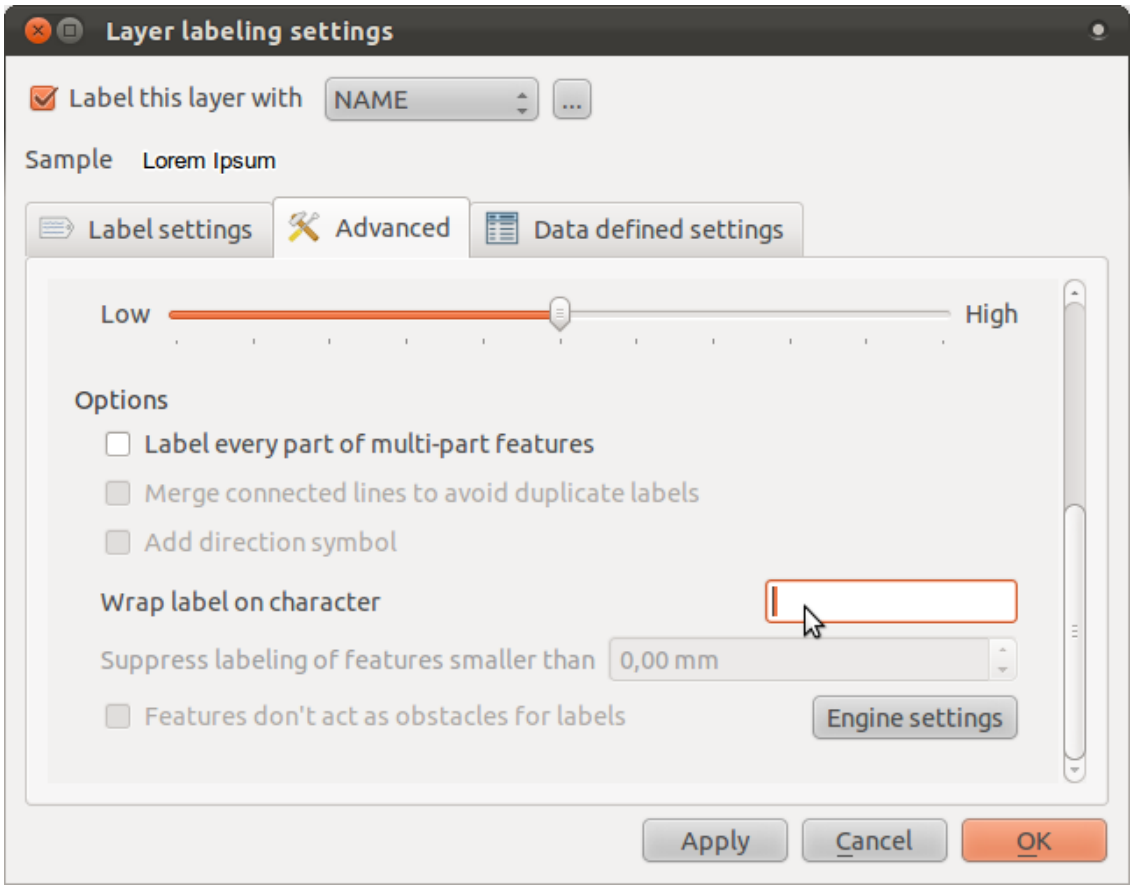
*Back to text*

### 18.6.2 *Label Customization* (Part 2)

One possible solution has this final product:

To arrive at this result:

- Use the font type `Arial` of size `10` and a *Label distance* of `1,5 mm`.
- In addition, this example uses the *Wrap label on character* option:

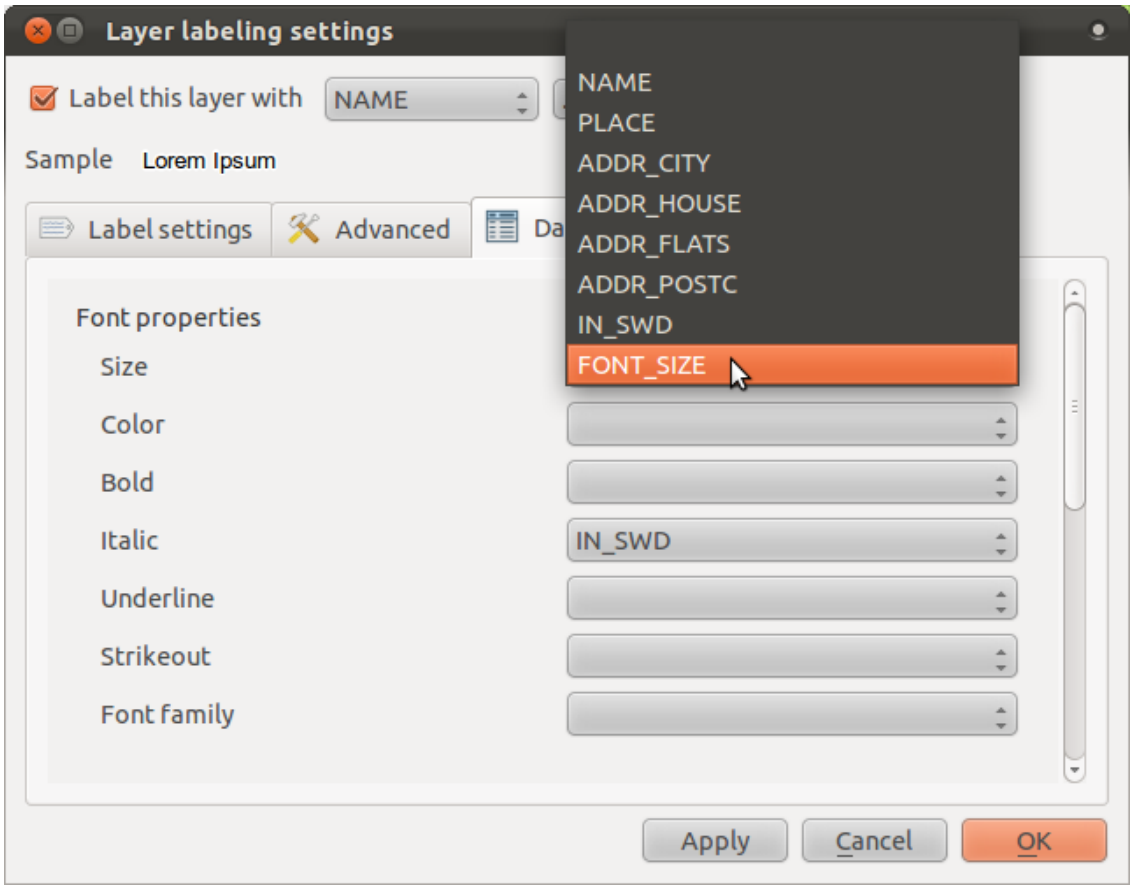



- Enter a `space` in this field and click *Apply* to achieve the same effect.

*Back to text*

### 18.6.3 *Using Data Defined Settings*

- Still in edit mode, set the `FONT_SIZE` values to whatever you prefer. The example uses `12` for towns, `10` for suburbs and villages, and `8` for the null entries.
- Remember to save changes and exit edit mode.
- In *Data defined settings*, do this:



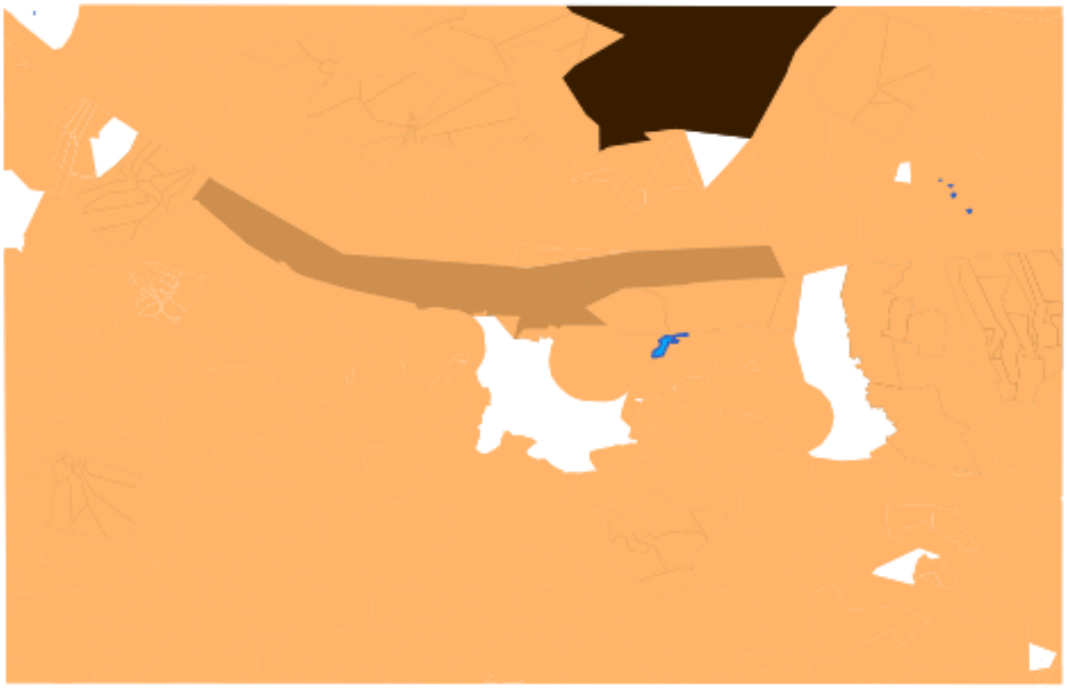

Your results, if using the above values, should be this:

*Back to text*

## 18.7 Results for *Classification*

### 18.7.1 *Refine the Classification*

- Use the same method as in the first exercise of the lesson to get rid of the lines:



The settings you used might not be the same, but with the values *Classes* = 7 and *Mode* = *Natural Breaks (Jenks)* (and using the same colors, of course), the map will look like this:

*Back to text*

## 18.8 Results for *Creating a New Vector Dataset*

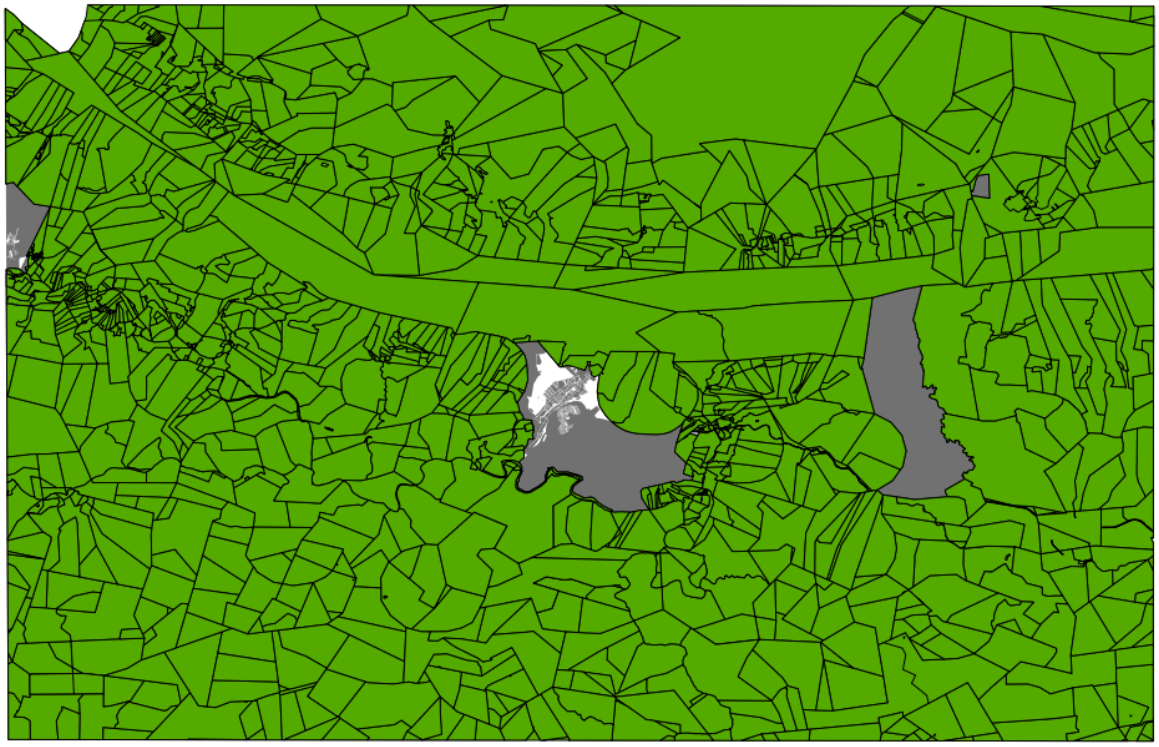
### 18.8.1 *Digitizing*

The symbology doesn't matter, but the results should look more or less like this:

*Back to text*

### 18.8.2 *Topology: Avoid Intersections*

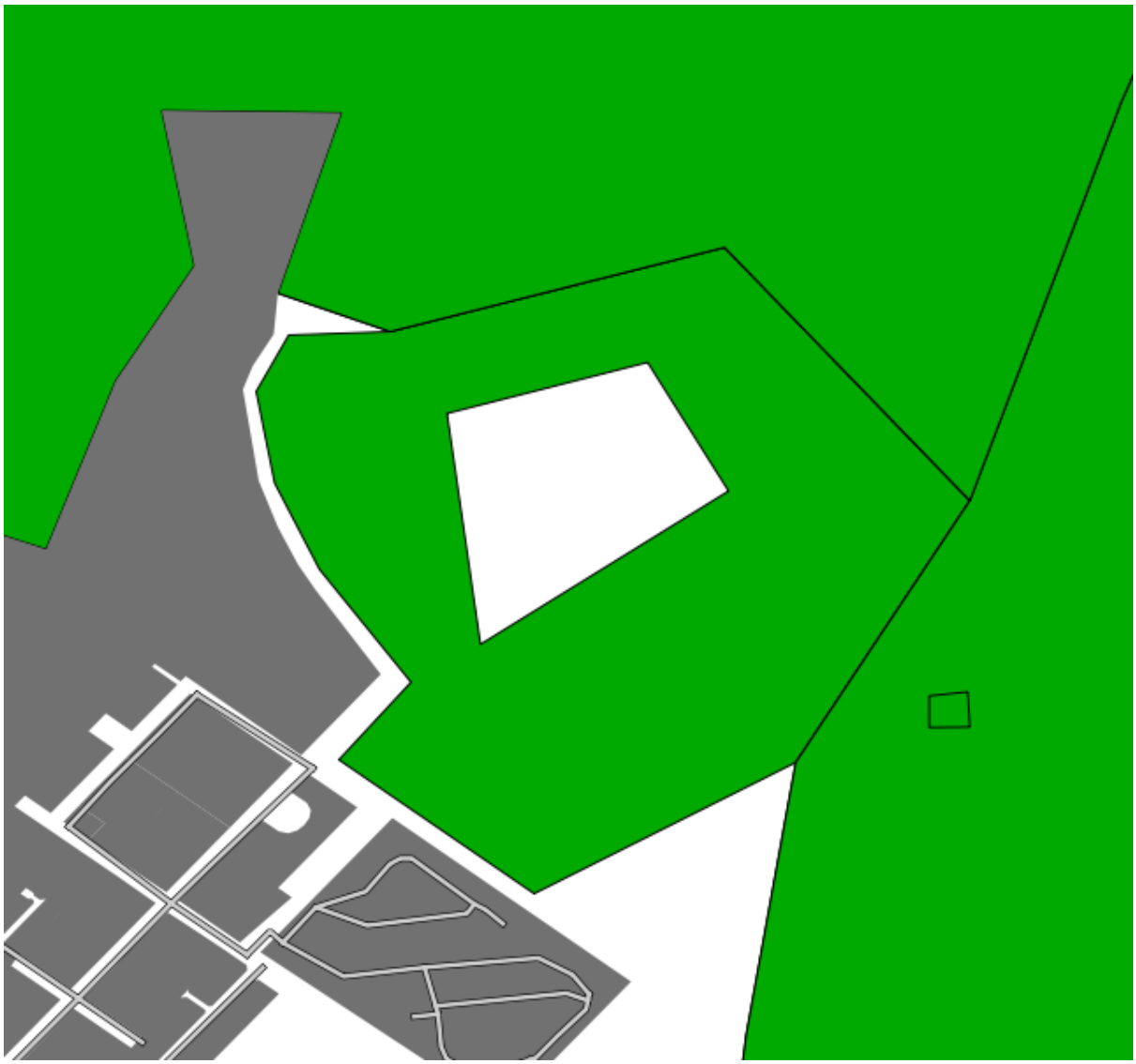
Your map should look like this:



*Back to text*

### 18.8.3 *Topology: Add Ring Tool*

The exact shape doesn't matter, but you should be getting a hole in your feature, like this one:

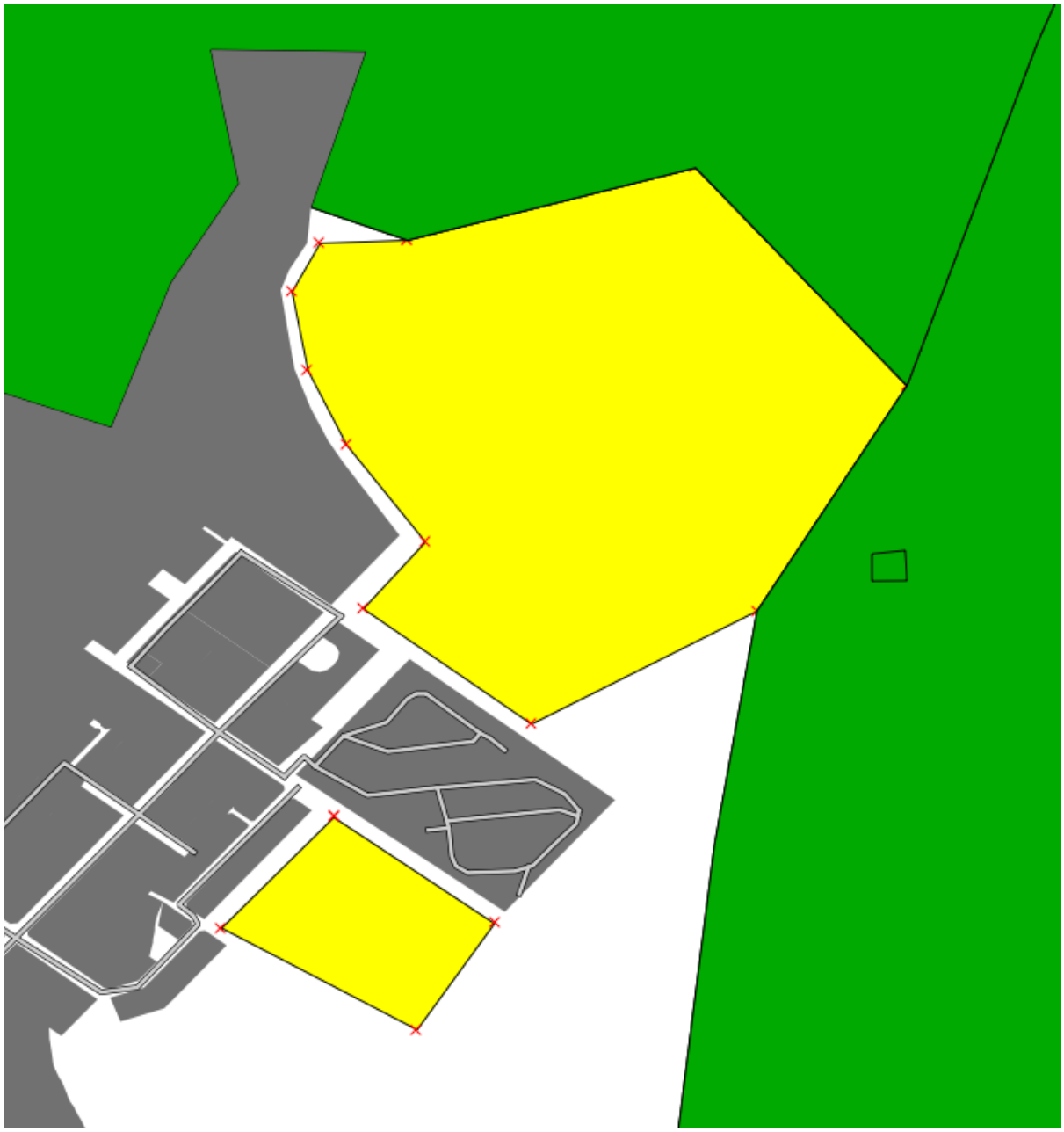


- Undo your edit before continuing with the exercise for the next tool.

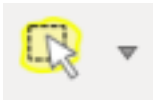
*Back to text*

### 18.8.4 *Topology: Add Part Tool*

The exact shape and location doesn't matter, but you should be getting an extra part on your feature, like this one:



- Note that you first need to select the farm with the selection tool ...



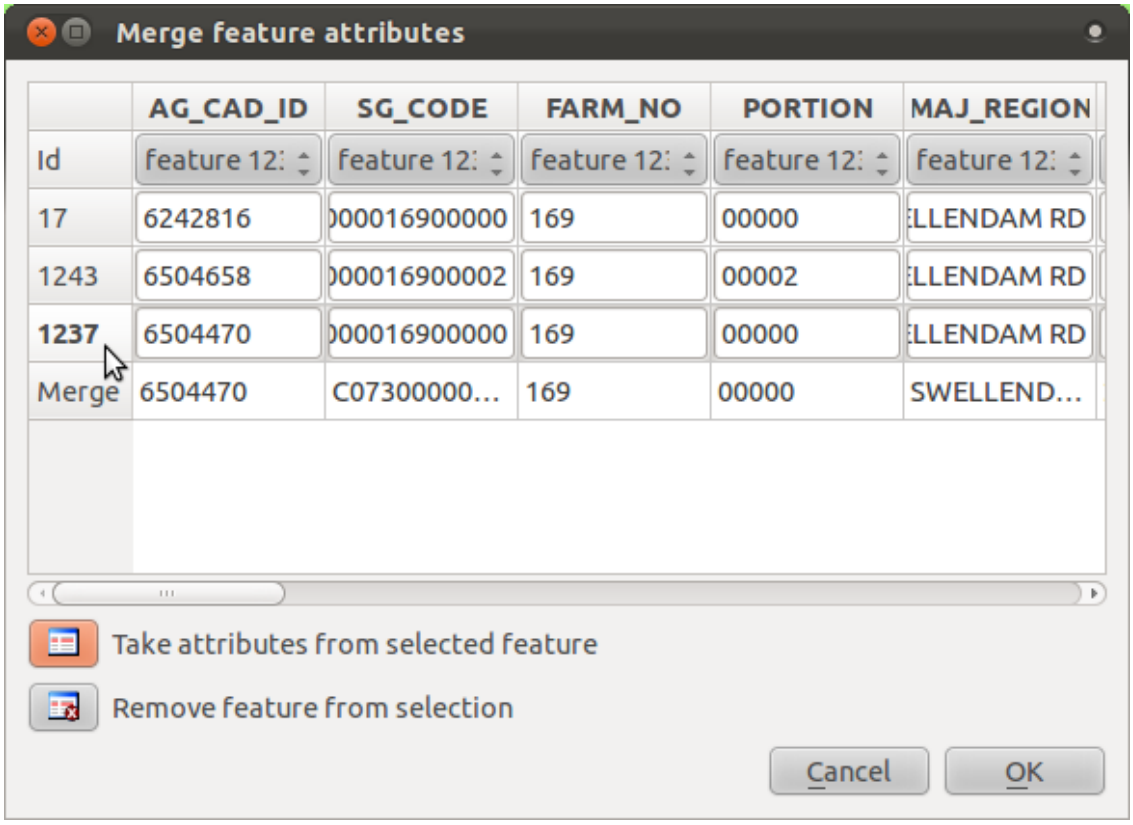
  ... so that QGIS knows which feature to add the new part to.

- Undo your edit before continuing with the exercise for the next tool.

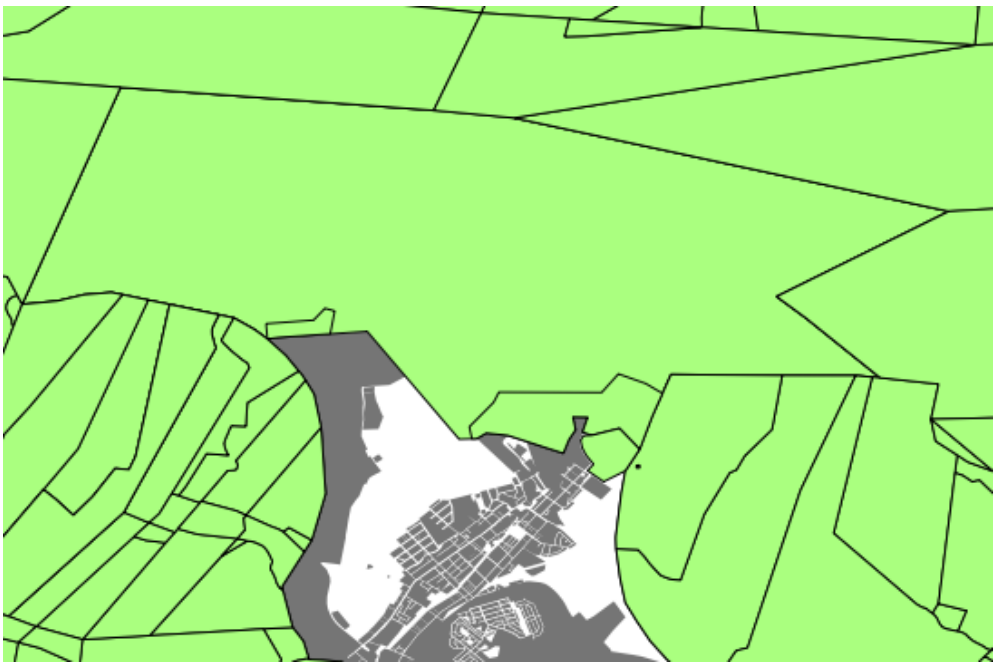
*Back to text*

### 18.8.5 *Merge Features*

- Use the feature with the *id* of `1237` as the source of your attributes (click on its entry in the dialog, then click the *Take attributes from selected feature* button):




- Using *Merge Selected Features* will merge the attributes as well as the geometries:



Whereas *Merge Attributes of Selected Features* will keep the geometries distinct, but give

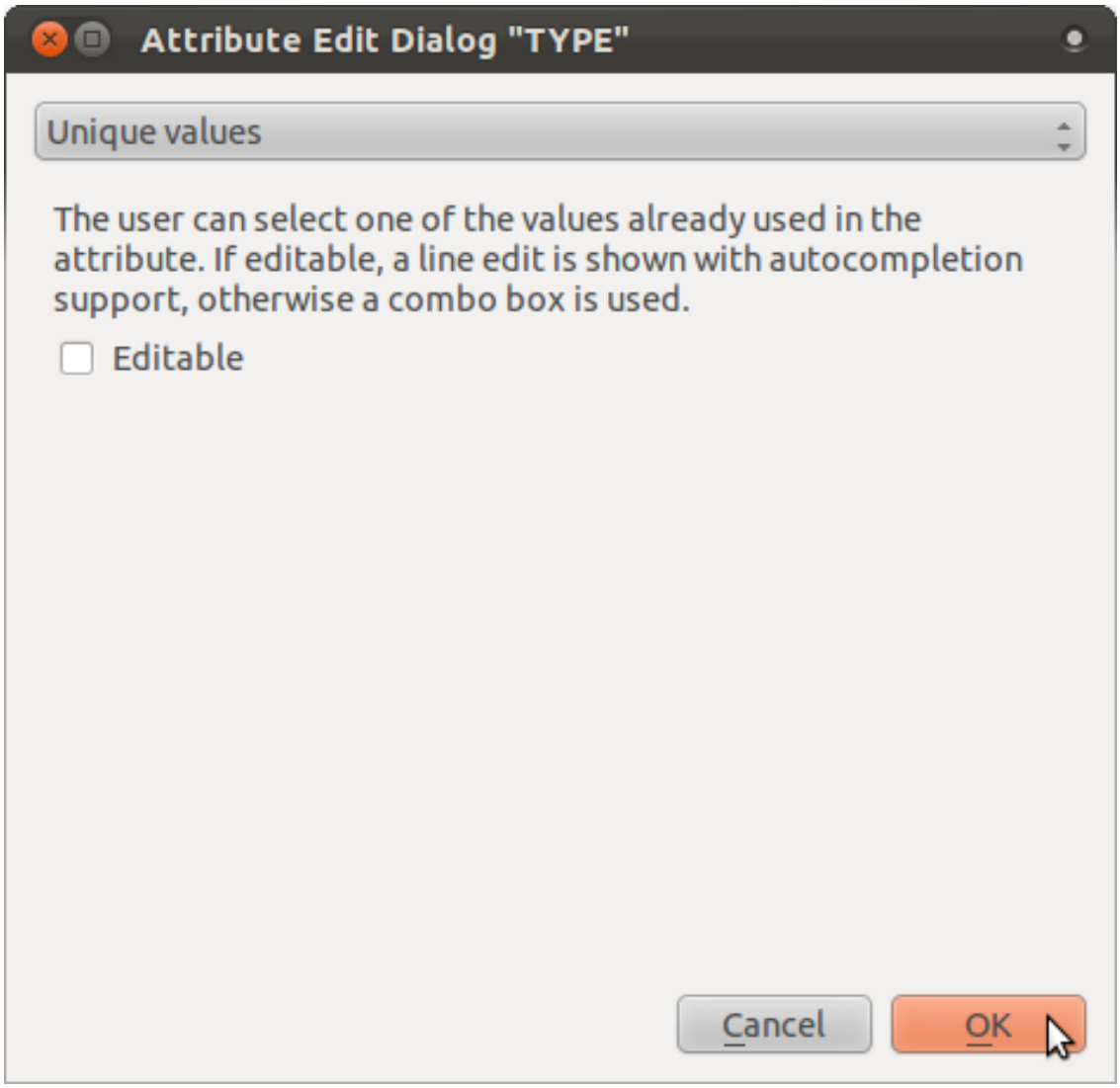
them the same attributes.

*Back to text*

### 18.8.6 *Forms*

For the *TYPE*, there is obviously a limited amount of types that a road can be, and if you check the attribute table for this layer, you'll see that they are predefined.

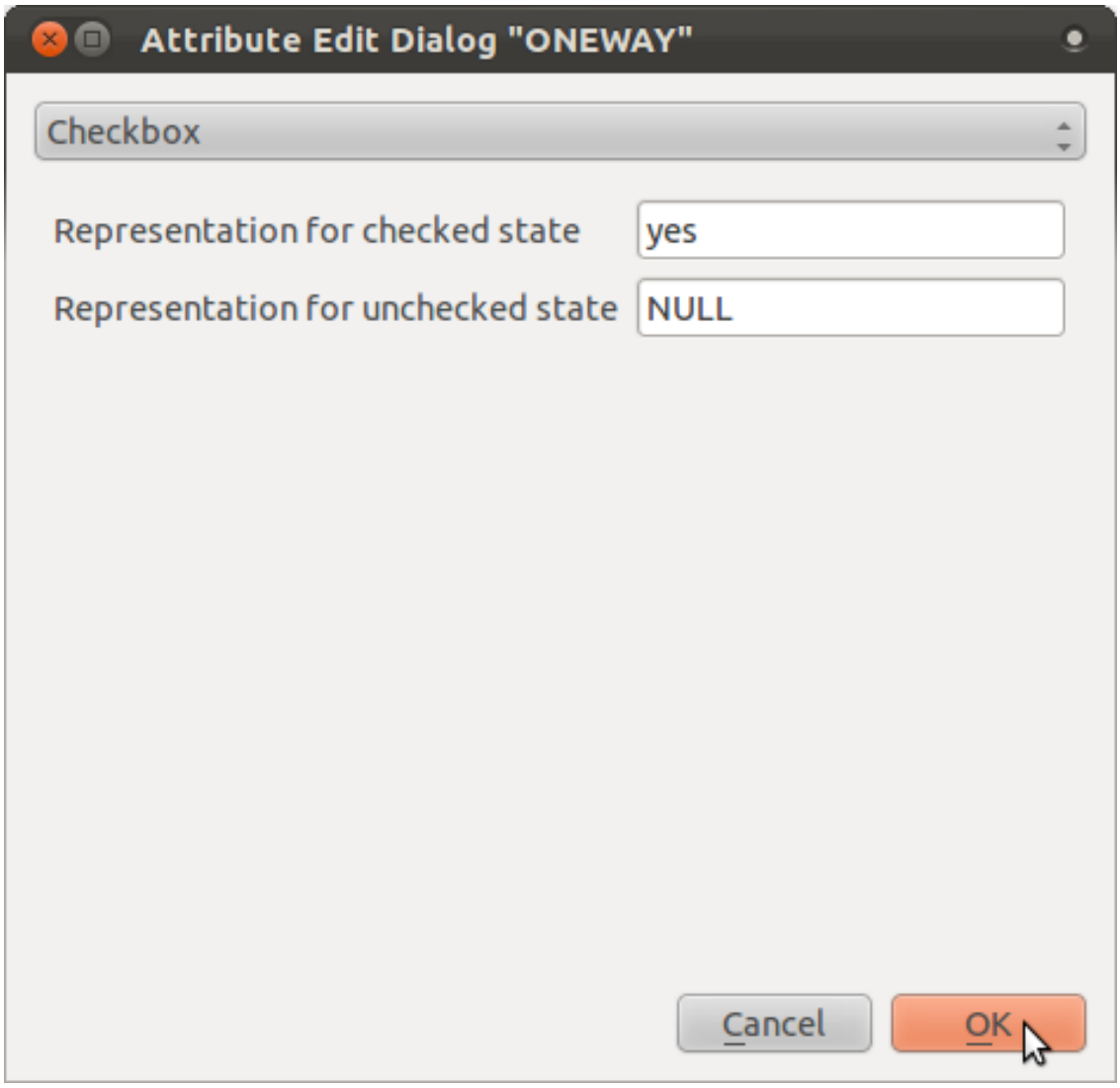
- Set the widget to *Unique values* and leave *Editable* as `False`:




- A road is either a one-way or it isn't, so *ONEWAY* should be a *Checkbox*.

Looking at the attribute table for *streets*, the value for when it's not a one-way seems to be *NULL*, and when it is, it's *yes*.

- Set those values:



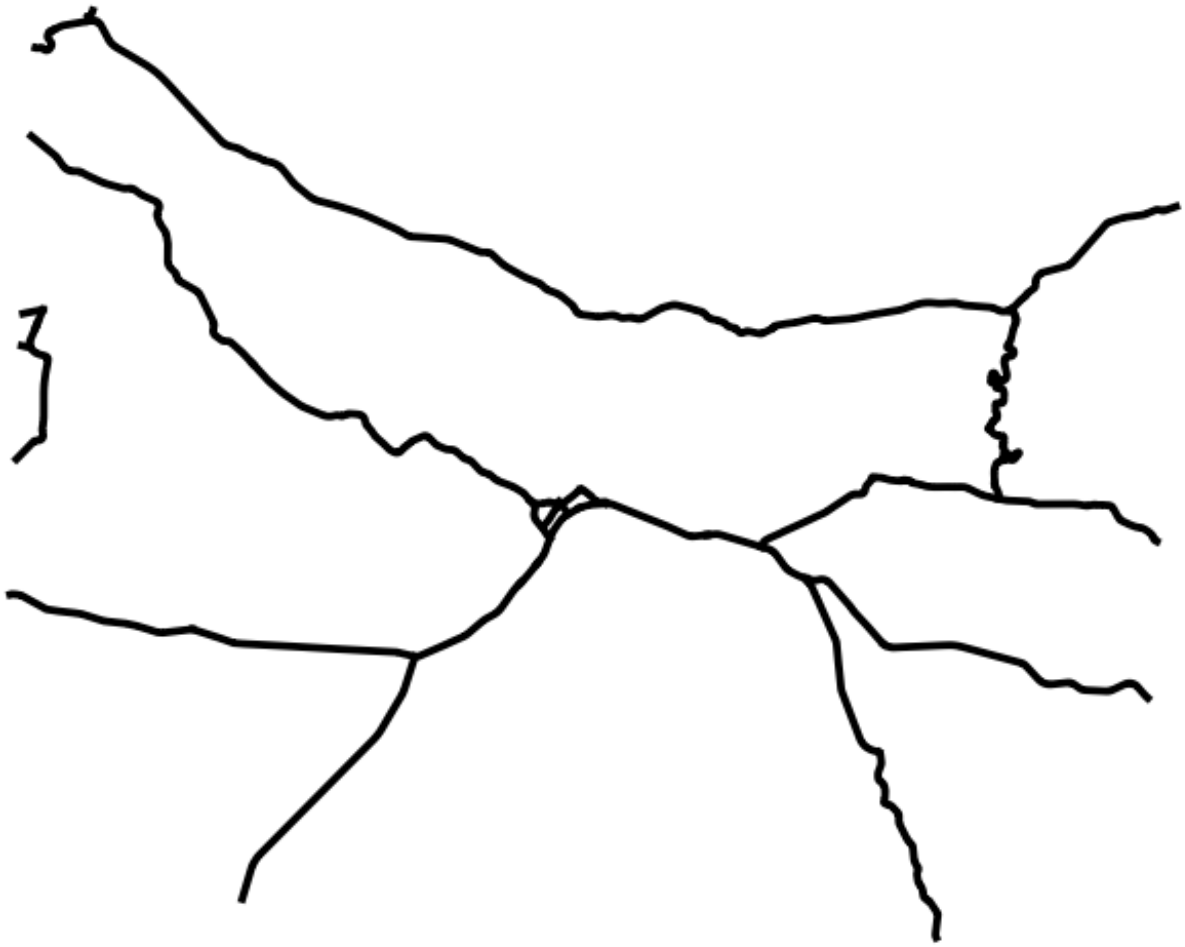

- If you use the *Identify* tool on a street now while edit mode is active, the dialog you get should look like this:

*Back to text*

## 18.9 Results for *Vector Analysis*

### 18.9.1 *Find Important Roads*

Your layer should now have these roads:

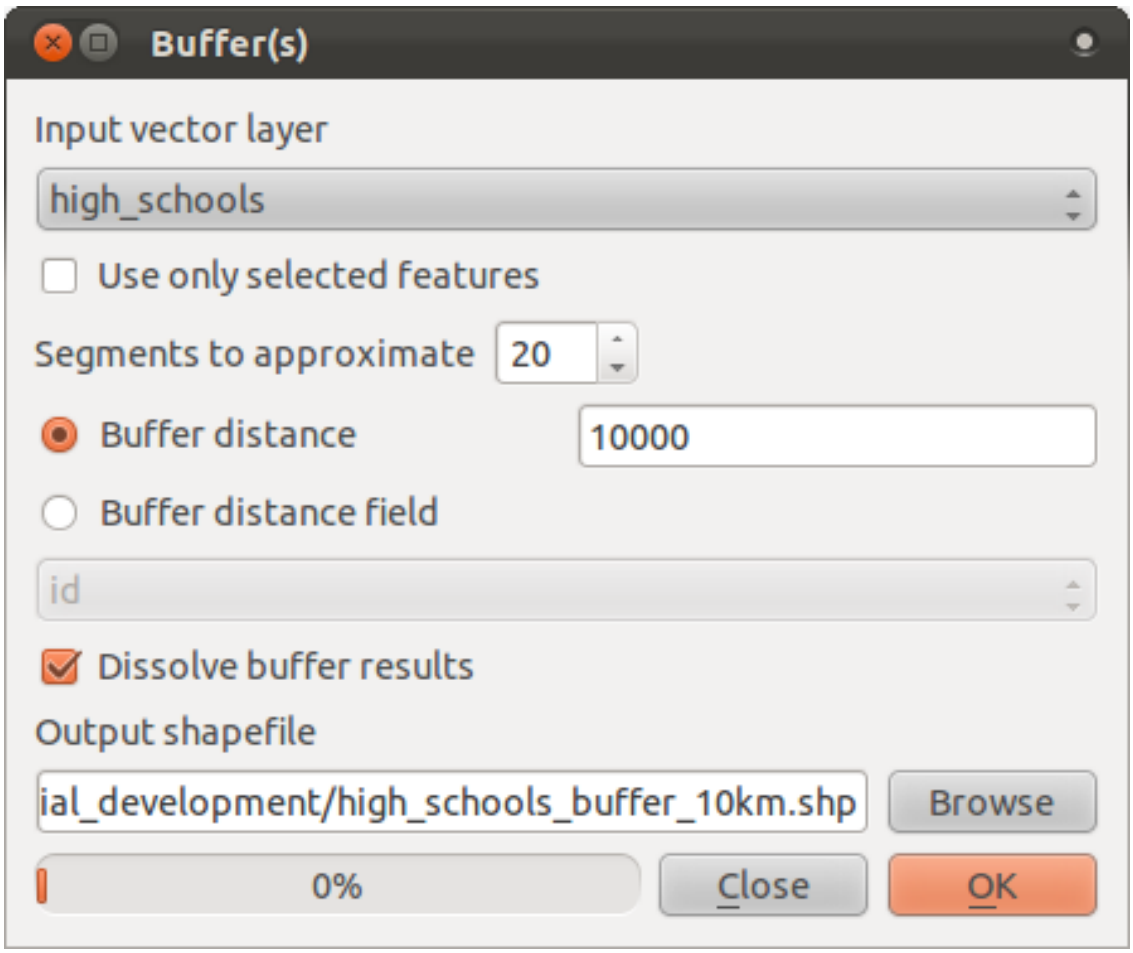


- Save this new layer (the same way you did before) under `exercise_data/residential_development/`, as `important_roads.shp`.
- Once the new layer has been added to your map, remove the old layer.
- If for some reason it gives you an error message saying that it can't add the layer to the map (this can happen), remember that you can add the layer yourself by using the *Add Vector Layer* button as you did before.

*Back to text*

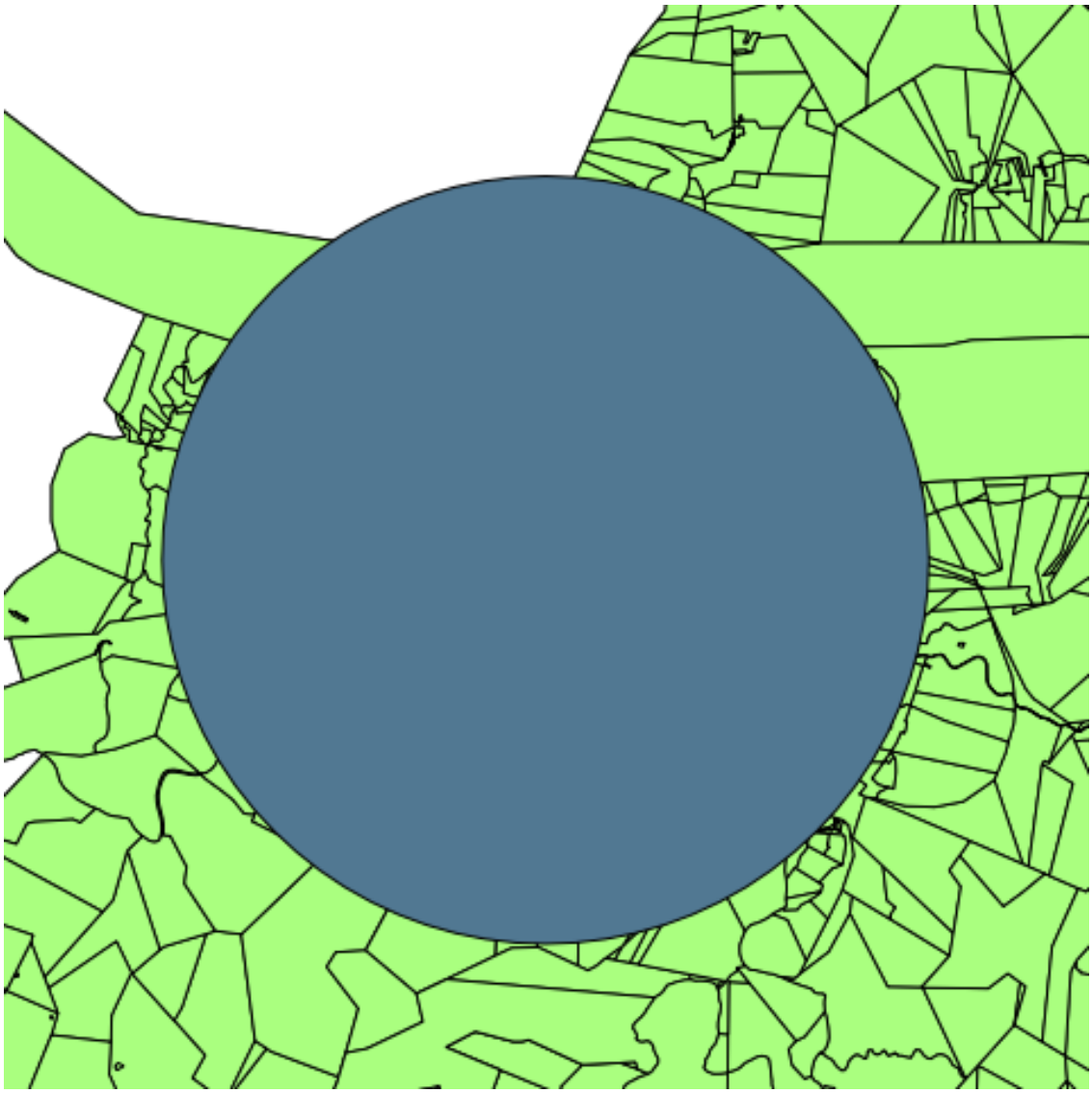
### 18.9.2 *Distance from High Schools*

- Your buffer dialog should look like this:

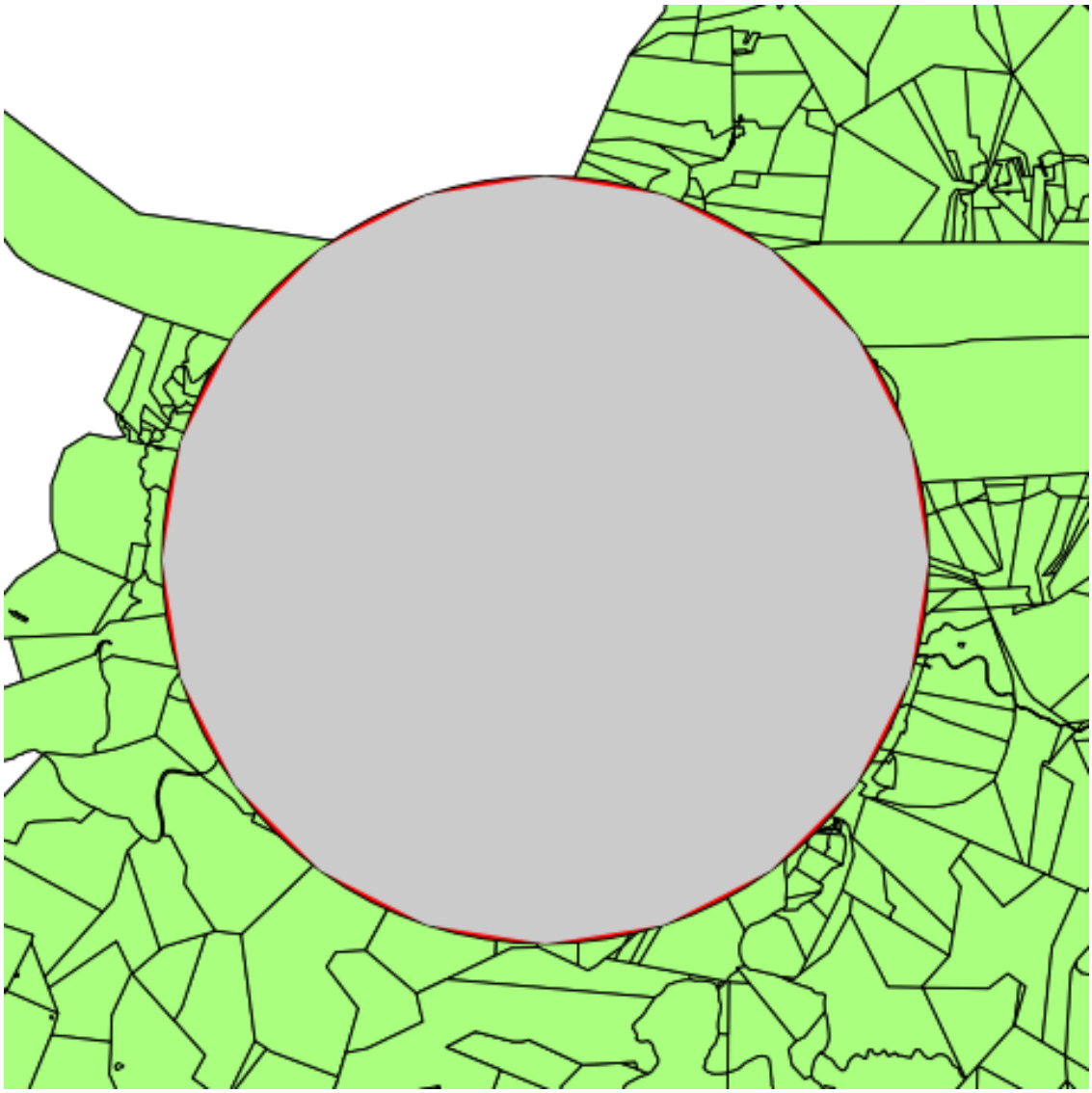



  The *Buffer distance* is `10000` meters (i.e., `10` kilometers).

- The *Segments to approximate* value is set to `20`. This is optional, but it's recommended, because it makes the output buffers look smoother. Compare this:



To this:



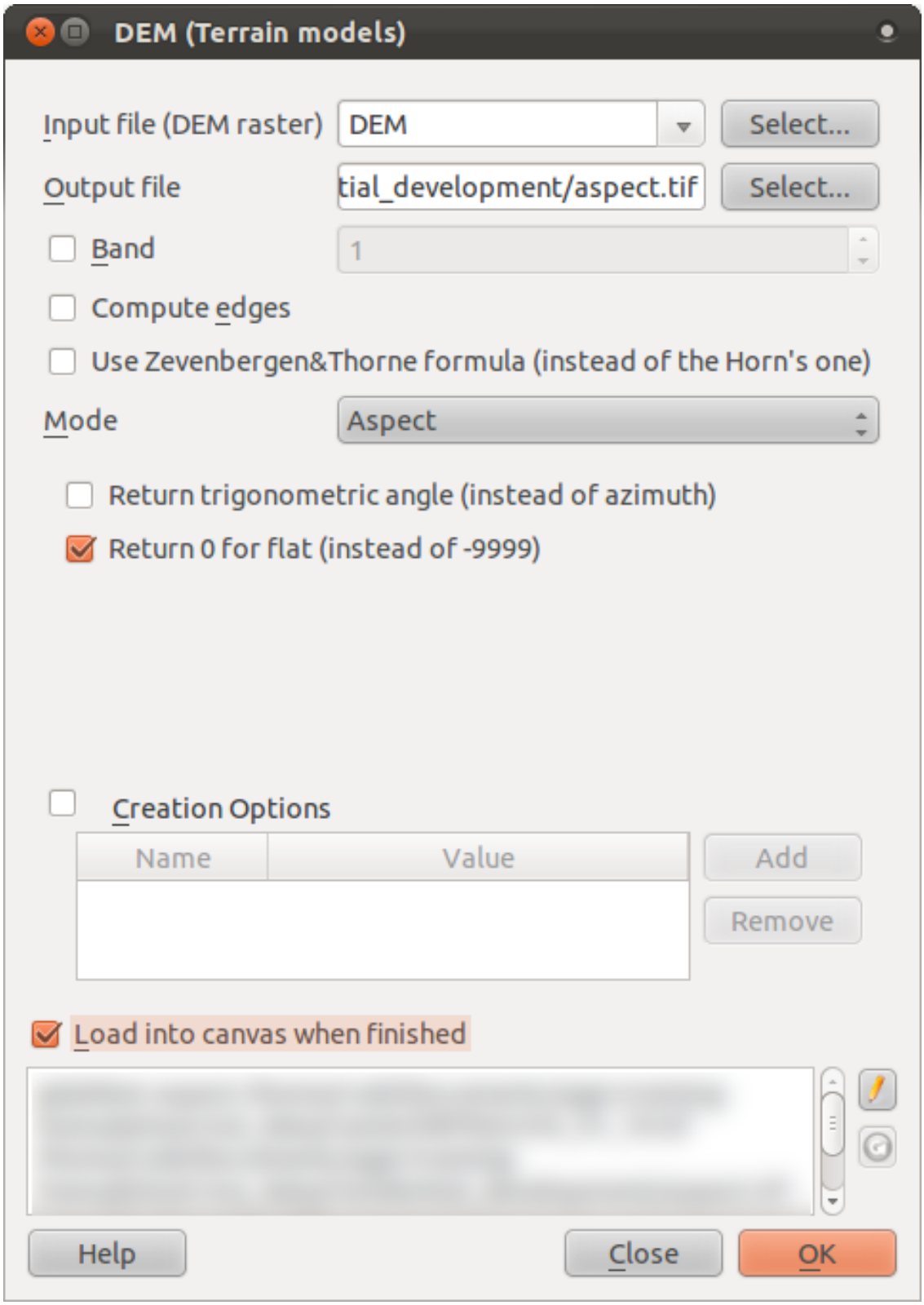
The red circle is the buffer with *Segments to approximate* set to `20`; the gray circle on top of it is the buffer with *Segments to approximate* set to `5`.

*Back to text*

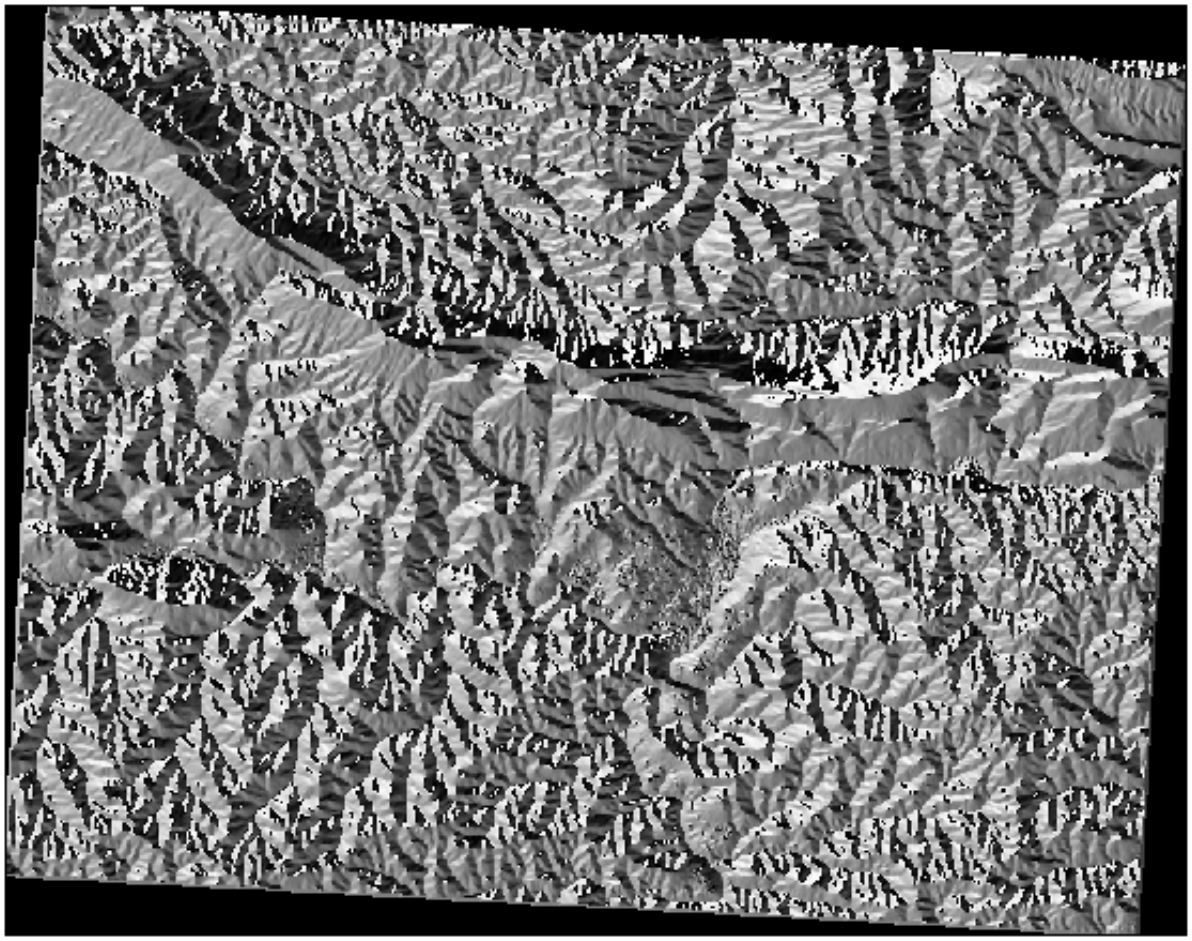
## 18.10 Results for *Raster Analysis*

### 18.10.1 *Calculate Aspect*

- Set your *DEM (Terrain analysis)* dialog up like this:


DEM (Terrain models)
Input file (DEM raster)
DEM
Select...
Output file
tial_development/aspect.tif
Select...
Band
1
Compute edges
Use Zevenbergen&Thorne formula (instead of the Horn's one)
Mode
Aspect
Return trigonometric angle (instead of azimuth)
Return 0 for flat (instead of -9999)
Creation Options
Name
Value
Add
Remove
Load into canvas when finished
Help
Close
OK

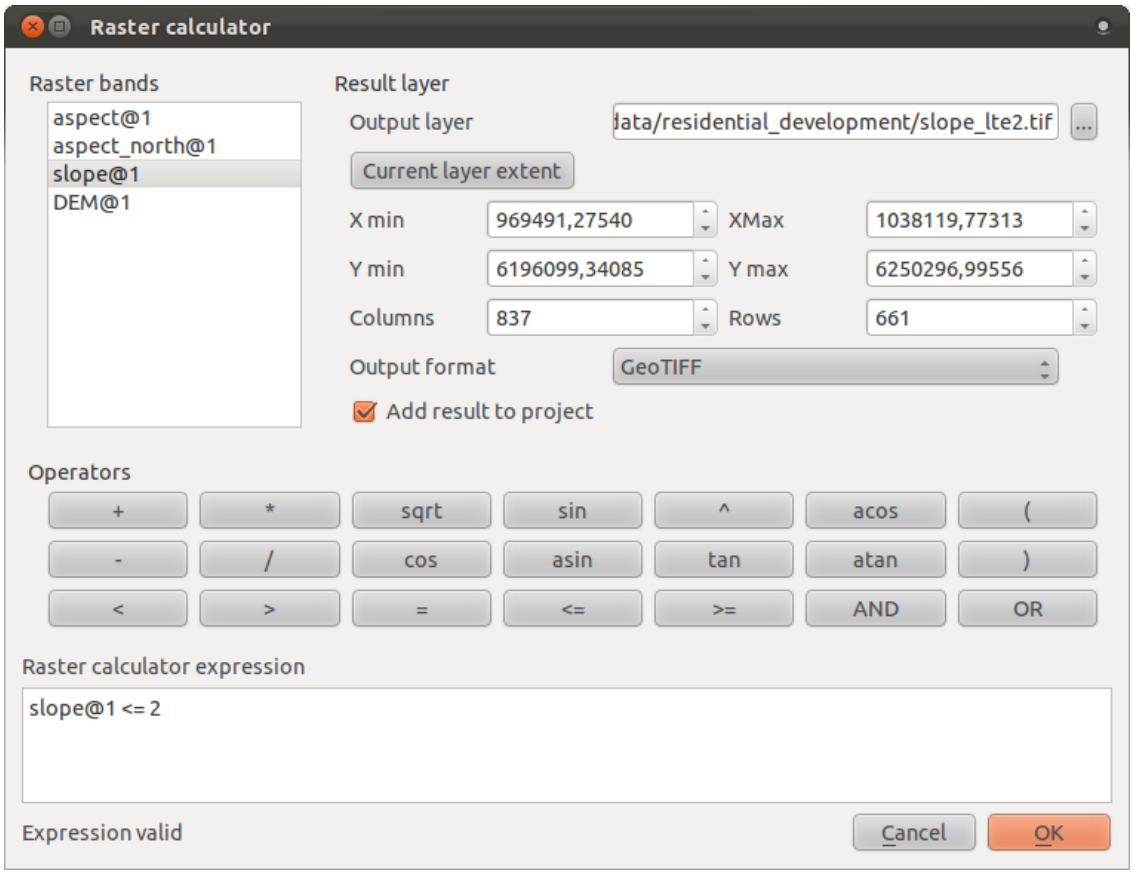
Your result:



*Back to text*

### 18.10.2 *Calculate Slope (less than 2 and 5 degrees)*

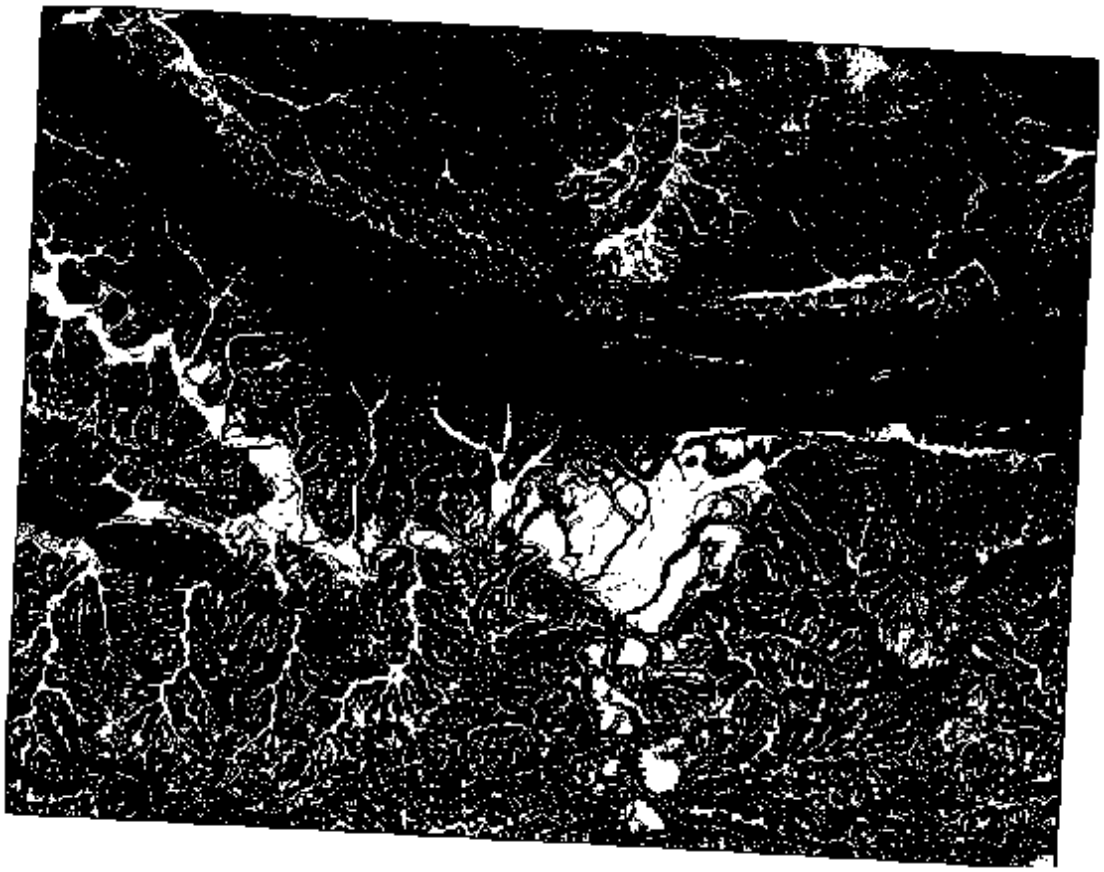
- Set your *Raster calculator* dialog up like this:




- For the 5 degree version, replace the `2` in the expression and file name with `5`.

Your results:

- 2 degrees:



- 5 degrees:

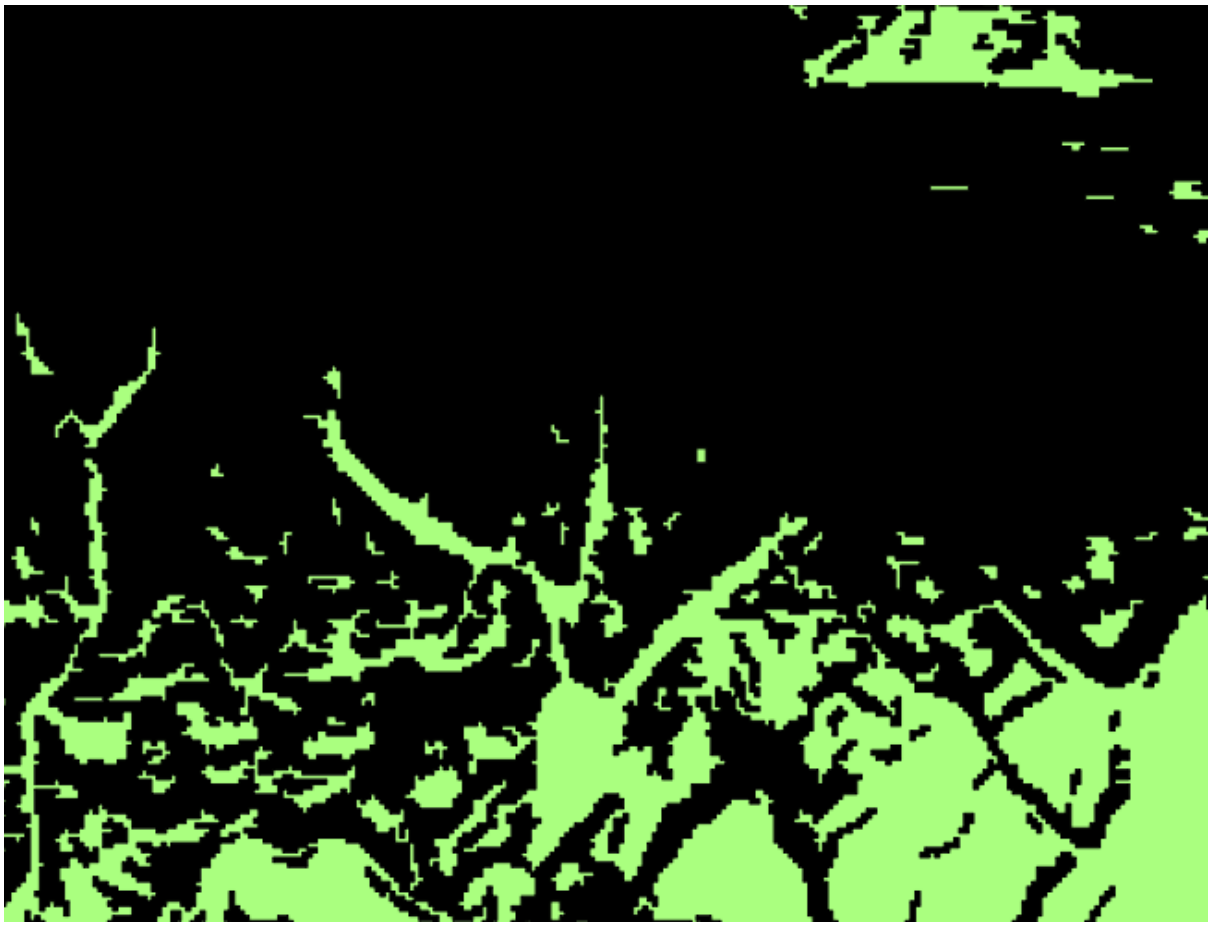
*Back to text*

## 18.11 Results for *Completing the Analysis*

### 18.11.1 *Raster to Vector*

- Open the *Query* dialog by right-clicking on the *all_terrain* layer in the *Layers list*.
- Then build the query `"suitable" = 1`.
- Click *OK* to filter out all the polygons where this condition isn't met.

When viewed over the original raster, the areas should overlap perfectly:

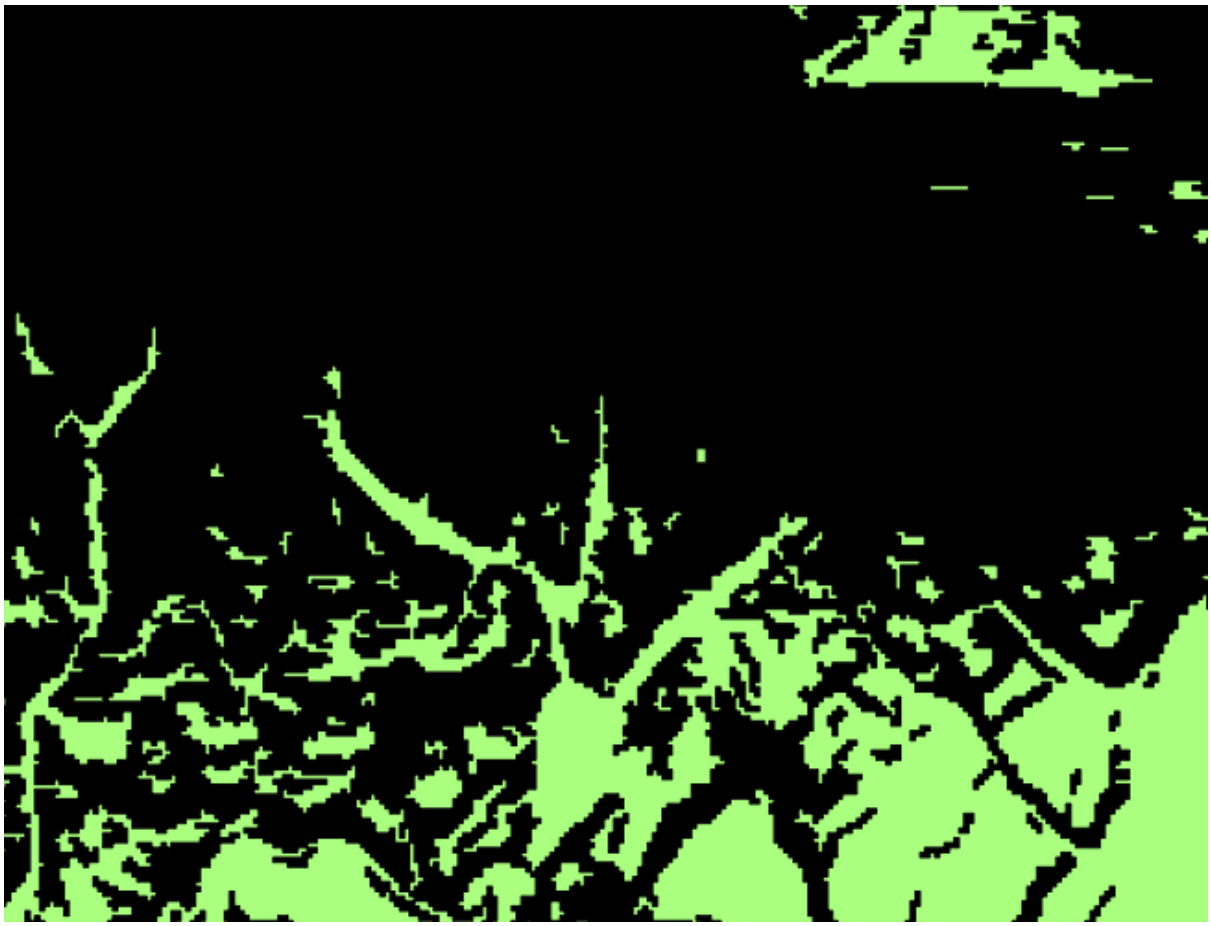


- You can save this layer by right-clicking on the *all_terrain* layer in the *Layers list* and choosing *Save As...*, then continue as per the instructions.

*Back to text*

### 18.11.2 *Inspecting the Results*

In this map, farms that are definitely not suitable are orange; farms that may still be suitable are blue:

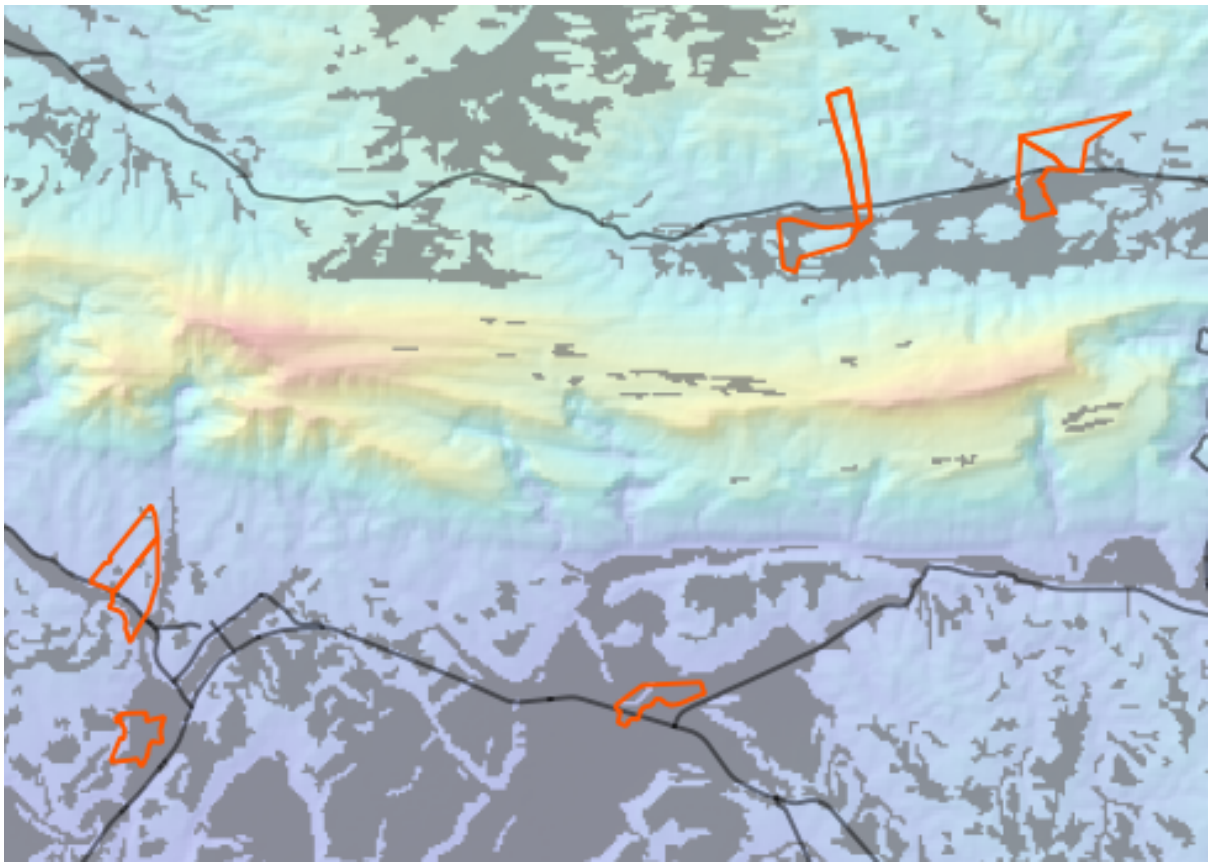


The orange farms have almost no suitable terrain within them, and so can't be used for the intended purpose.

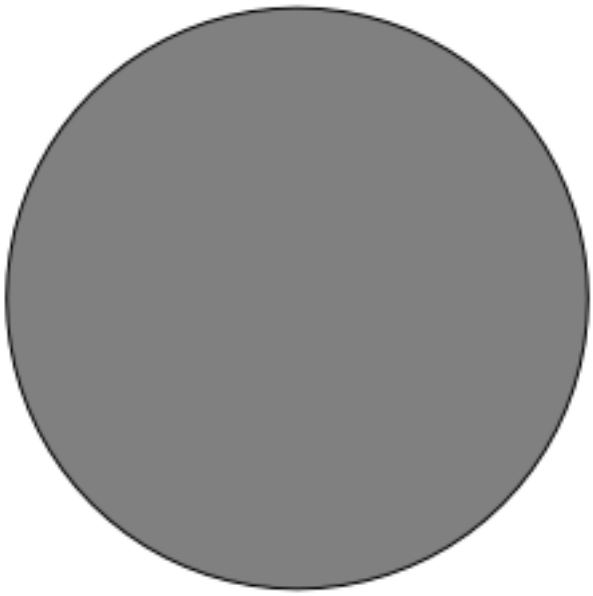
*Back to text*

### 18.11.3 *Refining the Analysis*

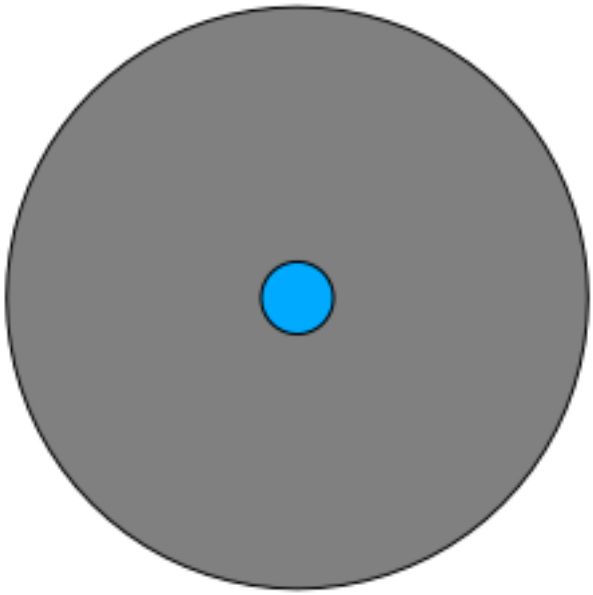
At the moment, your analysis should look like this (ignore the symbology):



Consider a circular area, continuous for 350 meters in all directions.



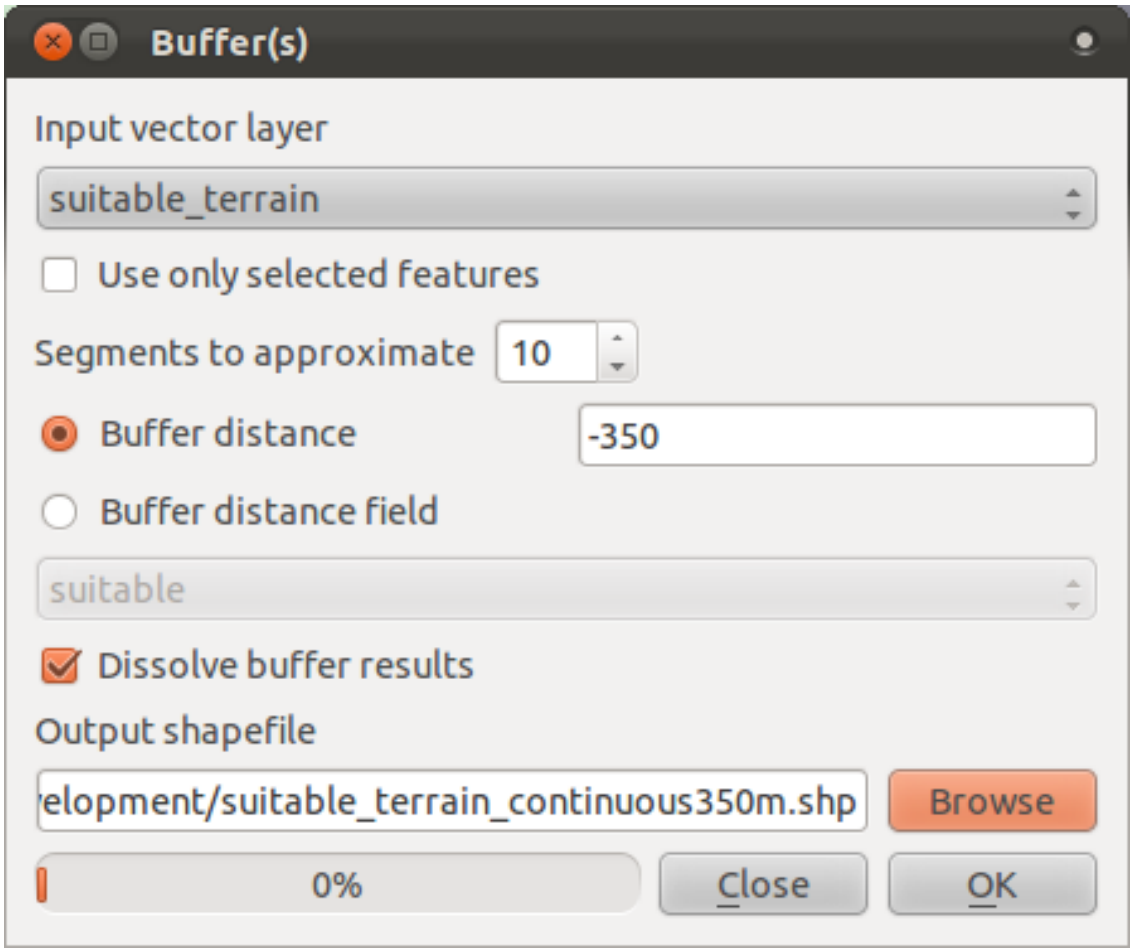
If it is greater than 350 meters in radius, then subtracting 350 meters from its size (from all directions) will result in a part of it being left in the middle.



Therefore, you can run an *interior buffer* of 350 meters on your existing *suitable_terrain* vector layer. In the output of the buffer function, whatever remains of the original layer will represent areas where there is suitable terrain for 350 meters beyond.

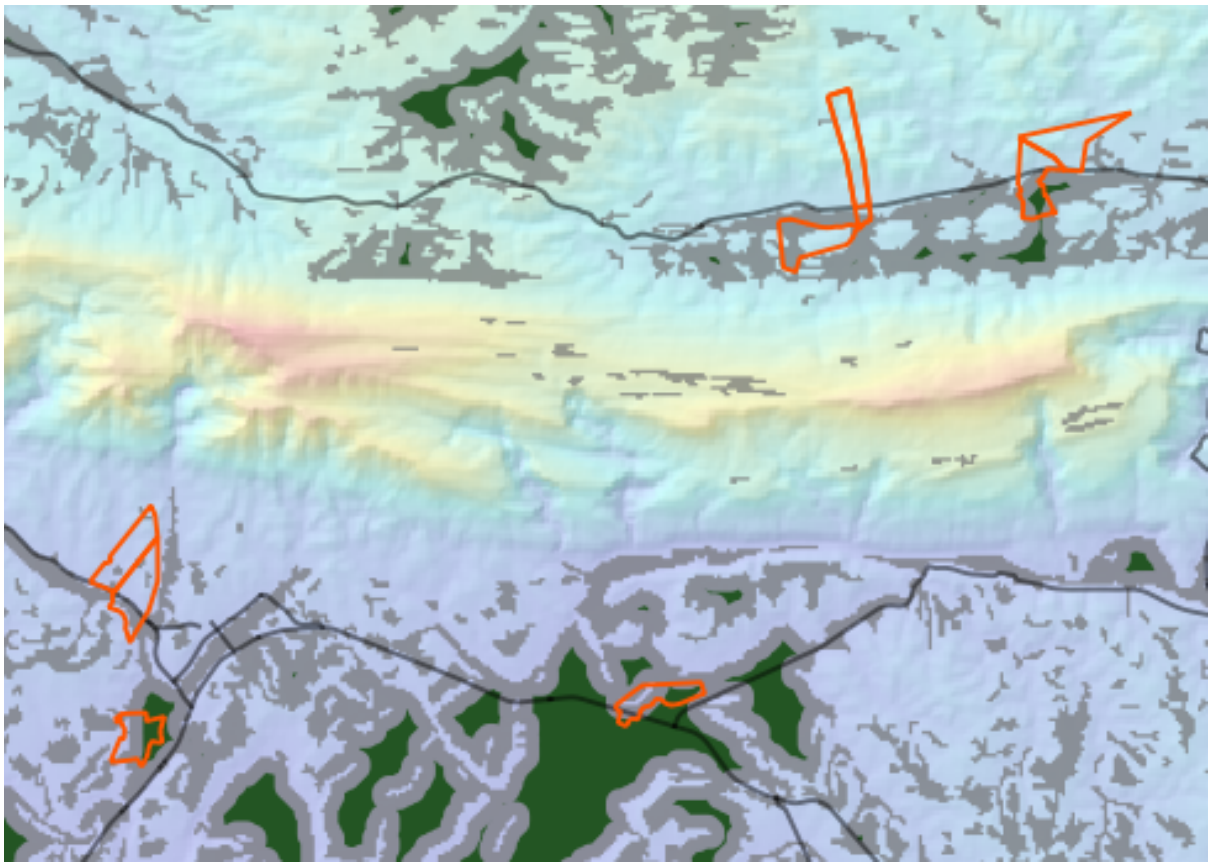
To demonstrate:

- Go to *Vector* → *Geoprocessing Tools* → *Buffer(s)* to open the Buffer(s) dialog.
- Set it up like this:

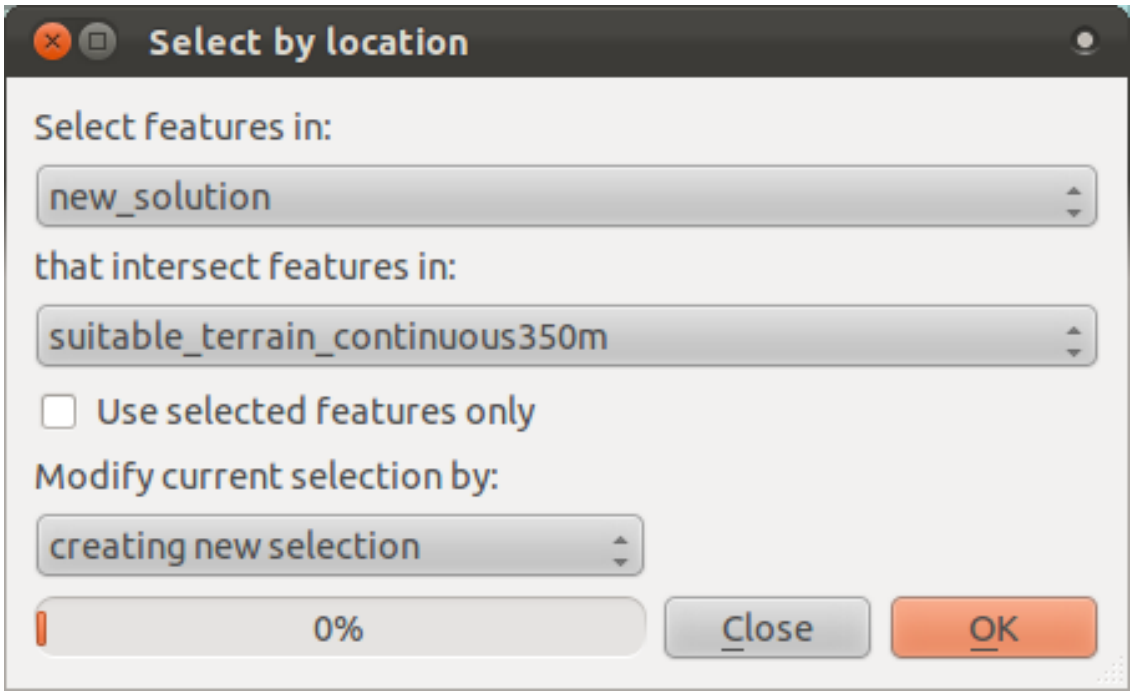



- Use the *suitable_terrain* layer with `10` segments and a buffer distance of `-350`. (The distance is automatically in meters because your map is using a projected CRS.)
- Save the output in `exercise_data/residential_development/` as `suitable_terrain_continuous350m.shp`.

Your results will look like this:

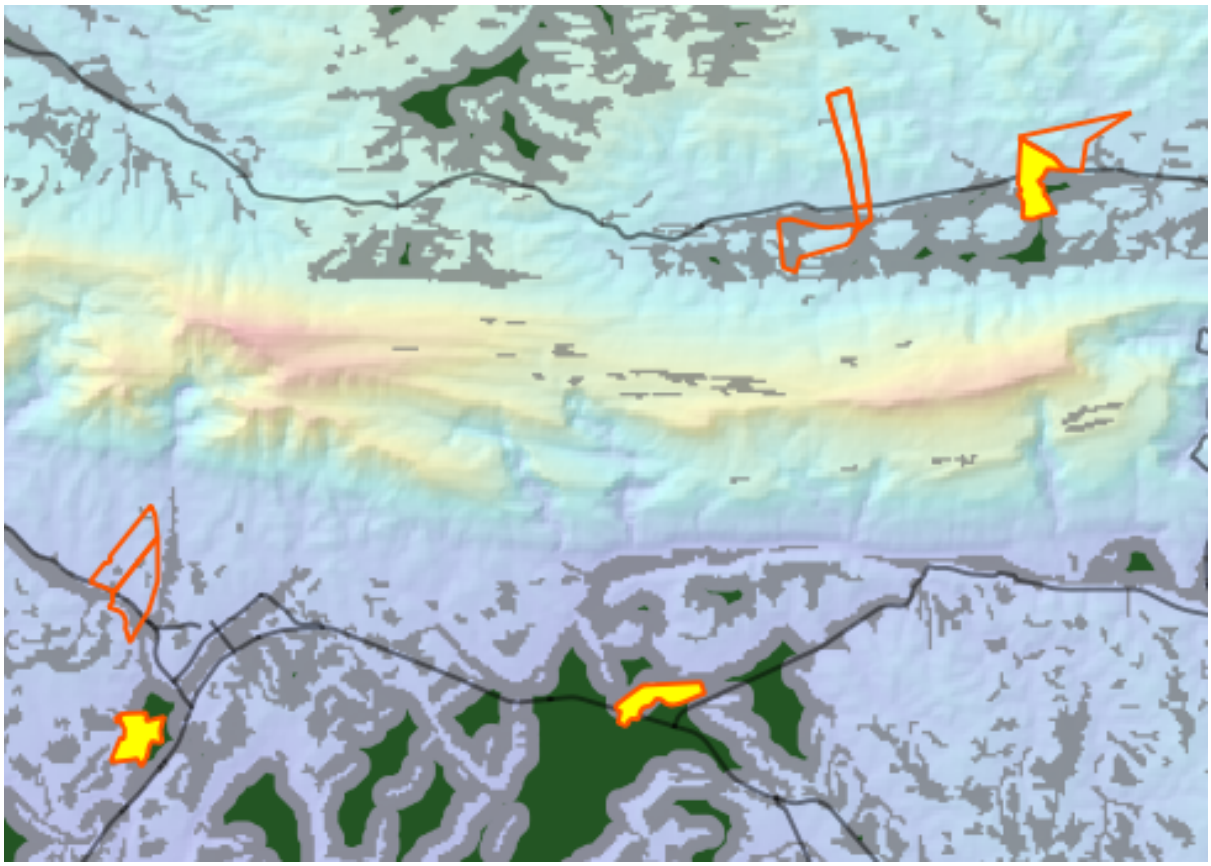


- Now use the *Select by Location* tool (*Vector* → *Research Tools* → *Select by location*).
- Set up like this:




- Select features in *new_solution* that intersect features in *suitable_terrain_continuous350m.shp*.

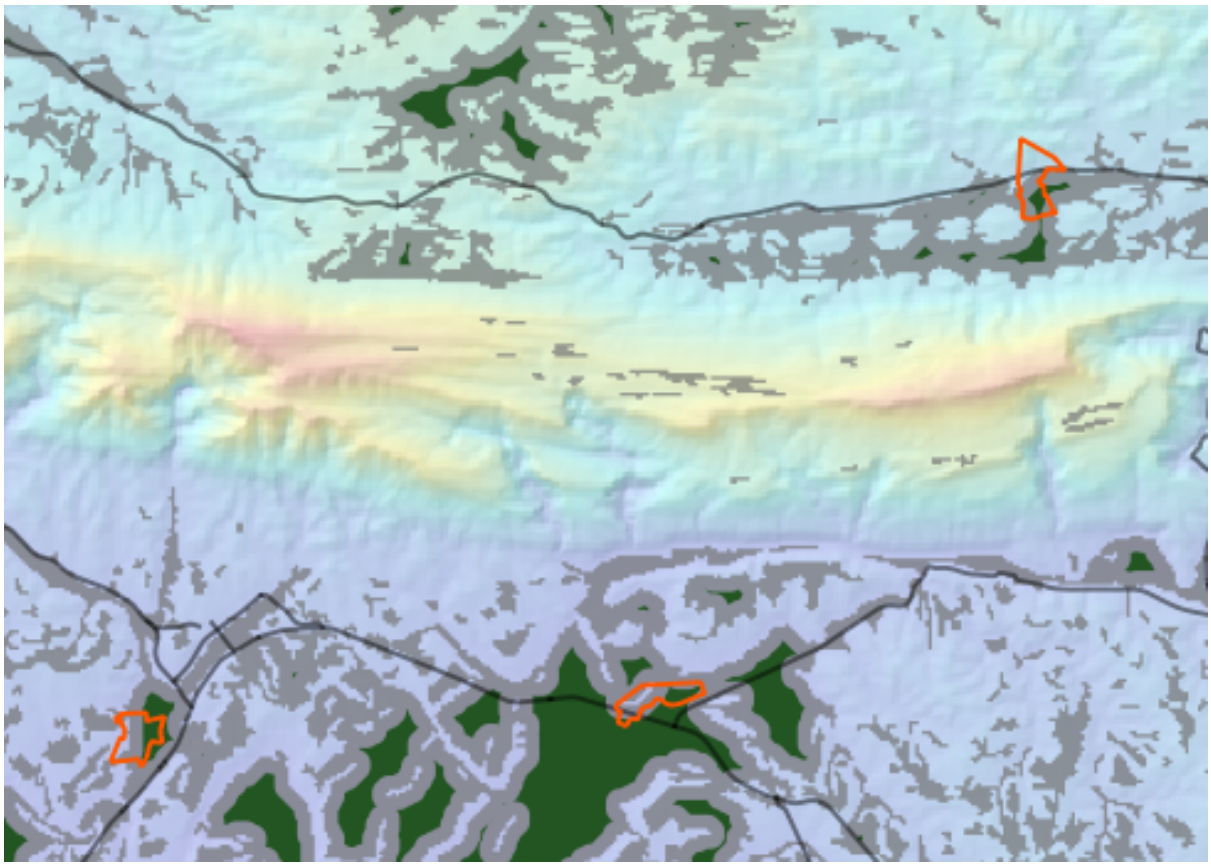
This is the result:



The yellow farms are selected.

- Save the selection under `exercise_data/residential_development/` as `final_answer.shp`.

The farms that meet the revised criteria are here:

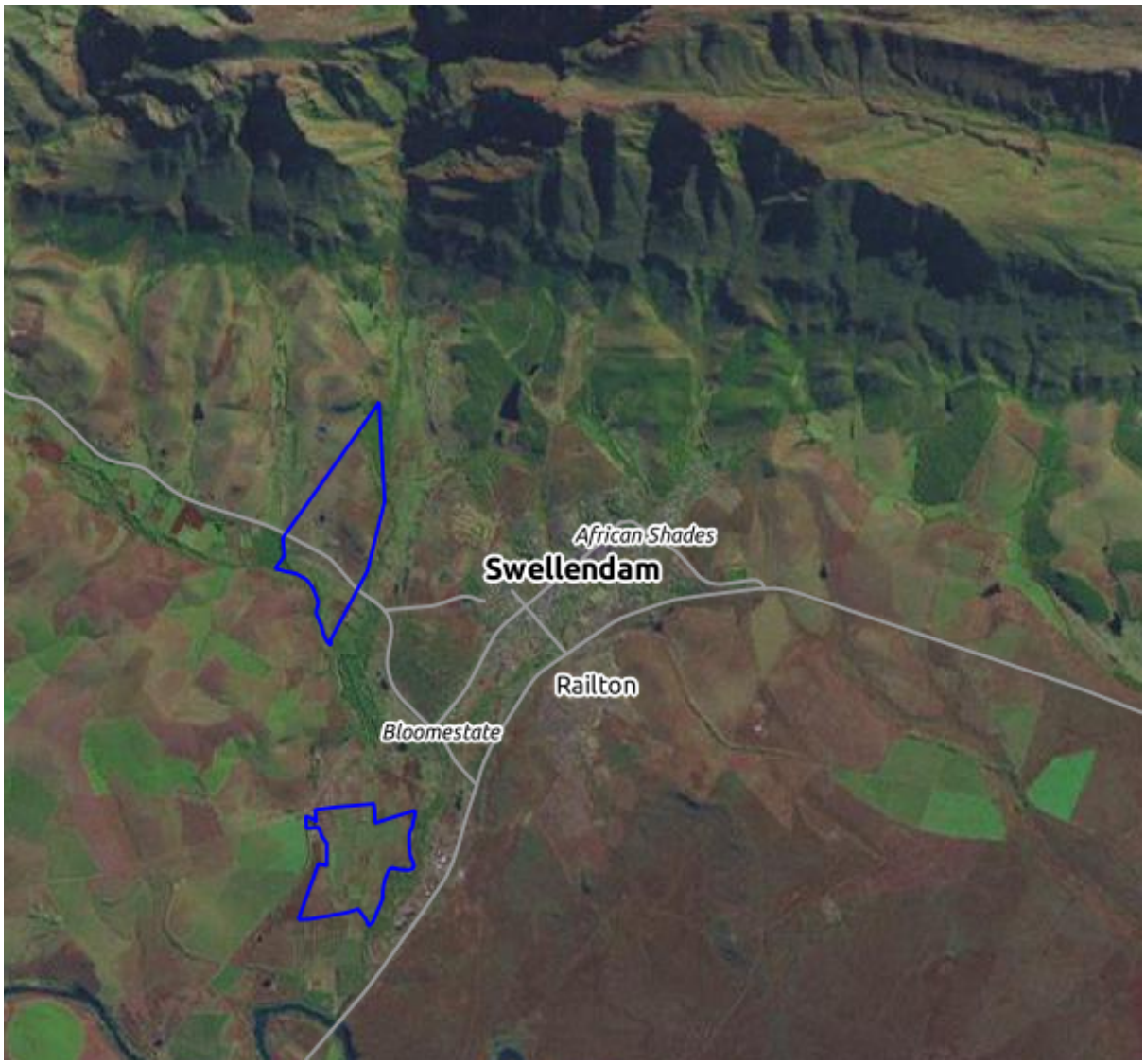


*Back to text*

## 18.12 Results for *WMS*

### 18.12.1 *Adding Another WMS Layer*

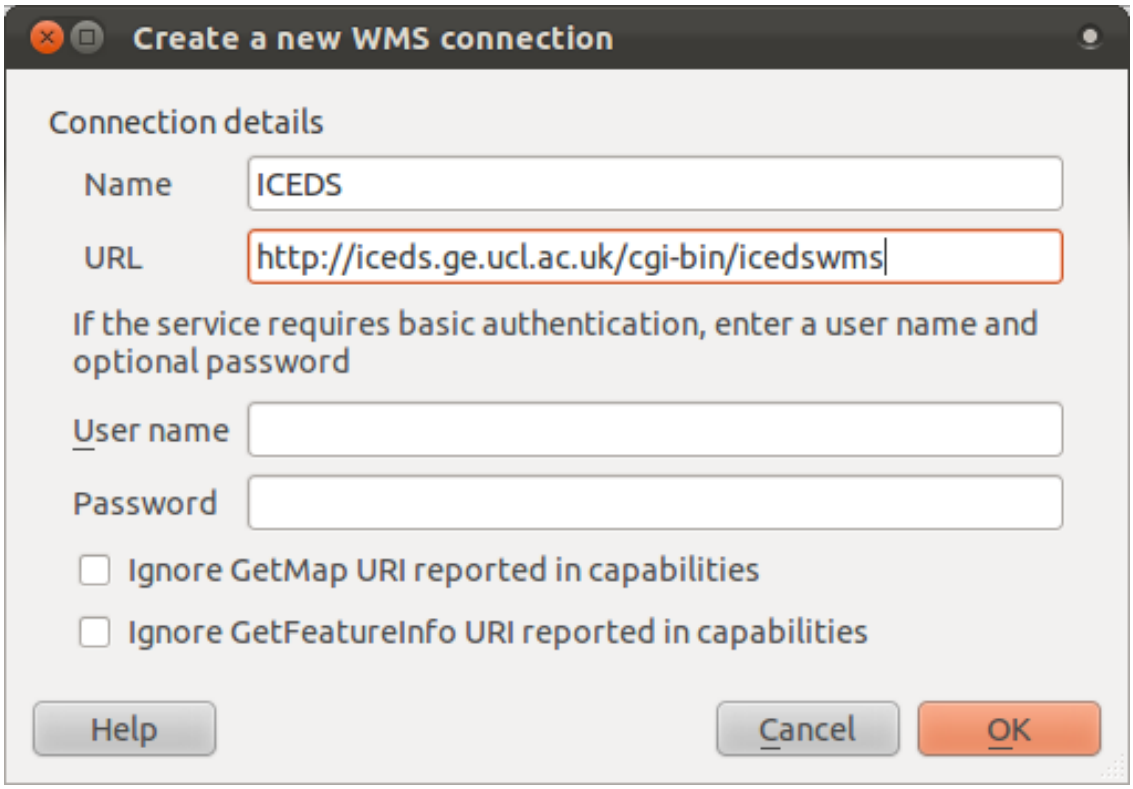
Your map should look like this:

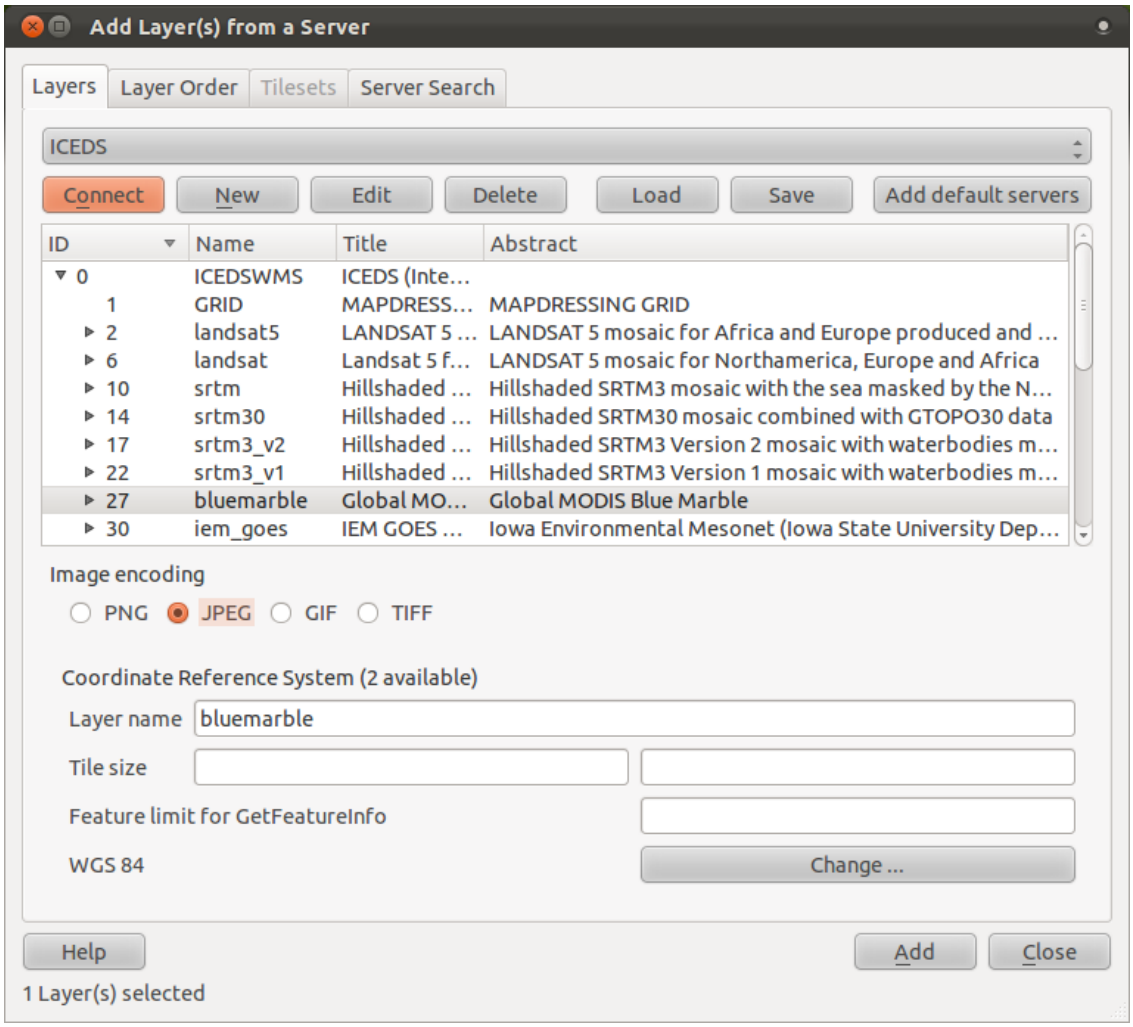



*Back to text*

### 18.12.2 *Adding a New WMS Server*

- Use the same approach as before to add the new server and the appropriate layer as hosted on that server:


Create a new WMS connection
Connection details
Name
ICEDS
URL
http://iceds.ge.ucl.ac.uk/cgi-bin/icedswms
If the service requires basic authentication, enter a user name and optional password
User name
Password
Ignore GetMap URI reported in capabilities
Ignore GetFeatureInfo URI reported in capabilities
Help
Cancel
OK


Add Layer(s) from a Server
Layers
Layer Order
Tilesets
Server Search
ICEDS
Connect
New
Edit
Delete
Load
Save
Add default servers
ID
Name
Title
Abstract
0 ICEDSWMS ICEDS (Inte...
1 GRID MAPDRESS... MAPDRESSING GRID
2 landsat5 LANDSAT 5 ... LANDSAT 5 mosaic for Africa and Europe produced and ...
6 landsat Landsat 5 f... LANDSAT 5 mosaic for Northamerica, Europe and Africa
10 srtm Hillshaded ... Hillshaded SRTM3 mosaic with the sea masked by the N...
14 srtm30 Hillshaded ... Hillshaded SRTM30 mosaic combined with GTOPO30 data
17 srtm3_v2 Hillshaded ... Hillshaded SRTM3 Version 2 mosaic with waterbodies m...
22 srtm3_v1 Hillshaded ... Hillshaded SRTM3 Version 1 mosaic with waterbodies m...
27 bluemarble Global MO... Global MODIS Blue Marble
30 iem_goes IEM GOES ... Iowa Environmental Mesonet (Iowa State University Dep...
Image encoding
PNG
JPEG
GIF
TIFF
Coordinate Reference System (2 available)
Layer name
bluemarble
Tile size
Feature limit for GetFeatureInfo
WGS 84
Change ...
Help
Add
Close
1 Layer(s) selected

- If you zoom into the Swellendam area, you'll notice that this dataset has a low resolution:

Therefore, it's better not to use this data for the current map. The Blue Marble data is more suitable at global or national scales.

*Back to text*

### 18.12.3 *Finding a WMS Server*

You may notice that many WMS servers are not always available. Sometimes this is temporary, sometimes it is permanent. An example of a WMS server that worked at the time of writing is the *World Mineral Deposits* WMS at `http://apps1.gdr.nrcan.gc.ca/cgi-bin/worldmin_en-ca_ows`. It does not require fees or have access constraints, and it is global. Therefore, it does satisfy the requirements. Keep in mind, however, that this is merely an example. There are many other WMS servers to choose from.

*Back to text*

## 18.13 Results for *Spatial Queries*

### 18.13.1 *The Units Used in Spatial Queries*

The units being used by the example query are degrees, because the CRS that the layer is using is WGS 84. This is a Geographic CRS, which means that its units are in degrees. A Projected CRS, like the UTM projections, is in meters.

Remember that when you write a query, you need to know which units the layer's CRS is in. This will allow you to write a query that will return the results that you expect.

*Back to text*

### 18.13.2 *Creating a Spatial Index*

```
CREATE INDEX cities_geo_idx
  ON cities
  USING gist (the_geom);
```

*Back to text*

## 18.14 Results for *Geometry Construction*

### 18.14.1 *Creating Linestrings*

```
alter table streets add column the_geom geometry;
alter table streets add constraint streets_geom_point_chk check
     (st_geometrytype(the_geom) = 'ST_LineString'::text OR the_geom IS NULL);
insert into geometry_columns values ('','public','streets','the_geom',2,4326,
     'LINESTRING');
create index streets_geo_idx
  on streets
  using gist
  (the_geom);
```

*Back to text*

### 18.14.2 *Linking Tables*

```
delete from people;
alter table people add column city_id int not null references cities(id);
```

(capture cities in QGIS)

```
insert into people (name,house_no, street_id, phone_no, city_id, the_geom)
   values ('Faulty Towers',
           34,
           3,
           '072 812 31 28',
           1,
           'SRID=4326;POINT(33 33)');

insert into people (name,house_no, street_id, phone_no, city_id, the_geom)
   values ('IP Knightly',
           32,
           1,
           '071 812 31 28',
           1,
           'SRID=4326;POINT(32 -34)');

insert into people (name,house_no, street_id, phone_no, city_id, the_geom)
   values ('Rusty Bedsprings',
           39,
           1,
           '071 822 31 28',
           1,
           'SRID=4326;POINT(34 -34)');
```

If you're getting the following error message:

```
ERROR:  insert or update on table "people" violates foreign key constraint
        "people_city_id_fkey"
DETAIL: Key (city_id)=(1) is not present in table "cities".
```

then it means that while experimenting with creating polygons for the cities table, you must have deleted some of them and started over. Just check the entries in your cities table and use any `id` which exists.

*Back to text*

## 18.15 Results for *Simple Feature Model*

### 18.15.1 *Populating Tables*

```
create table cities (id serial not null primary key,
                     name varchar(50),
                     the_geom geometry not null);
 alter table cities
 add constraint cities_geom_point_chk
 check (st_geometrytype(the_geom) = 'ST_Polygon'::text );
```

*Back to text*

### 18.15.2 Populate the Geometry_Columns Table

```
insert into geometry_columns values
      ('','public','cities','the_geom',2,4326,'POLYGON');
```

*Back to text*

### 18.15.3 Adding Geometry

```
select people.name,
       streets.name as street_name,
       st_astext(people.the_geom) as geometry
from   streets, people
where  people.street_id=streets.id;
```

Result:

```
       name        |   street_name    |    geometry
-------------------+------------------+---------------
 Rusty Bedsprings | High street      |
 QGIS Geek        | High street      |
 Joe Bloggs       | New Main Street  |
 IP Knightly      | QGIS Road        |
 Fault Towers     | QGIS Road        | POINT(33 -33)
(5 rows)
```

As you can see, our constraint allows nulls to be added into the database.

*Back to text*