In this White Paper, we will explore using the **Ogr2Ogr** Command Line tool, within **GDAL**, to undertake attribute and geographic queries. In addition, we will undertake several geoprocessing routines, including clipping objects and perform translation between common spatial data formats.

"GDAL is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single raster abstract data model and single vector abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing, including Ogr2Ogr."

Reference: http://www.gdal.org/index.html

The following webpage outlines the list of commands available for Ogr2Ogr, including options to Append, Clip and Select.

http://www.gdal.org/ogr2ogr.html

The following resources have been used to provide examples for this White Paper.

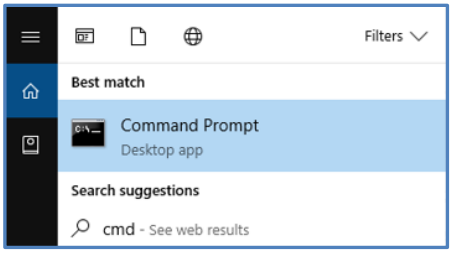
- https://github.com/dwtkns/gdal-cheat-sheet
- http://spatialmounty.blogspot.co.uk/2013/05/ogr2ogr-examples-for-spatialite-postgis.html
- https://github.com/clhenrick/OGR-SQL

The data used within this White paper is freely available from the Ordnance Survey and is used within all our DynamicMaps Open Source GIS training courses: http://www.dynamicmaps.co.uk/training/

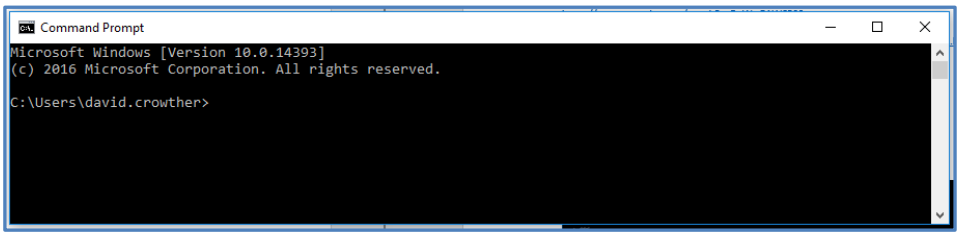
- Introduction to QGIS - http://www.dynamicmaps.co.uk/introduction-to-qgis-desktop-gis/
- Introduction to GeoServer - http://www.dynamicmaps.co.uk/introduction-to-geoserver/
- Integrated Open Source GIS - http://www.dynamicmaps.co.uk/open-source-gis-integration/

1 - Open the Ogr2Ogr Command Line

The Ogr2Ogr command line tool can be accessed using the **CMD Shell** of your PC/Server. From the Start menu, choose the CMD tool.



The Command tool will open in Shell window.

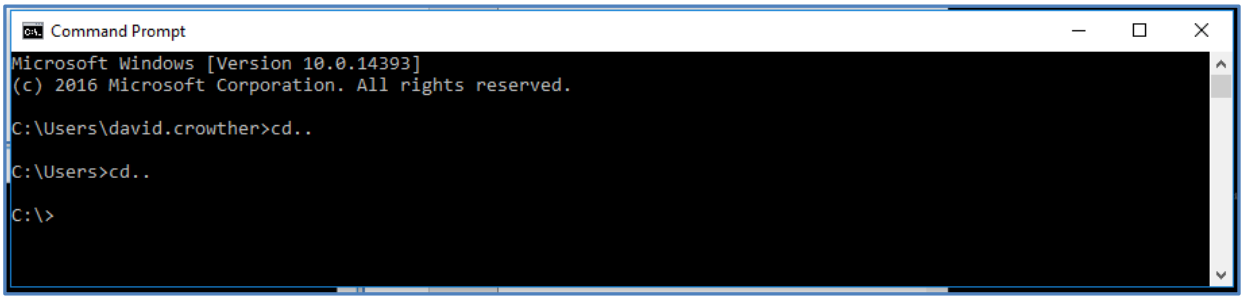


To access the Ogr2Ogr command tools, you need to navigate to the location where GDAL has been installed. In my case, I have used the installer provided by the OSGeo project which contains the GDAL tools:

https://live.osgeo.org/en/win_installers.html

and my installation is located at: **C:\OSGeo4W64\bin**

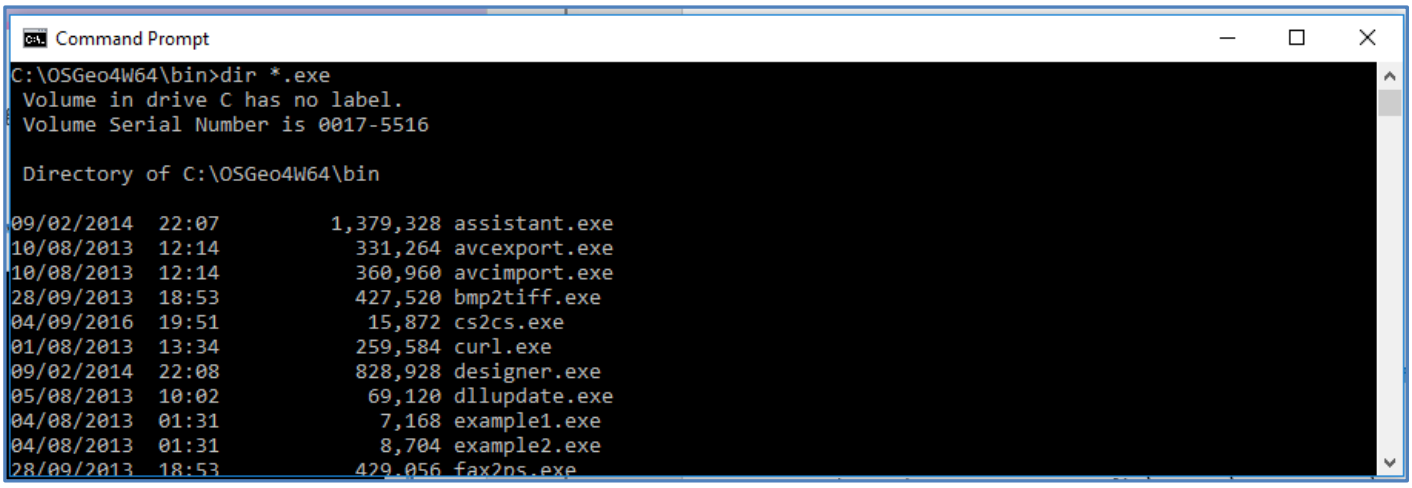
Using the command cd.. (change directory) navigate to the root of your local drive, for example C:\



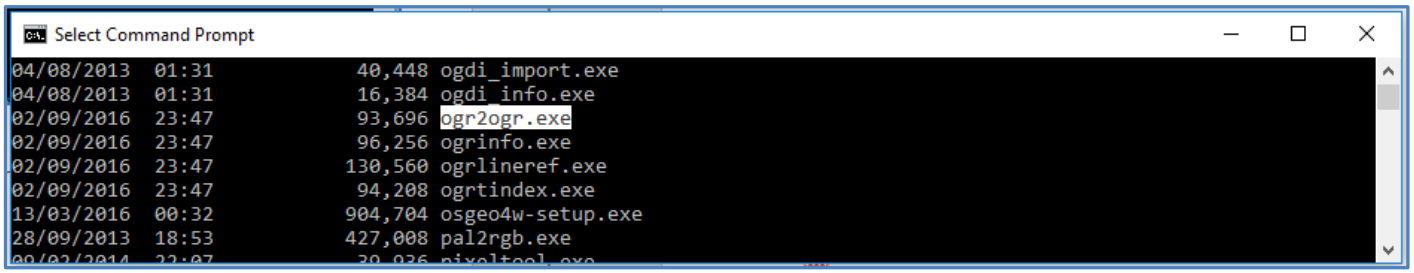
Then from the C:\ drive use the cd (change directory) command to navigate to the location where GDAL has been installed – **C:\ osgeo4w64\bin**

To access Ogr2Ogr2 you can now list the applications available within the Ogr2Ogr2 BIN folder.

> dir *.exe



One of the applications available will be Ogr2Ogr.



In addition to the Ogr2Ogr tools you can also access **Ogrinfo** which we will explore first.

2 – Using the OgrInfo Command

GDAL includes two vector based spatial tools; Ogr2Ogr which can be used to translate, clip and re-project spatial data, and OgrInfo which enables you to view the full details about a spatial file, e.g. listing fields and identifying projections.

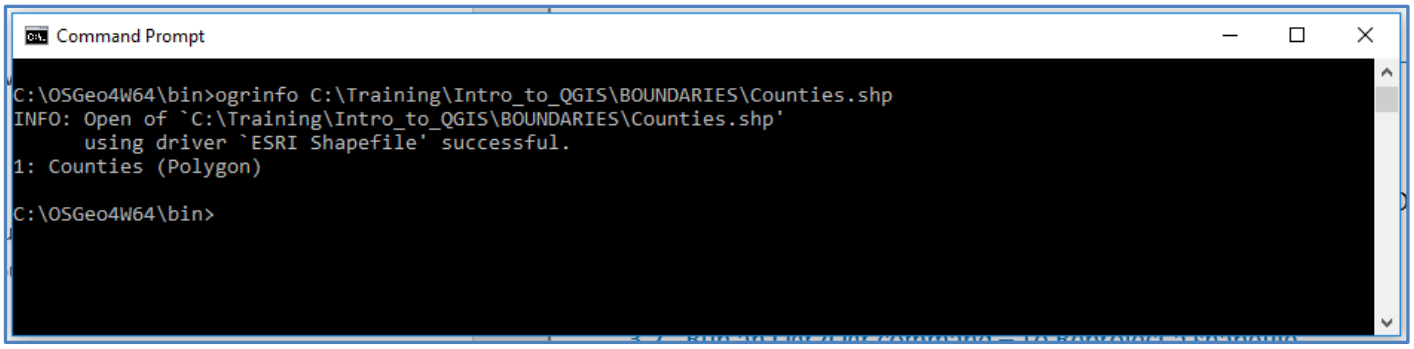
2.1 – Identify the geometry of an ESRI Shapefile

Command:

ogrinfo Location\ShapefileName.shp

Example:

ogrinfo C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp



The result shows that the **Counties.shp** file is a **Polygon** layer.

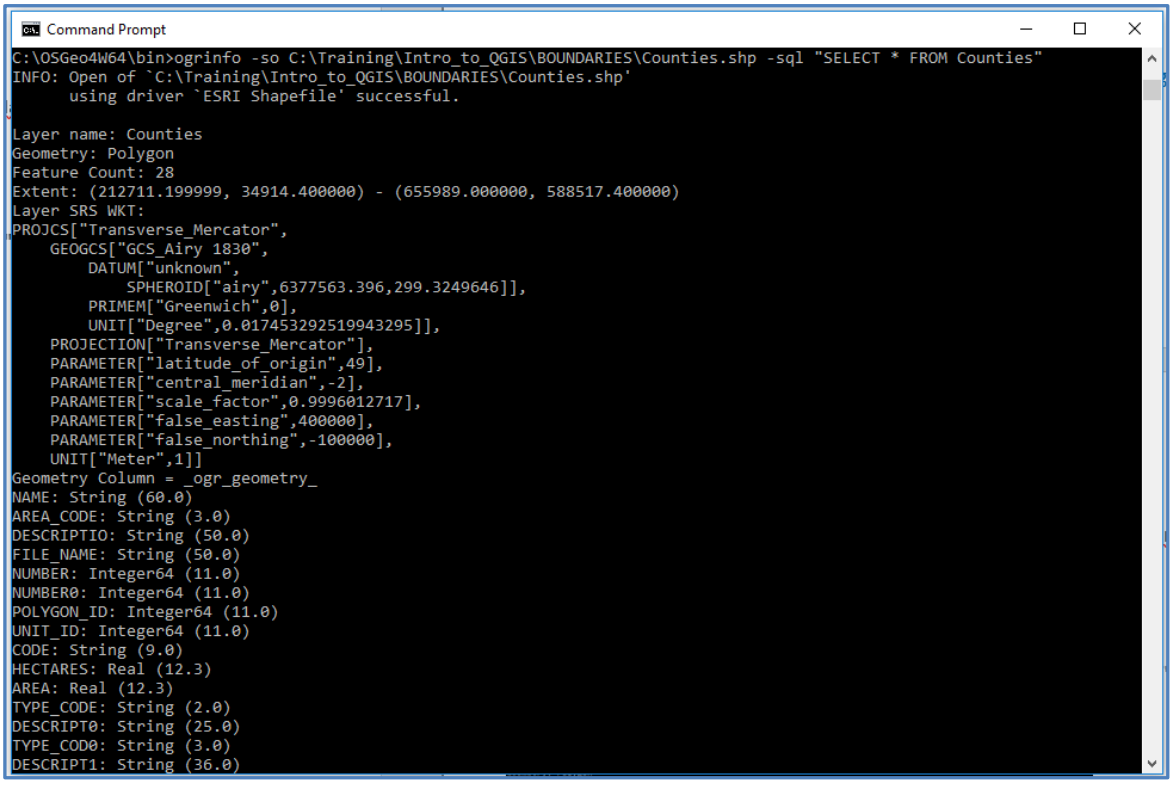
2.2 – Get full details of an ESRI Shapefile

Command:

Ogrinfo -so Location\ShapefileName.shp -sql “Select * From ShapefileName”

Example:

ogrinfo -so C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -sql "SELECT * FROM Counties"



The result provides the following details for the **Counties** layer:

Geometry Type, Spatial Extents, Projection, Field Names and Field Types.

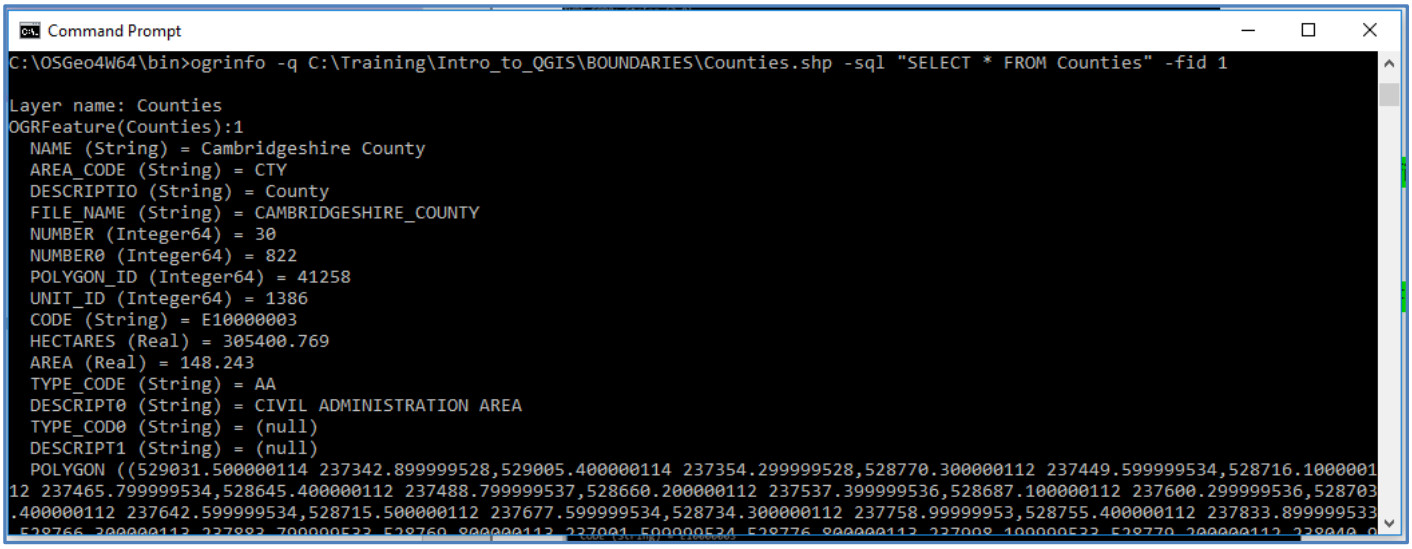
2.3 – Get full details for a Specific Record of an ESRI Shapefile

Command:

ogrinfo -q Location\ShapefileName.shp -sql "Select * From ShapefileName" -fid 1

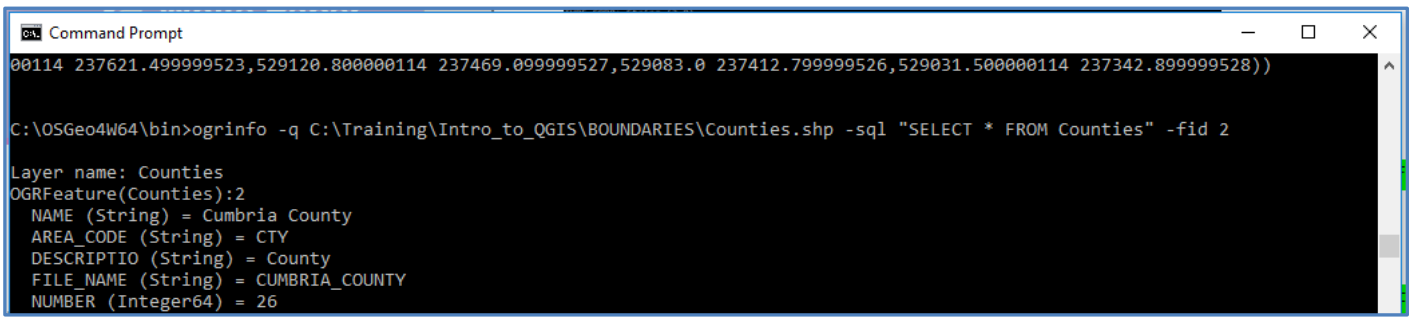
Example:

ogrinfo -q C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -sql "SELECT * FROM Counties" -fid 1



The result lists the attributes for each column including the geometry for the chosen record of the **Counties** layer.

Change FID to 2 and see the values change – now showing details for Cumbria.



3 – Using the Ogr2Ogr Command

The second OGR related GDAL tool is Ogr2Ogr, which provides options to translate spatial files, clip spatial geometry and re-project spatial datasets.

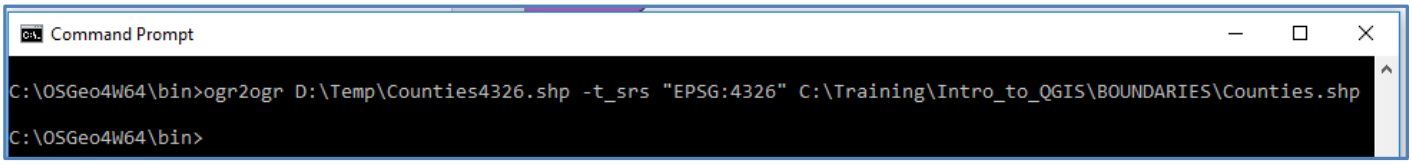
3.1 – Reproject an existing ESRI Shapefile

Command:

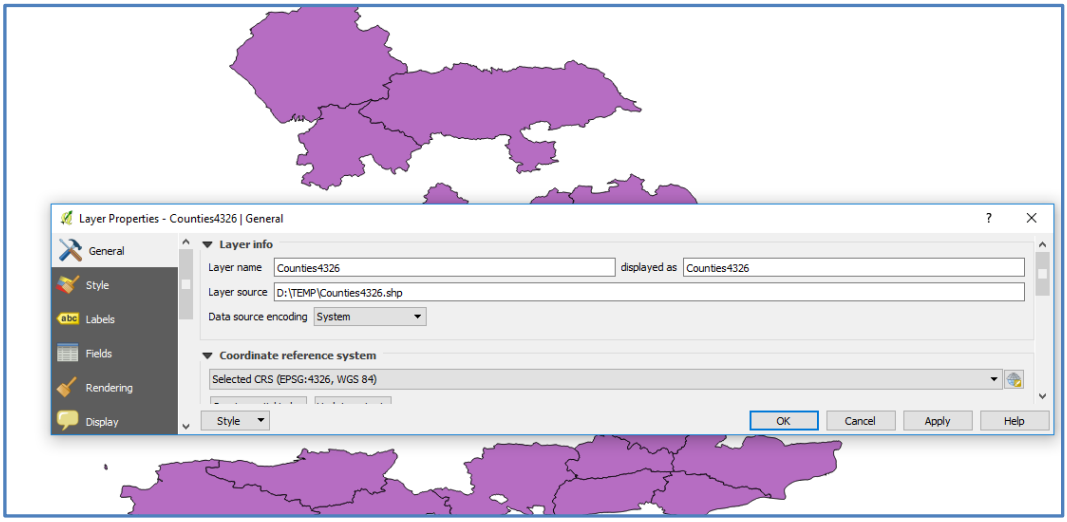
ogr2ogr Location\OutputShapefileName.shp -t_srs "EPSG:4326" Location\InputShapefileName.shp

Example:

ogr2ogr D:\Temp\Counties4326.shp -t_srs "EPSG:4326"
C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp



The result will convert the original Counties Shapefile, projected as British National Grid (27700) to a new Shapefile called Counites4326 now projected as WGS 84 (4326).



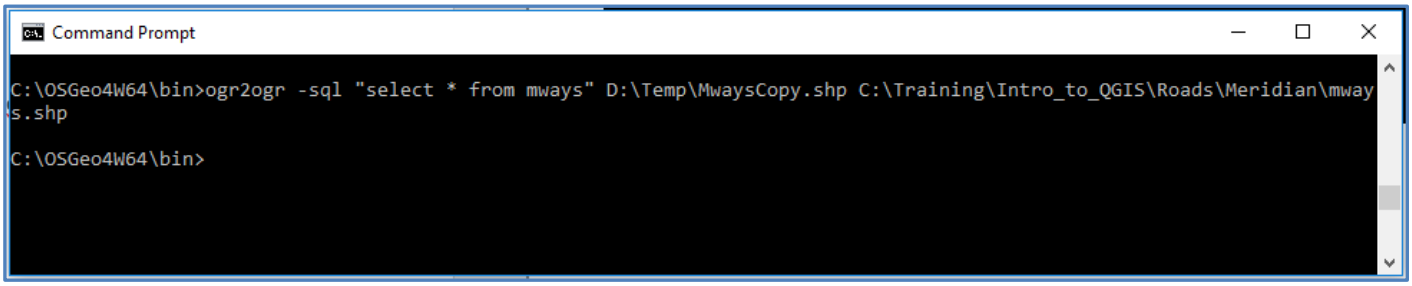
3.2 – Save a Copy of an ESRI Shapefile

Command:

ogr2ogr -sql "select * from Input LayerName)" Location\OutputFileName.shp Location\InputShapefileName.shp

Example:

ogr2ogr -sql "select * from mways" D:\Temp\MwaysCopy.shp C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp



The result will be a saved copy of the original input Shapefile, with the same projection and attributes.

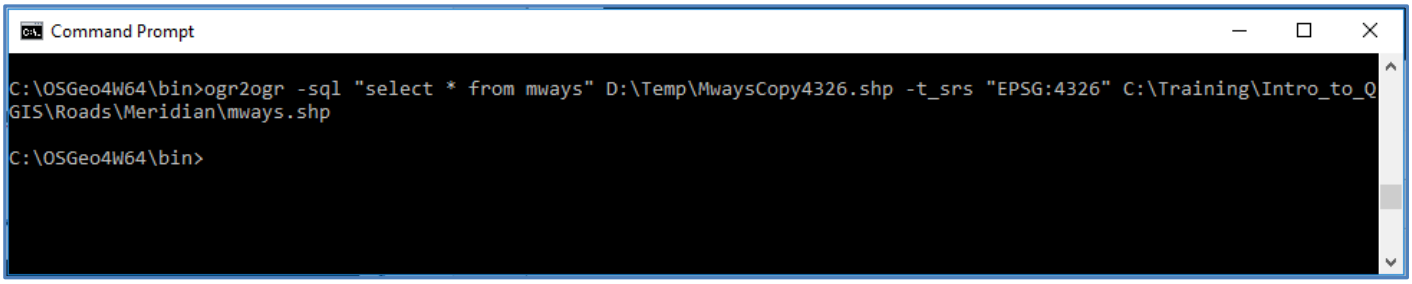
3.3 – Save a Copy of an ESRI Shapefile with a new Projection

Command:

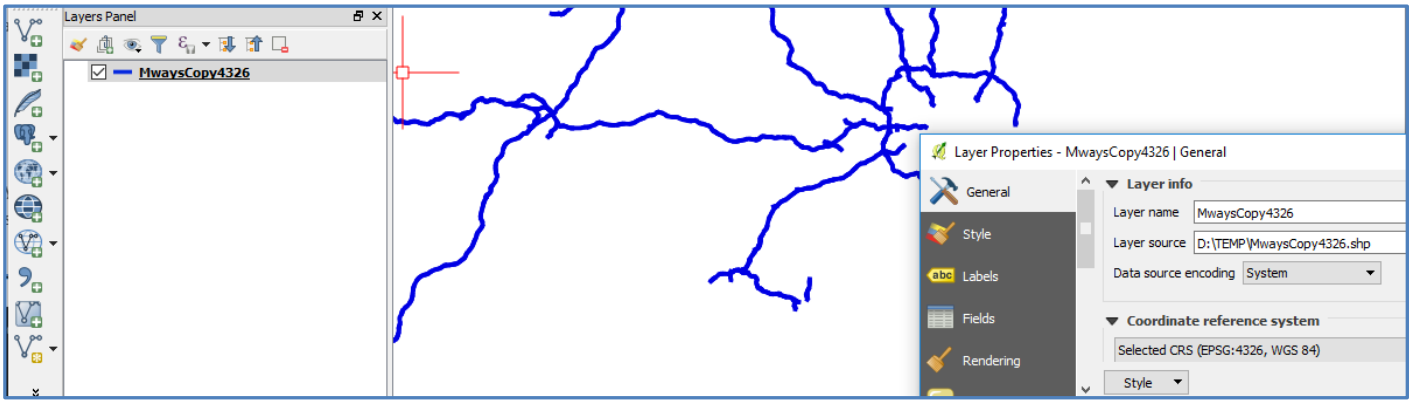
ogr2ogr -sql "select * from Input LayerName)" Location\OutputFileName.shp -t_srs "EPSG:4326" Location\InputShapefileName.shp

Example:

ogr2ogr -sql "select * from mways" D:\Temp\MwaysCopy4326.shp -t_srs "EPSG:4326" C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp



The result will be a saved copy of the original input Shapefile, but now re-projected to 4326.



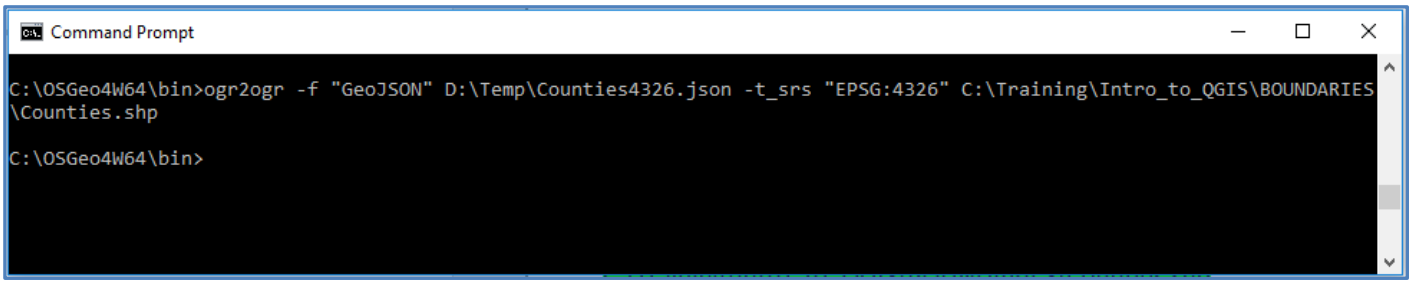
3.4 – Translate an ESRI Shapefile to GeoJSON

Command:

ogr2ogr -f "GeoJSON" Location\OutputFileName.json -t_srs "EPSG:4326" Location\InputShapefileName.shp

Example:

ogr2ogr -f "GeoJSON" D:\Temp\Counties4326.json -t_srs "EPSG:4326"
C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp



The result will translate the original Counties Shapefile into a GeoJSON (json) file which has also been re-projected to WGS 84 (4326).

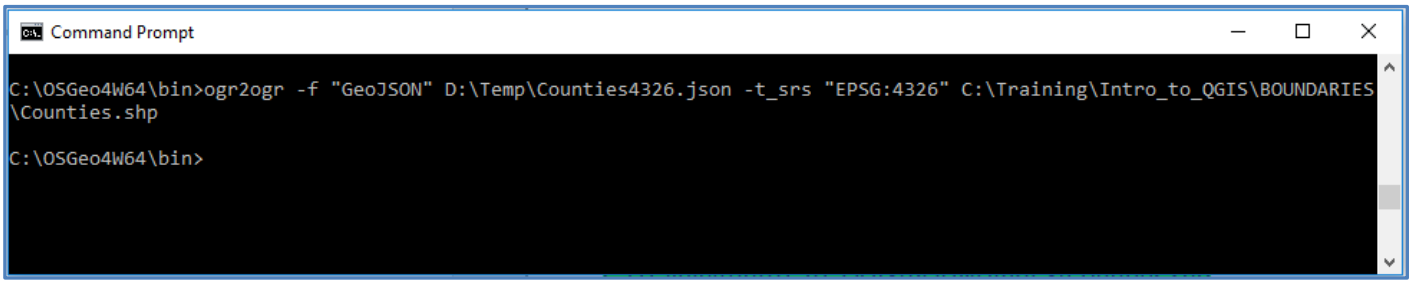
3.5 – Translate an ESRI Shapefile to KML and re-project to 4326

Command:

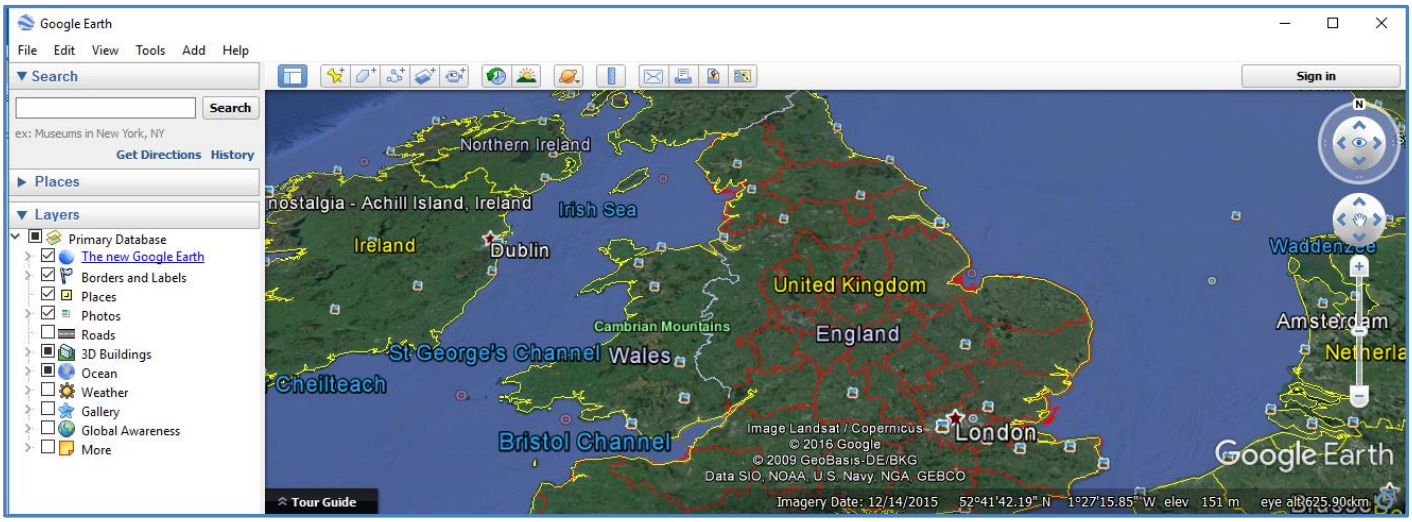
ogr2ogr -f "KML" Location\OutputFileName.kml -t_srs "EPSG:4326" Location\InputShapefileName.shp -dsco DescriptionField='Fieldname'

Example:

ogr2ogr -f "KML" D:\Temp\KMLCounties.kml C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -dsco DescriptionField='DESCRIPTIO'



The result will translate the original Counties Shapefile into a KML file which can then be opened into Google Earth.



Note – to be a valid KML, the input data needs to have a Name and a Description field.

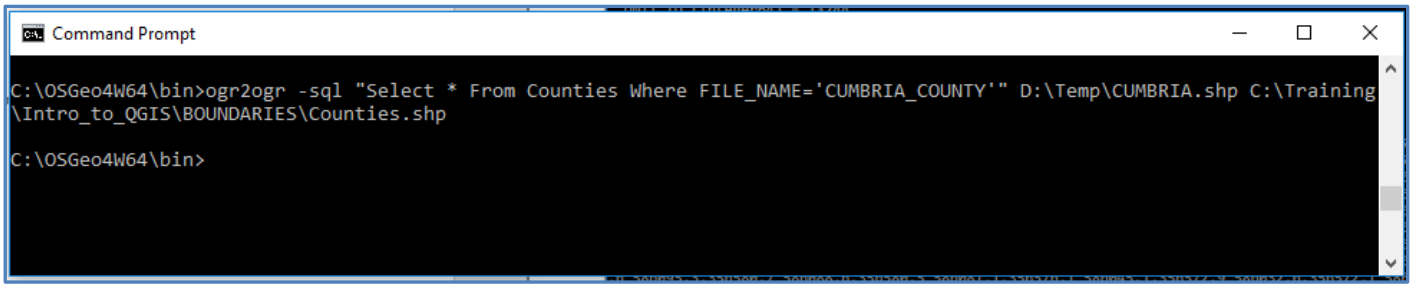
3.6 – Select Records and Create a New Shapefile

Command:

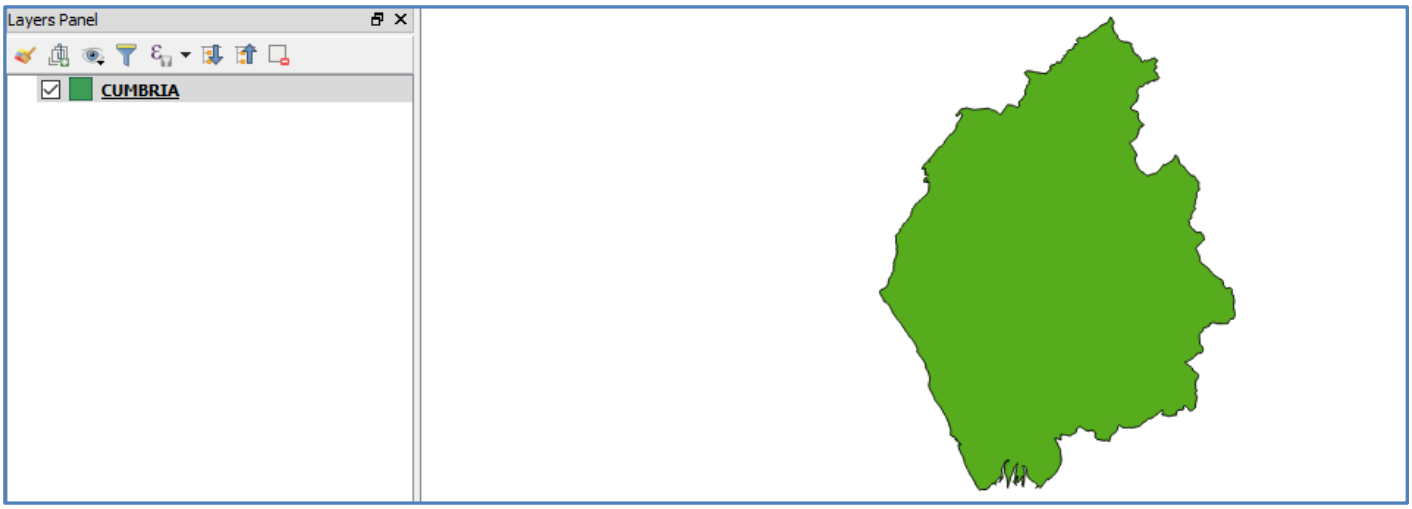
ogr2ogr -sql "Select * From Input LayerName Where Fieldname='Value'" Location\OutputFileName.shp Location\InputShapefileName.shp

Example:

ogr2ogr -sql "Select * From Counties Where FILE_NAME='CUMBRIA_COUNTY'" D:\Temp\CUMBRIA.shp C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp



The result will be a saved copy of the original input Shapefile, but with only the selected features in the output file.



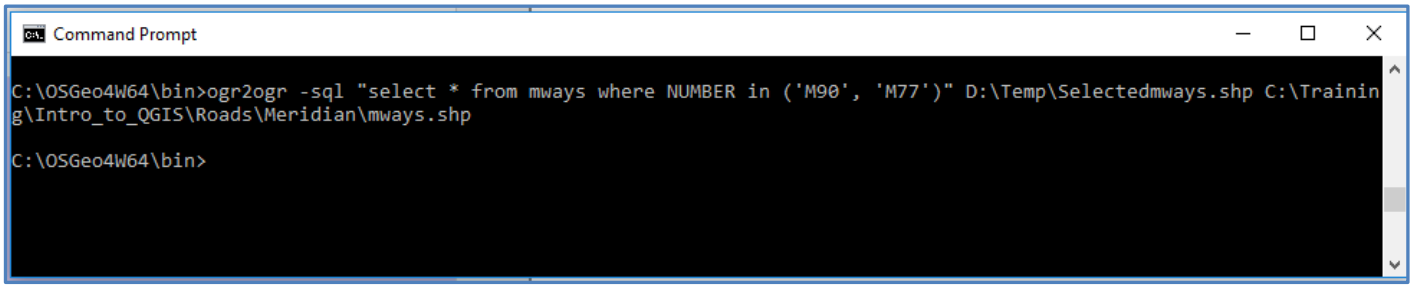
3.7 – Select Multiple Records and Create a New Shapefile

Command:

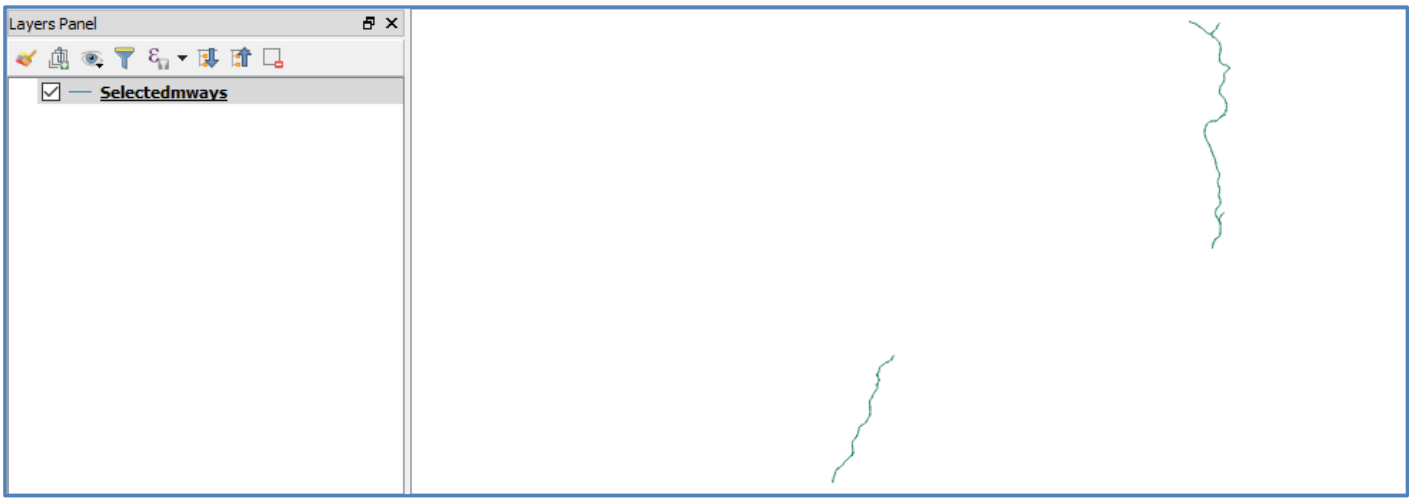
ogr2ogr -sql "Select * From Input LayerName Where type in ('Value1', 'Value2')" Location\OutputFileName.shp Location\InputShapefileName.shp

Example:

ogr2ogr -sql "select * from mways where NUMBER in ('M90', 'M77')" D:\Temp\Selectedmways.shp C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp



The result will be a saved copy of the original input Shapefile, but with only the selected features in the output file.



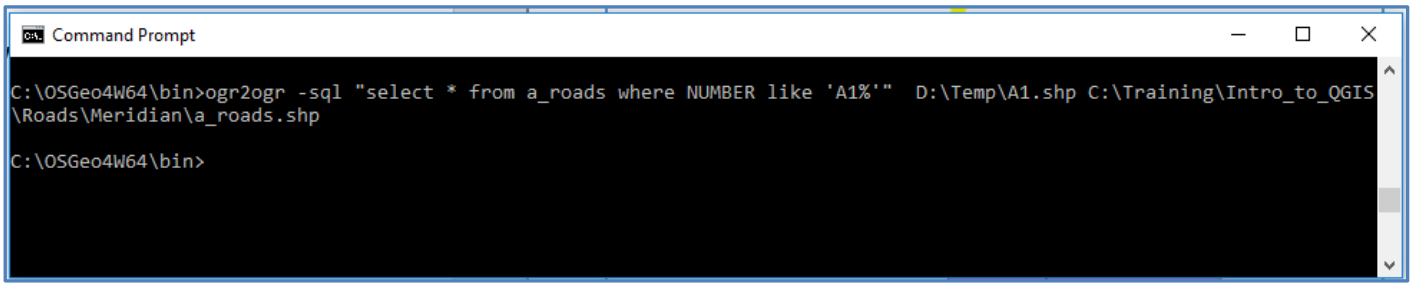
3.8 – Select Multiple Records using a LIKE command and Create a New Shapefile

Command:

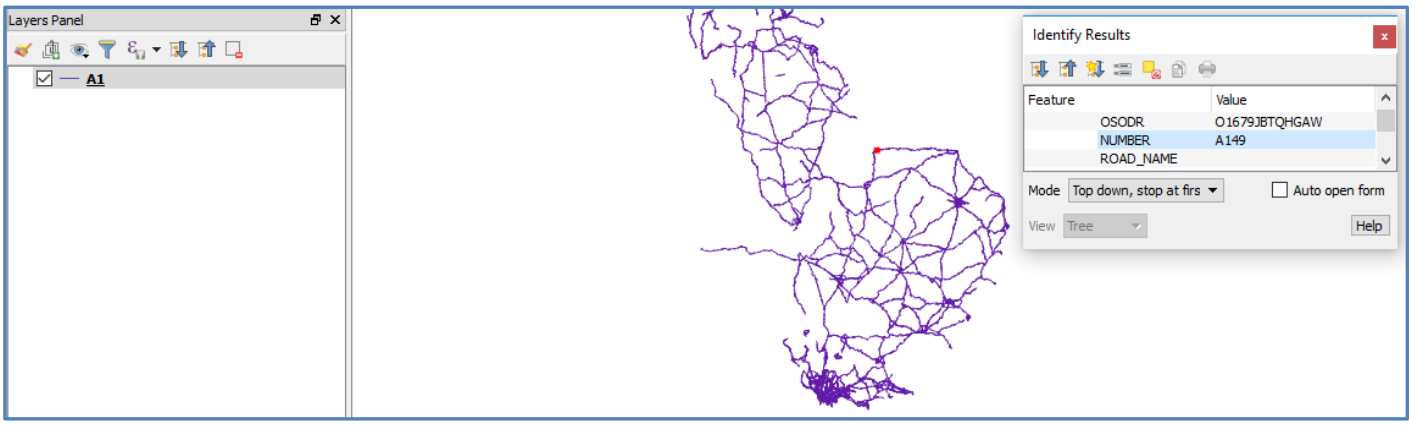
ogr2ogr -sql "Select * From Input LayerName Where field like '%value'" Location\OutputFileName.shp Location\InputShapefileName.shp

Example:

ogr2ogr -sql "select * from a_roads where NUMBER like 'A1%'" D:\Temp\A1.shp C:\Training\Intro_to_QGIS\Roads\Meridian\a_roads.shp



The result will be a saved copy of the original input Shapefile, but with only the selected features in the output file i.e. where the roadname/number starts with 'A1…'



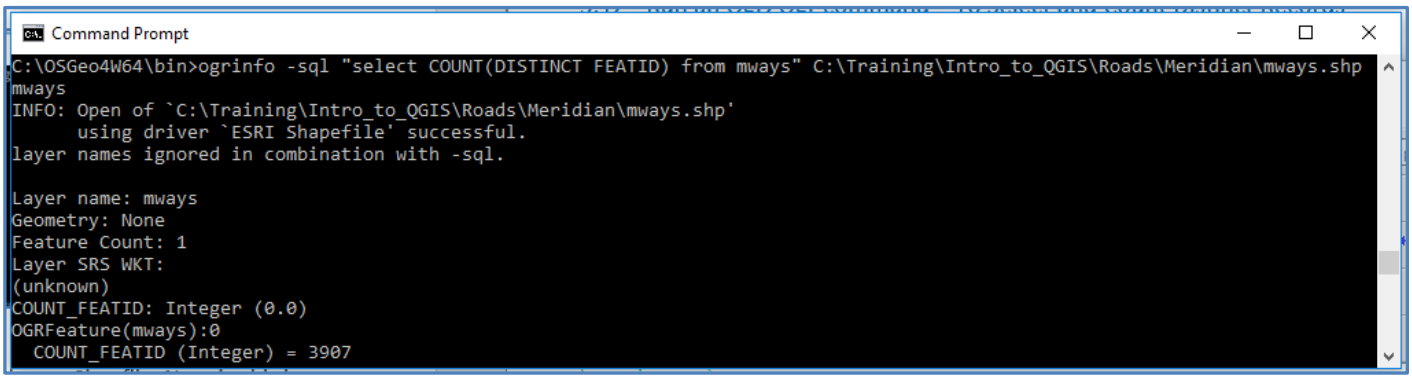
3.9 – Count Records in an ESRI Shapefile

Command:

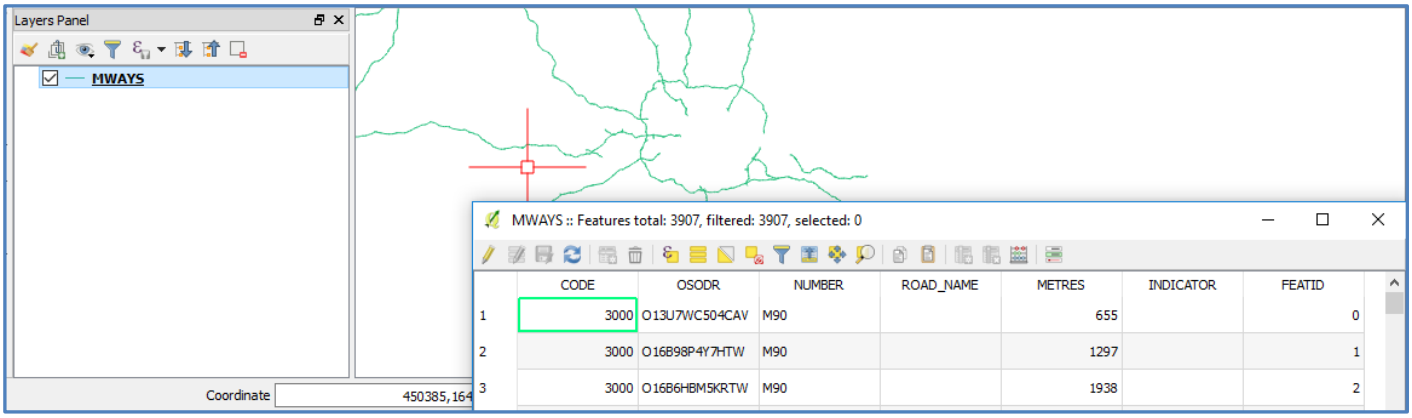
ogrinfo -sql "select COUNT(DISTINCT FieldName) from Input LayerName" Location\InputShapefileName.shp

Example:

ogrinfo -sql "select COUNT(DISTINCT FEATID) from mways"
C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp mways



The CMD Shell will update to list the number of distinct records in the chosen Shapefile. Note in this instance we have chosen to COUNT the DISTINT records using the FEATURE ID, so the result (3907) should be equal to a Count of all records.



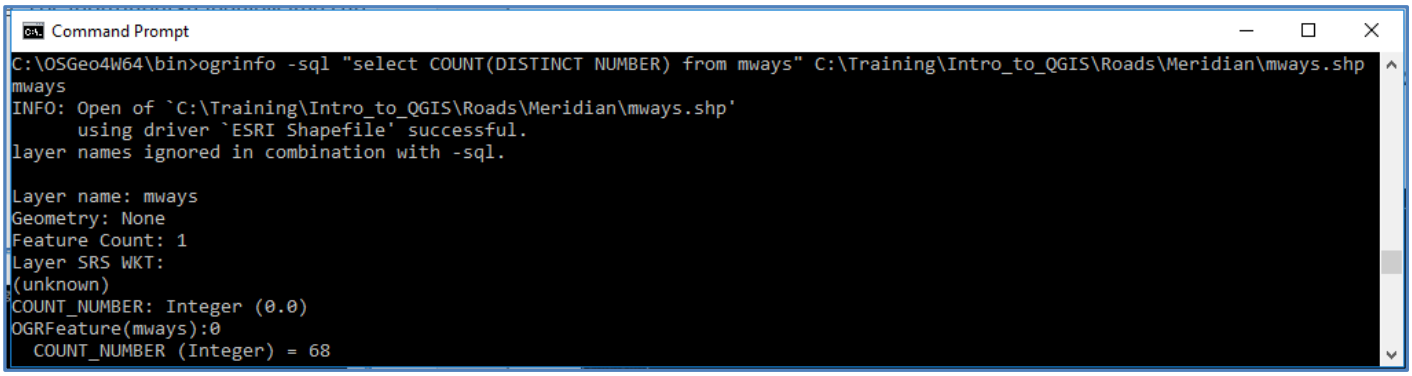
3.10 – Count DISTINCT Records in an ESRI Shapefile

Command:

ogrinfo -sql "select COUNT(DISTINCT FieldName) from Input LayerName" Location\InputShapefileName.shp

Example:

ogrinfo -sql "select COUNT(DISTINCT NUMBER) from mways"
C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp mways



The CMD Shell will update to list the number of distinct records in the chosen Shapefile for the specified field. Note in this instance we have chosen to COUNT the DISTINT records using the road NUMBER field, and as there are duplicate records with the same road Number, the output value identifies how many distinct Road Number values there are.

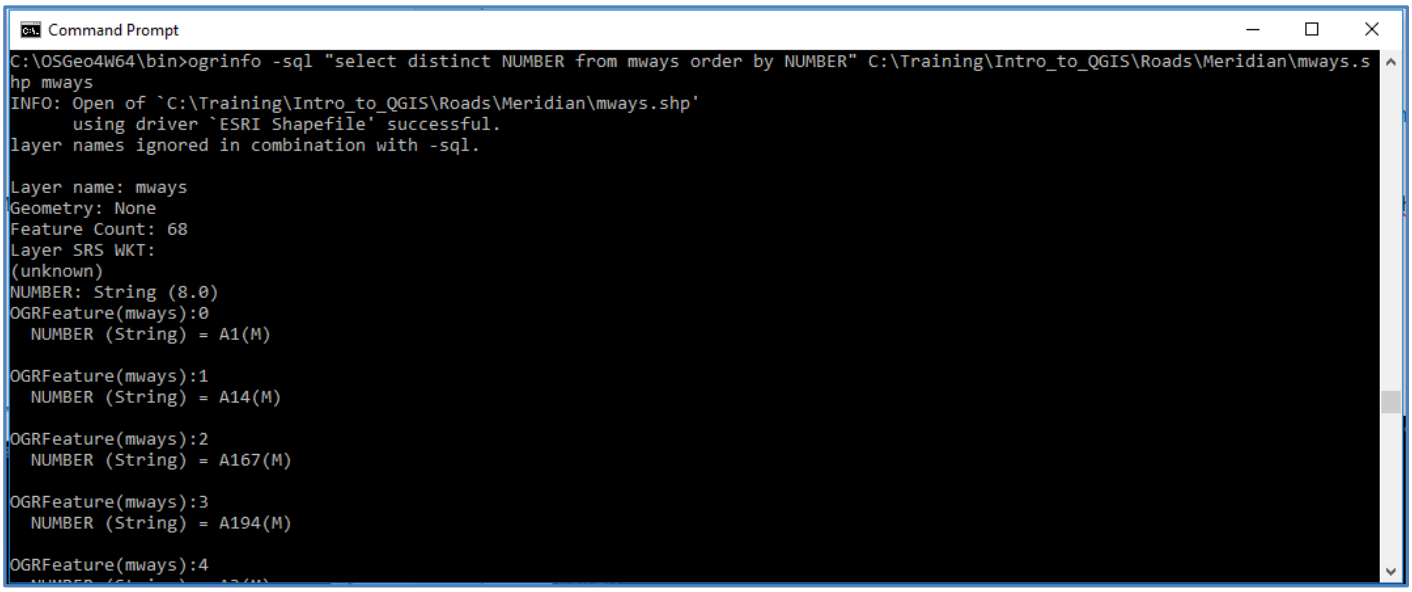
3.11 – List and Order DISTINCT Records in an ESRI Shapefile for a Chosen Field

Command:

ogrinfo -sql "select distinct FieldName from roads order by FieldName" Location\InputShapefileName.shp LayerName

Example:

ogrinfo -sql "select distinct NUMBER from mways order by NUMBER" C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp mways



The CMD Shell will update to list the Distinct Values in the chosen field (road NUMBER) and order them alphabetically in ascending order.

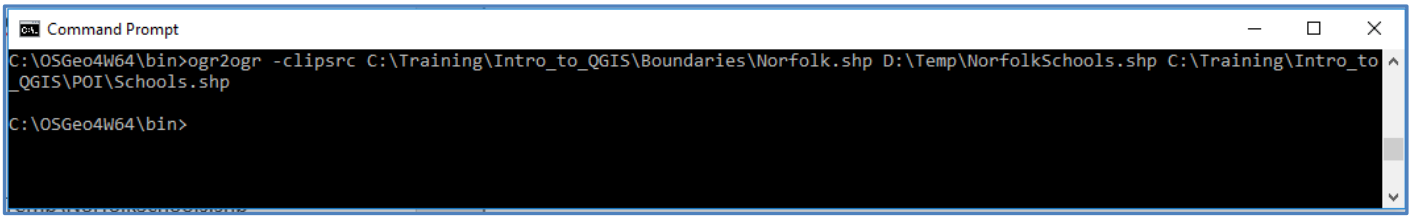
3.12 – Clip Spatial Data (Points in Polygon)

Command:

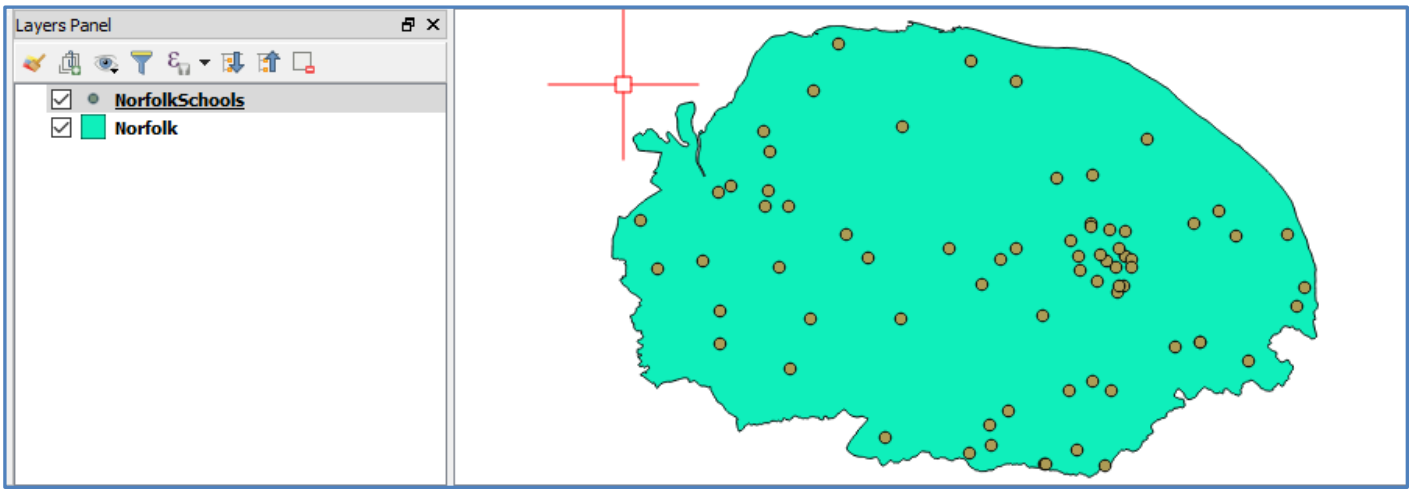
ogr2ogr -clipsrc Location\InputBoundary.shp Location\OutputClippedShapefile.shp Location\InputPoints.shp

Example:

ogr2ogr -clipsrc C:\Training\Intro_to_QGIS\Boundaries\Norfolk.shp D:\Temp\NorfolkSchools.shp C:\Training\Intro_to_QGIS\POI\Schools.shp



The result will be a new Shapefile which only contains the School Points which are within the Norfolk Boundary Clip area.



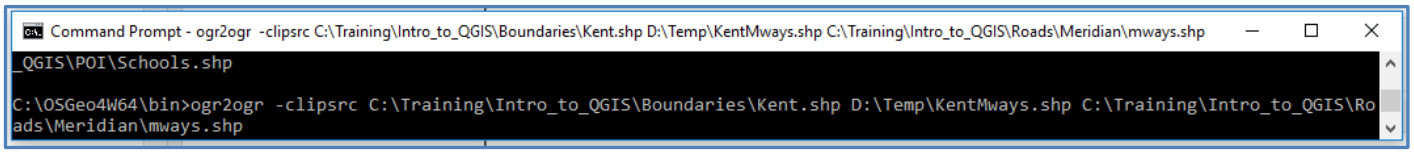
3.12 – Clip Spatial Data (Lines in Polygon)

Command:

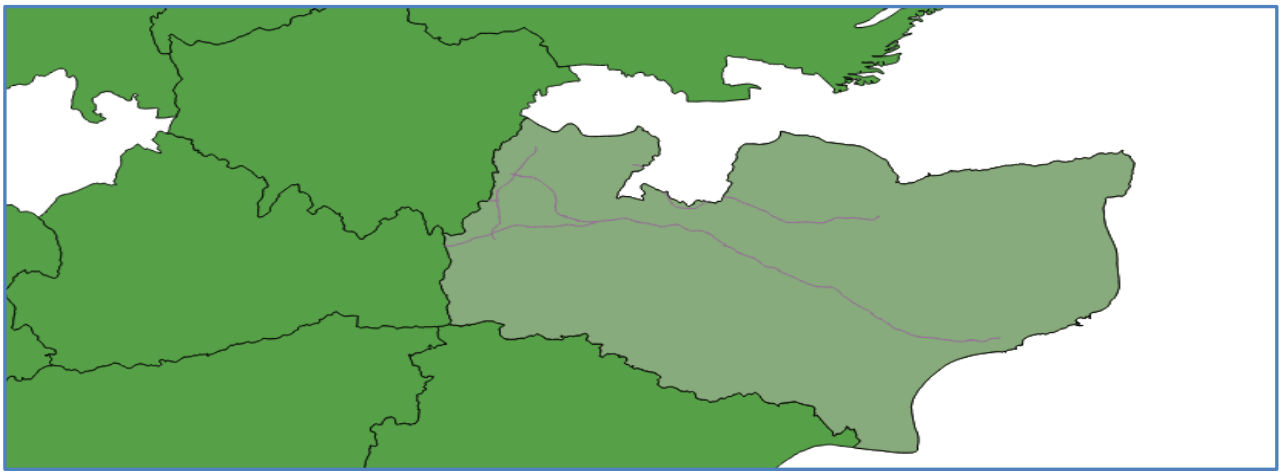
ogr2ogr -clipsrc Location\InputBoundary.shp Location\OutputClippedShapefile.shp Location\InputLines.shp

Example:

ogr2ogr -clipsrc C:\Training\Intro_to_QGIS\Boundaries\Kent.shp D:\Temp\KentMways.shp C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp



The result will be a new Shapefile which only contains the Aroad Lines which are clipped within the Kent Boundary area.



3.13 – Clip Spatial Data Using a Bounding Box

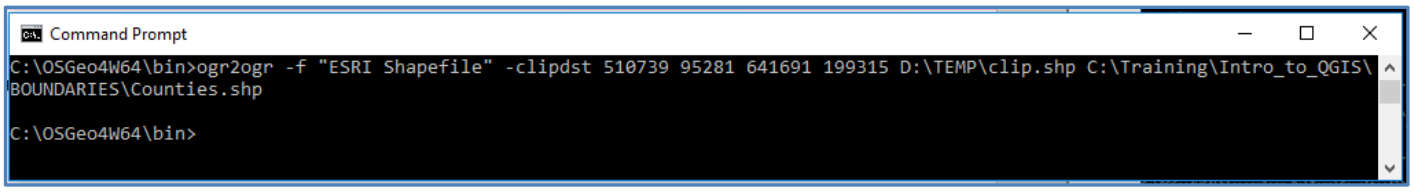
Command:

ogr2ogr -f "ESRI Shapefile" -clipdst 510739 95281 641691 199315 Location\OutputClippedShapefile.shp Location\InputShapeFile.shp

Example:

ogr2ogr -f "ESRI Shapefile" -clipdst 510739 95281 641691 199315 D:\TEMP\clip.shp C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp

Min X = 510739, Min Y = 95281, Max X = 641691 Max Y = 199315



The result will be a new Shapefile which has been clipped using the X&Y coordinates as defined in the bounding box.