

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>

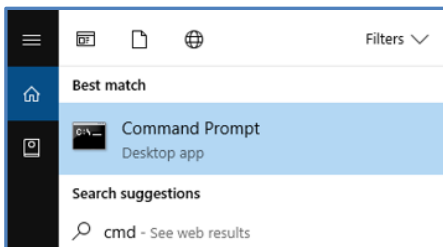
The data used within this White paper is freely available from the Ordnance Survey and is used within all our DynamicMaps Open Source GIS training courses: <http://www.dynamicmaps.co.uk/training/>

- 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/>

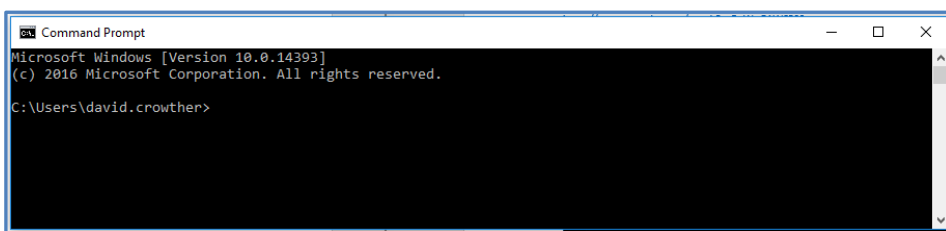


## 1 - Open the Ogr2Ogr Command Line

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.

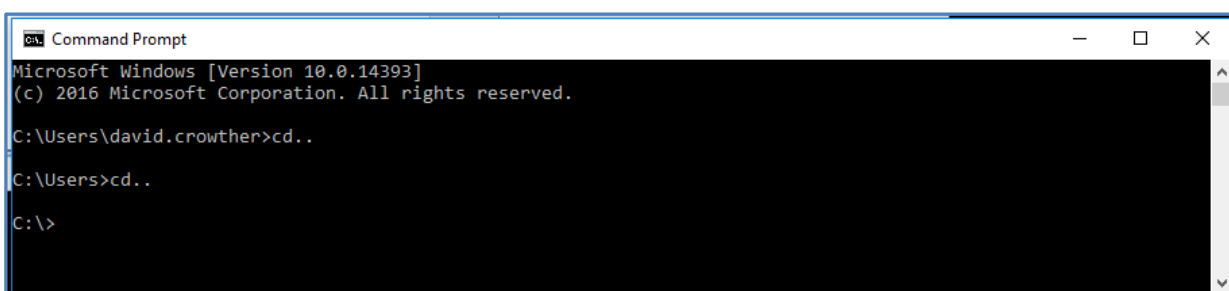


To access the Ogr2Ogr command tools, you need to navigate to the location where GDAL has been installed. In my case, I have used the installer provided by the OSGeo project which contains the GDAL tools:

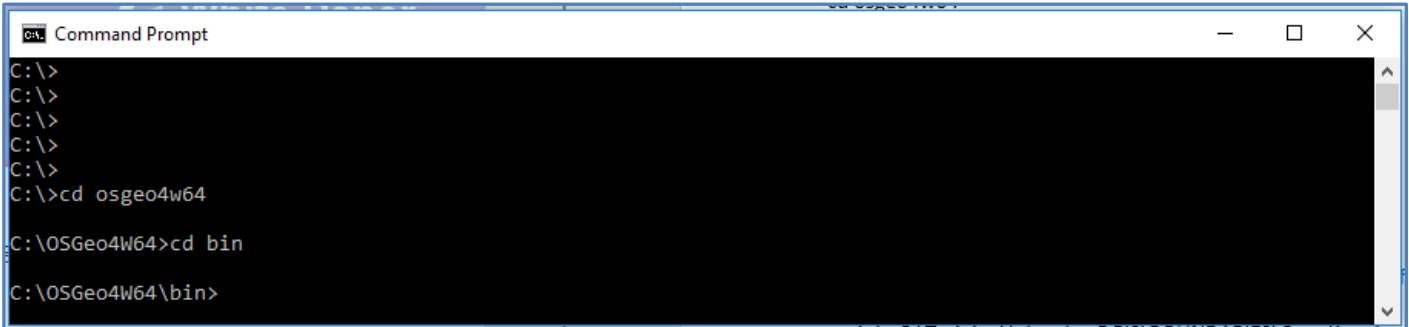
[https://live.osgeo.org/en/win\\_installers.html](https://live.osgeo.org/en/win_installers.html)

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\



```

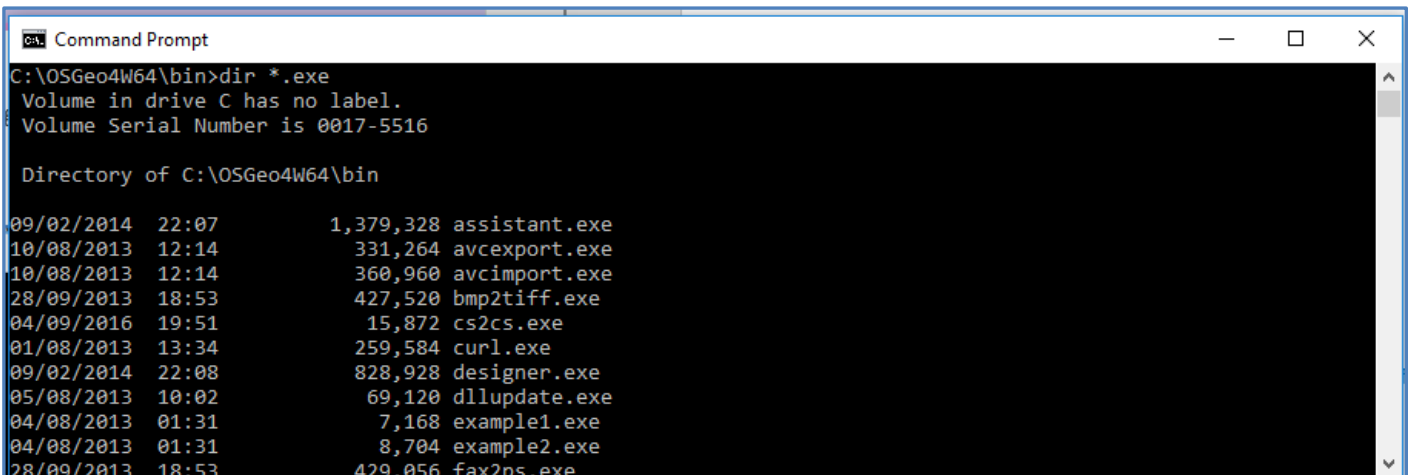
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>cd osgeo4w64

C:\OSGeo4W64>cd bin

C:\OSGeo4W64\bin>
    
```

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

> dir \*.exe



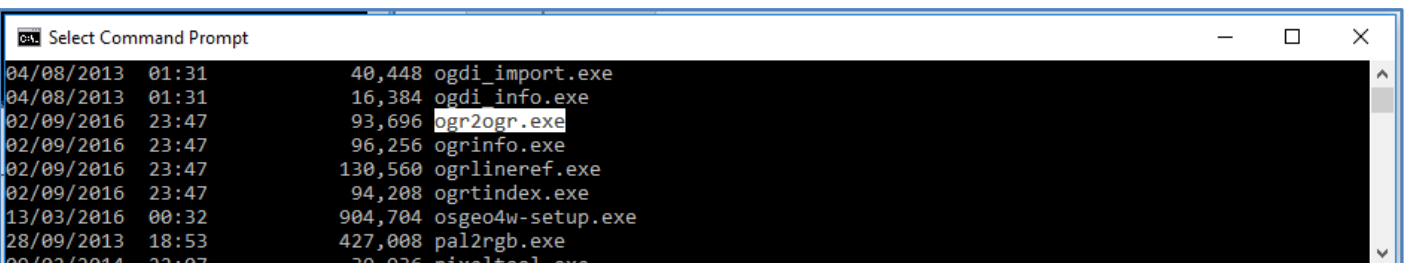
```

C:\OSGeo4W64\bin>dir *.exe
Volume in drive C has no label.
Volume Serial Number is 0017-5516

Directory of C:\OSGeo4W64\bin

09/02/2014  22:07          1,379,328  assistant.exe
10/08/2013  12:14          331,264  avcexport.exe
10/08/2013  12:14          360,960  avcimport.exe
28/09/2013  18:53          427,520  bmp2tiff.exe
04/09/2016  19:51           15,872  cs2cs.exe
01/08/2013  13:34          259,584  curl.exe
09/02/2014  22:08          828,928  designer.exe
05/08/2013  10:02           69,120  dllupdate.exe
04/08/2013  01:31           7,168  example1.exe
04/08/2013  01:31           8,704  example2.exe
28/09/2013  18:53          429,056  fax2ps.exe
    
```

One of the applications available will be Ogr2Ogr.



```

C:\> Select Command Prompt

04/08/2013  01:31           40,448  ogdi_import.exe
04/08/2013  01:31           16,384  ogdi_info.exe
02/09/2016  23:47           93,696  ogr2ogr.exe
02/09/2016  23:47           96,256  ogrinfo.exe
02/09/2016  23:47          130,560  ogrlineref.exe
02/09/2016  23:47           94,208  ogrtindex.exe
13/03/2016  00:32          904,704  osgeo4w-setup.exe
28/09/2013  18:53          427,008  pal2rgb.exe
09/02/2014  22:07           30,036  rivialtool.exe
    
```

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.

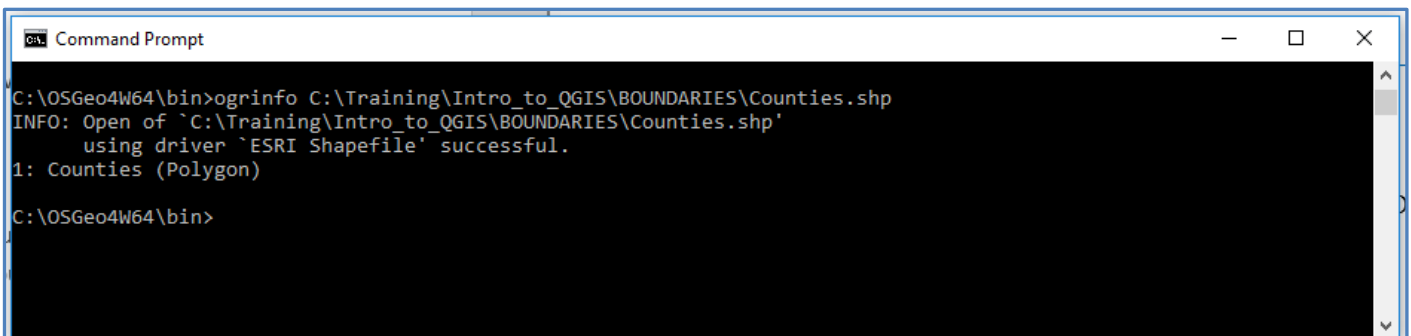
### 2.1 – Identify the geometry of an ESRI Shapefile

#### Command:

```
ogrinfo Location\ShapefileName.shp
```

#### Example:

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



```
Command Prompt
C:\OSGeo4W64\bin>ogrinfo C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
INFO: Open of `C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp'
      using driver `ESRI Shapefile' successful.
1: Counties (Polygon)
C:\OSGeo4W64\bin>
```

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

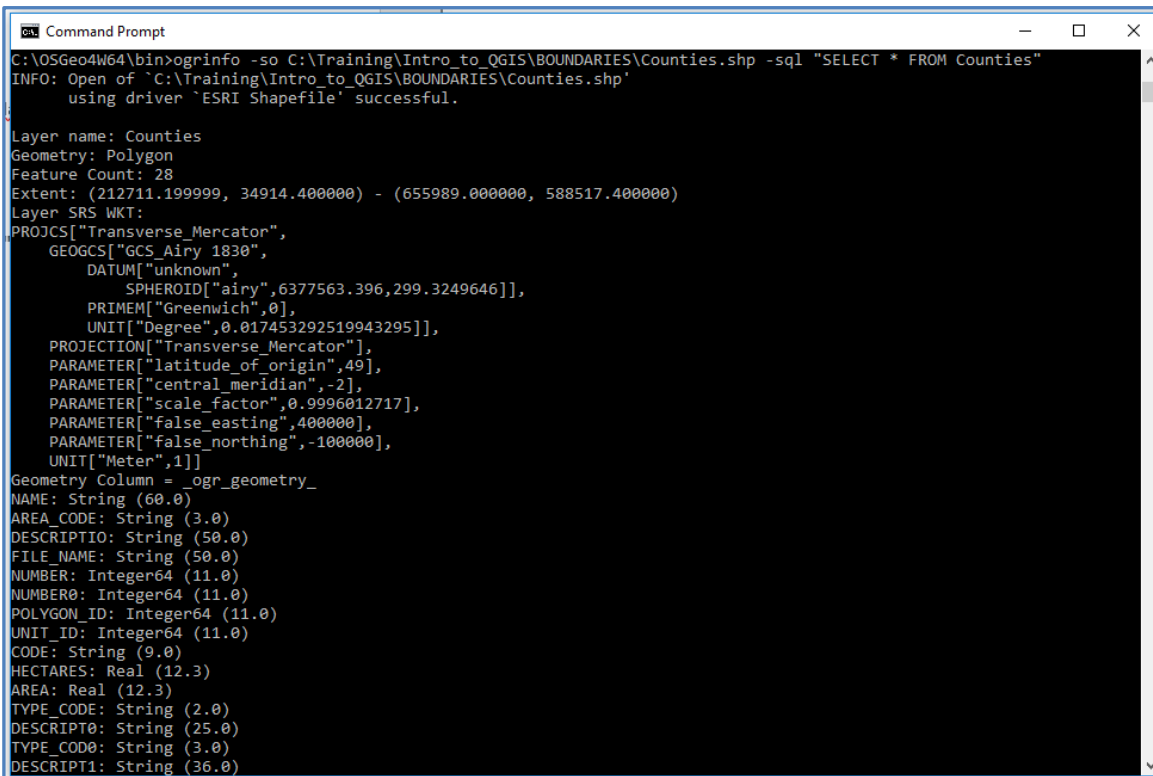
## 2.2 – Get full details of an ESRI Shapefile

### 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"
```



```

C:\OSGeo4W64\bin>ogrinfo -so C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -sql "SELECT * FROM Counties"
INFO: Open of `C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp'
      using driver `ESRI Shapefile' successful.

Layer name: Counties
Geometry: Polygon
Feature Count: 28
Extent: (212711.199999, 34914.400000) - (655989.000000, 588517.400000)
Layer SRS WKT:
PROJCS["Transverse_Mercator",
  GEOGCS["GCS_Airy_1830",
    DATUM["unknown",
      SPHEROID["airy",6377563.396,299.3249646]],
    PRIMEM["Greenwich",0],
    UNIT["Degree",0.017453292519943295]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",49],
  PARAMETER["central_meridian",-2],
  PARAMETER["scale_factor",0.9996012717],
  PARAMETER["false_easting",400000],
  PARAMETER["false_northing",-100000],
  UNIT["Meter",1]]
Geometry Column = _ogr_geometry_
NAME: String (60.0)
AREA_CODE: String (3.0)
DESCRIPTIO: String (50.0)
FILE_NAME: String (50.0)
NUMBER: Integer64 (11.0)
NUMBER0: Integer64 (11.0)
POLYGON_ID: Integer64 (11.0)
UNIT_ID: Integer64 (11.0)
CODE: String (9.0)
HECTARES: Real (12.3)
AREA: Real (12.3)
TYPE_CODE: String (2.0)
DESCRIPT0: String (25.0)
TYPE_COD0: String (3.0)
DESCRIPT1: String (36.0)
  
```

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

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

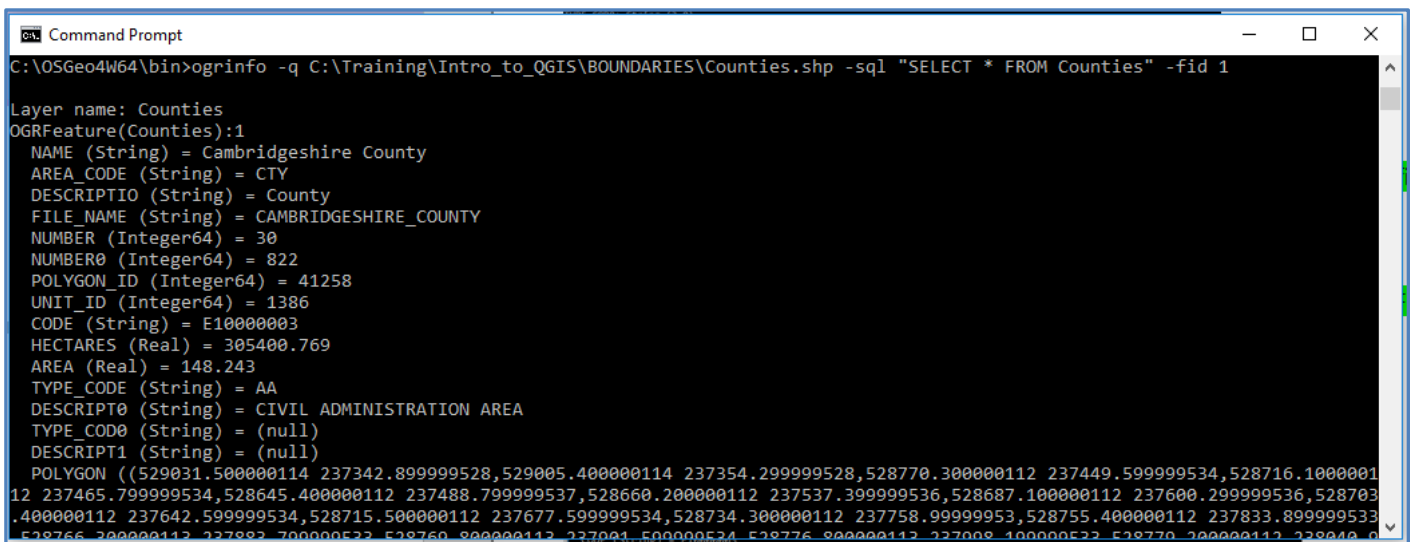
## 2.3 – Get full details for a Specific Record of an ESRI Shapefile

### 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
```

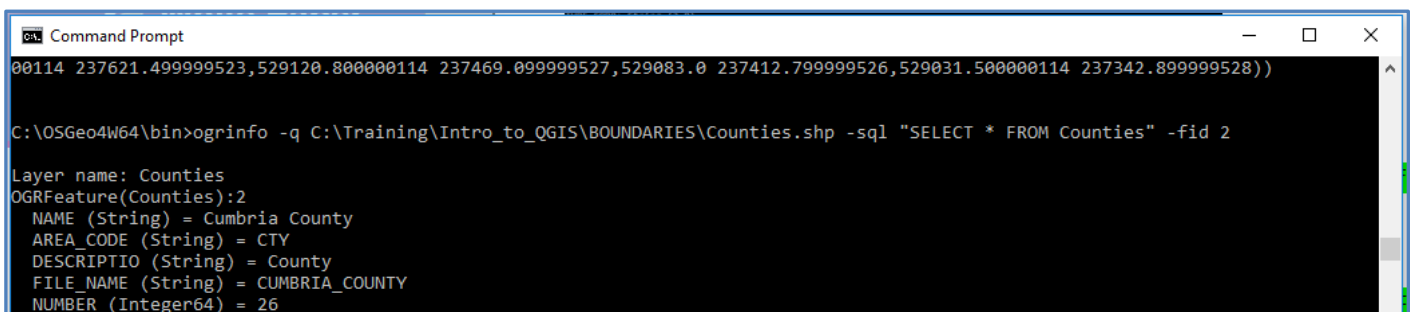


```

C:\OSGeo4W64\bin>ogrinfo -q C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -sql "SELECT * FROM Counties" -fid 1
Layer name: Counties
OGRFeature(Counties):1
  NAME (String) = Cambridgeshire County
  AREA_CODE (String) = CTY
  DESCRIPTIO (String) = County
  FILE_NAME (String) = CAMBRIDGESHIRE_COUNTY
  NUMBER (Integer64) = 30
  NUMBER0 (Integer64) = 822
  POLYGON_ID (Integer64) = 41258
  UNIT_ID (Integer64) = 1386
  CODE (String) = E10000003
  HECTARES (Real) = 305400.769
  AREA (Real) = 148.243
  TYPE_CODE (String) = AA
  DESCRIPT0 (String) = CIVIL ADMINISTRATION AREA
  TYPE_COD0 (String) = (null)
  DESCRIPT1 (String) = (null)
  POLYGON ((529031.500000114 237342.899999528,529005.400000114 237354.299999528,528770.300000112 237449.599999534,528716.1000001
12 237465.799999534,528645.400000112 237488.799999537,528660.200000112 237537.399999536,528687.100000112 237600.299999536,528703
.400000112 237642.599999534,528715.500000112 237677.599999534,528734.300000112 237758.99999953,528755.400000112 237833.899999533
528766.200000112 237882.700000522 528760.800000112 237001.500000524 528776.900000112 237008.100000522 528770.200000112 238040.0
  
```

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.



```

C:\OSGeo4W64\bin>ogrinfo -q C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -sql "SELECT * FROM Counties" -fid 2
Layer name: Counties
OGRFeature(Counties):2
  NAME (String) = Cumbria County
  AREA_CODE (String) = CTY
  DESCRIPTIO (String) = County
  FILE_NAME (String) = CUMBRIA_COUNTY
  NUMBER (Integer64) = 26
  
```

## 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.

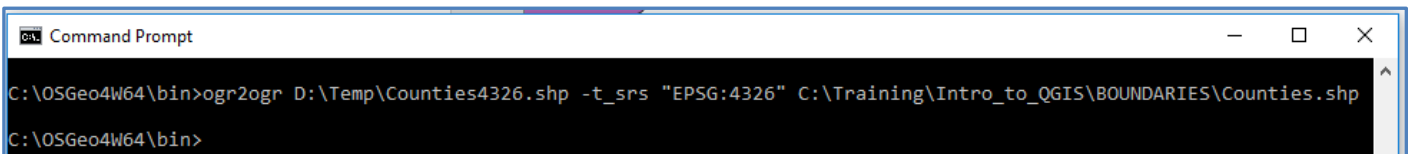
### 3.1 – Reproject an existing ESRI Shapefile

#### Command:

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

#### Example:

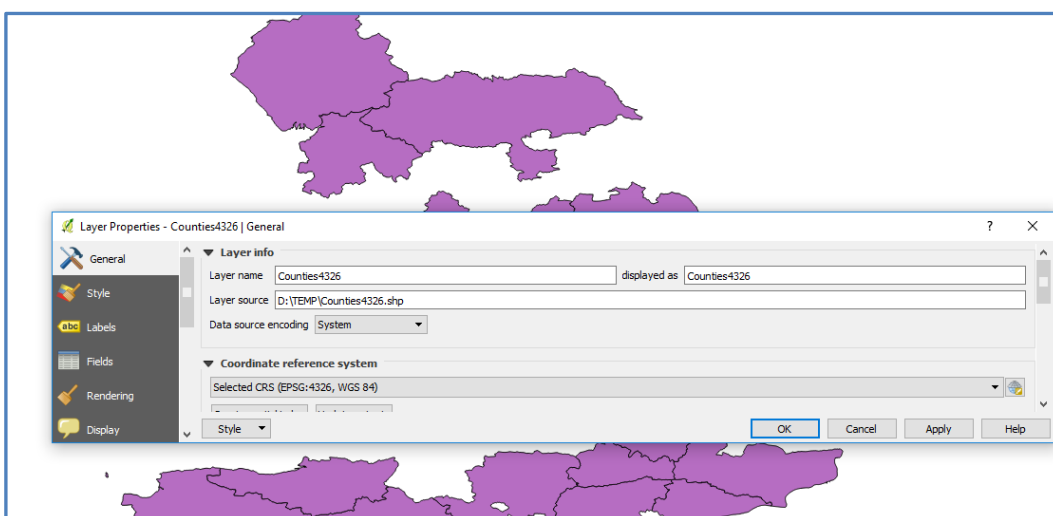
```
ogr2ogr D:\Temp\Counties4326.shp -t_srs "EPSG:4326"  
C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
```



```

C:\OSGeo4W64\bin>ogr2ogr D:\Temp\Counties4326.shp -t_srs "EPSG:4326" C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
C:\OSGeo4W64\bin>
  
```

The result will convert the original Counties Shapefile, projected as British National Grid (27700) to a new Shapefile called Counties4326 now projected as WGS 84 (4326).



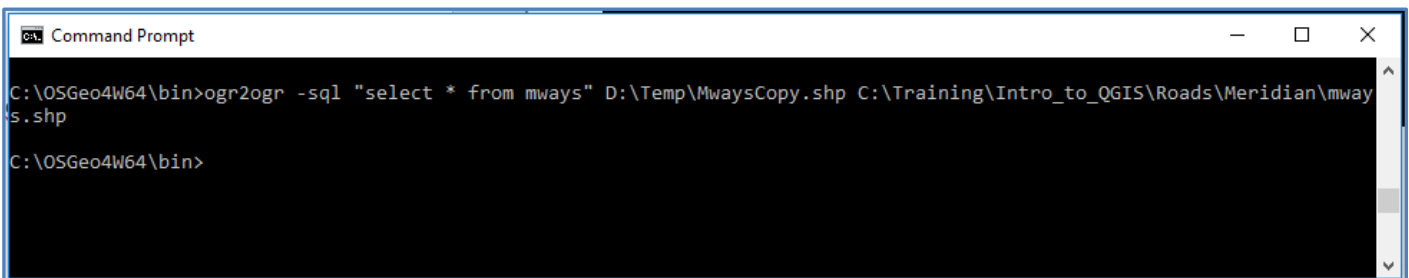
### 3.2 – Save a Copy of an ESRI Shapefile

#### 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
```



```

C:\OSGeo4W64\bin>ogr2ogr -sql "select * from mways" D:\Temp\MwaysCopy.shp C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp
C:\OSGeo4W64\bin>
  
```

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





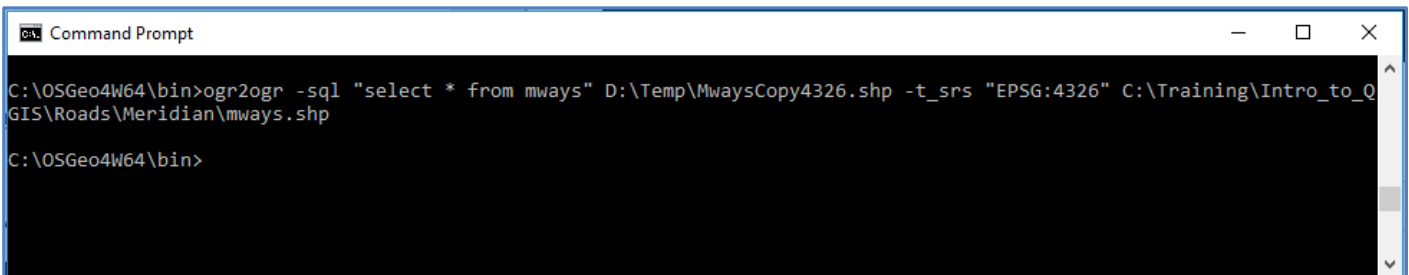
### 3.3 – Save a Copy of an ESRI Shapefile with a new Projection

#### 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
```



```

C:\OSGeo4W64\bin>ogr2ogr -sql "select * from mways" D:\Temp\MwaysCopy4326.shp -t_srs "EPSG:4326" C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp
C:\OSGeo4W64\bin>
  
```

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



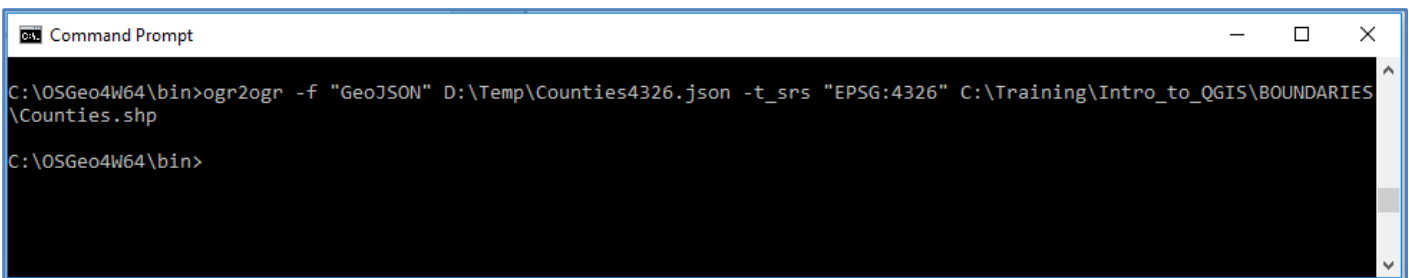
### 3.4 – Translate an ESRI Shapefile to GeoJSON

#### Command:

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

#### Example:

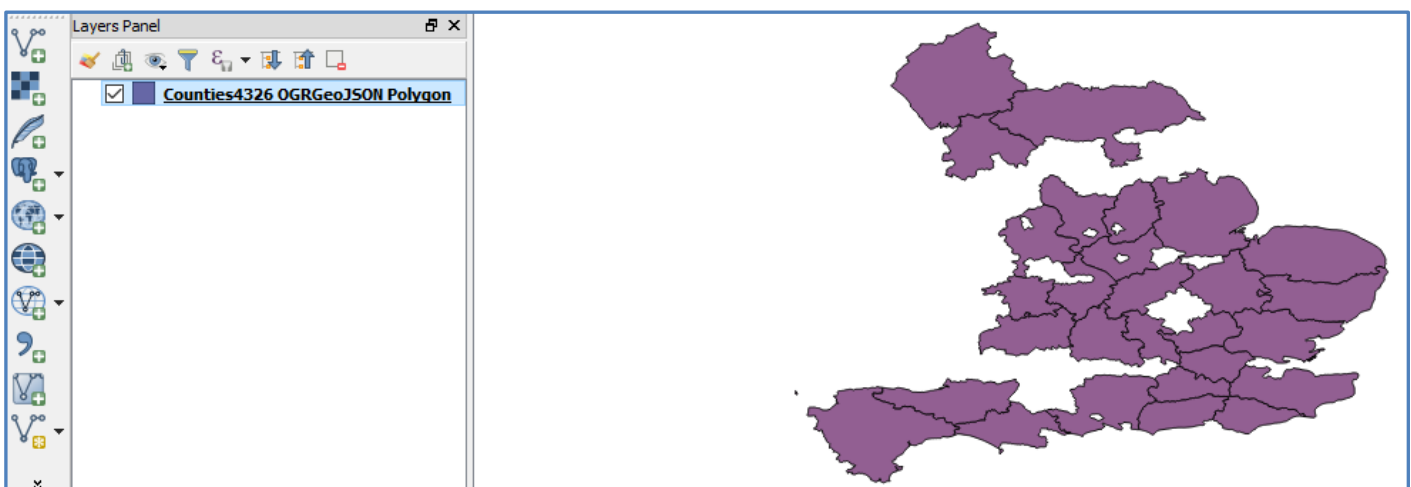
```
ogr2ogr -f "GeoJSON" D:\Temp\Counties4326.json -t_srs "EPSG:4326"  
C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
```



```

C:\OSGeo4W64\bin>ogr2ogr -f "GeoJSON" D:\Temp\Counties4326.json -t_srs "EPSG:4326" C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
C:\OSGeo4W64\bin>
  
```

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



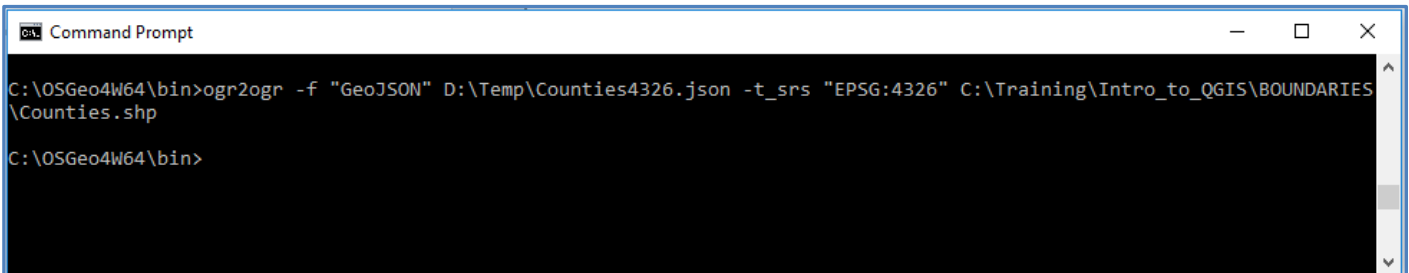
### 3.5 – Translate an ESRI Shapefile to KML and re-project to 4326

#### Command:

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

#### Example:

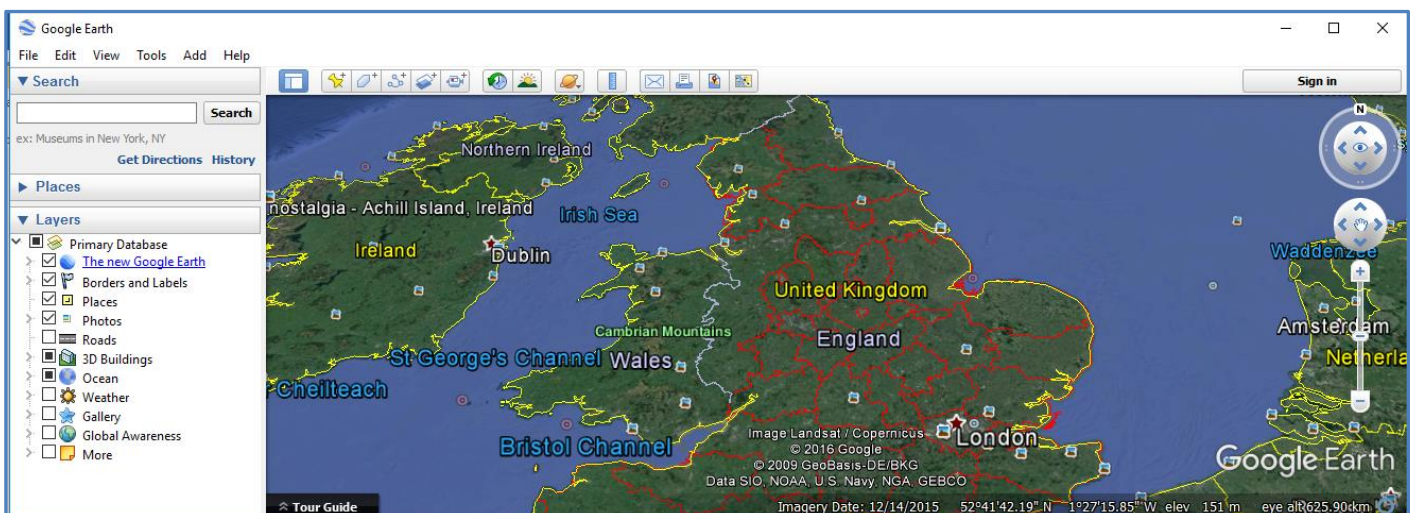
```
ogr2ogr -f "KML" D:\Temp\KMLCounties.kml C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp -dsc DescriptionField='DESCRIPTION'
```



```

C:\OSGeo4W64\bin>ogr2ogr -f "GeoJSON" D:\Temp\Counties4326.json -t_srs "EPSG:4326" C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
C:\OSGeo4W64\bin>
  
```

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.**

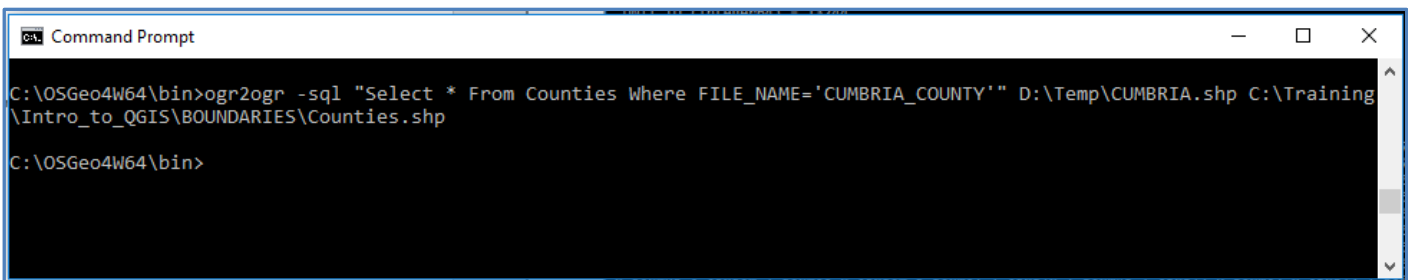
### 3.6 – Select Records and Create a New Shapefile

#### 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
```



```

Command Prompt
C:\OSGeo4W64\bin>ogr2ogr -sql "Select * From Counties Where FILE_NAME='CUMBRIA_COUNTY'" D:\Temp\CUMBRIA.shp C:\Training\Intro_to_QGIS\BOUNDARIES\Counties.shp
C:\OSGeo4W64\bin>
  
```

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



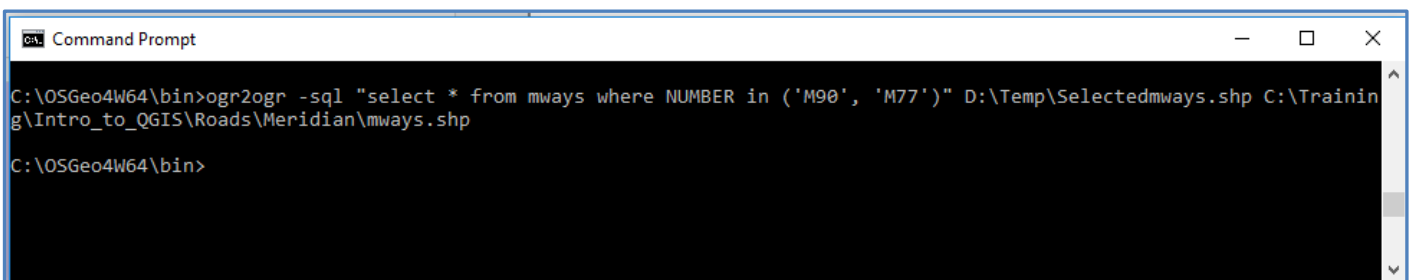
### 3.7 – Select Multiple Records and Create a New Shapefile

#### Command:

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

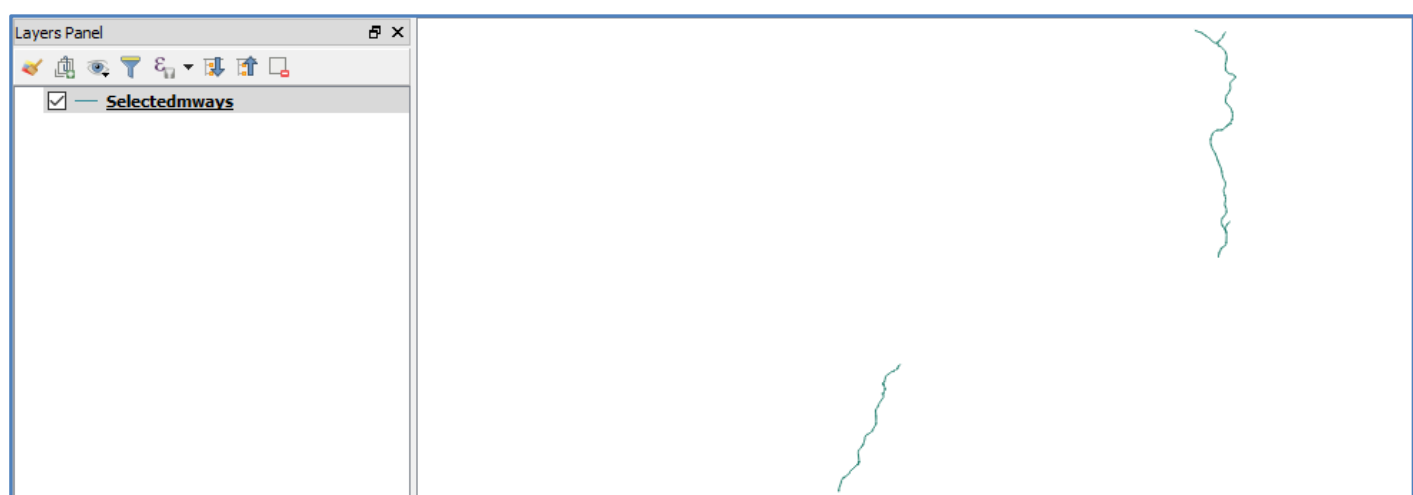
#### Example:

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



```
Command Prompt
C:\OSGeo4W64\bin>ogr2ogr -sql "select * from mways where NUMBER in ('M90', 'M77')" D:\Temp\Selectedmways.shp C:\Trainin
g\Intro_to_QGIS\Roads\Meridian\mways.shp
C:\OSGeo4W64\bin>
```

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



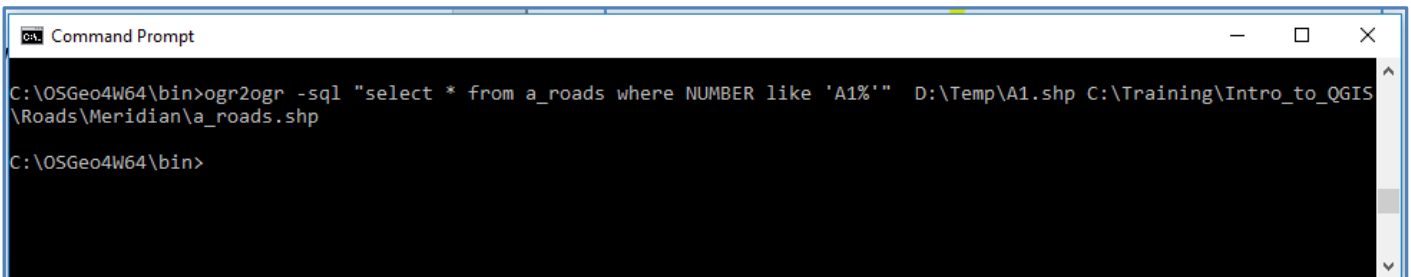
### 3.8 – Select Multiple Records using a LIKE command and Create a New Shapefile

#### Command:

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

#### Example:

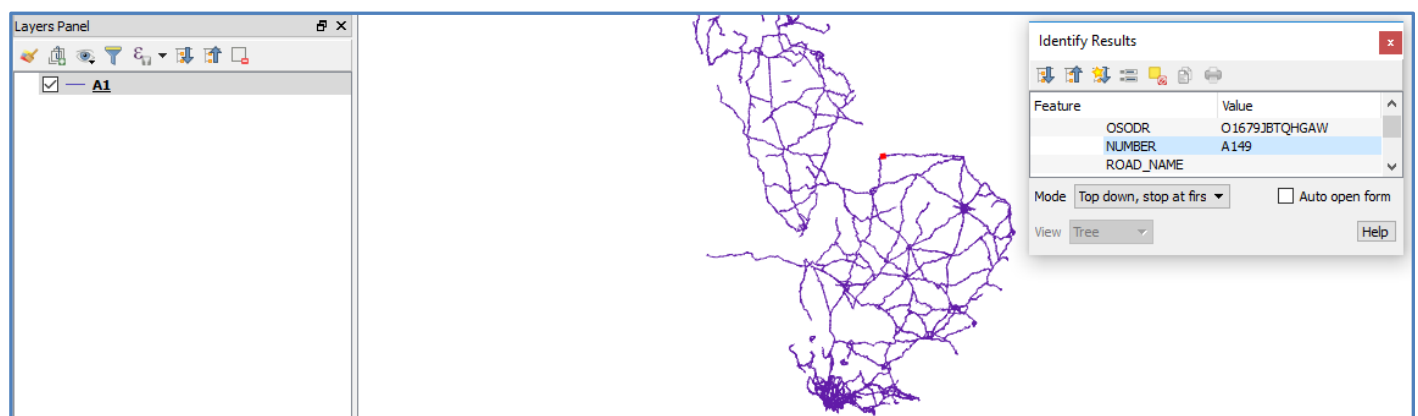
```
ogr2ogr -sql "select * from a_roads where NUMBER like 'A1%'" D:\Temp\A1.shp
C:\Training\Intro_to_QGIS\Roads\Meridian\a_roads.shp
```



```

Command Prompt
C:\OSGeo4W64\bin>ogr2ogr -sql "select * from a_roads where NUMBER like 'A1%'" D:\Temp\A1.shp C:\Training\Intro_to_QGIS\Roads\Meridian\a_roads.shp
C:\OSGeo4W64\bin>
  
```

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...'



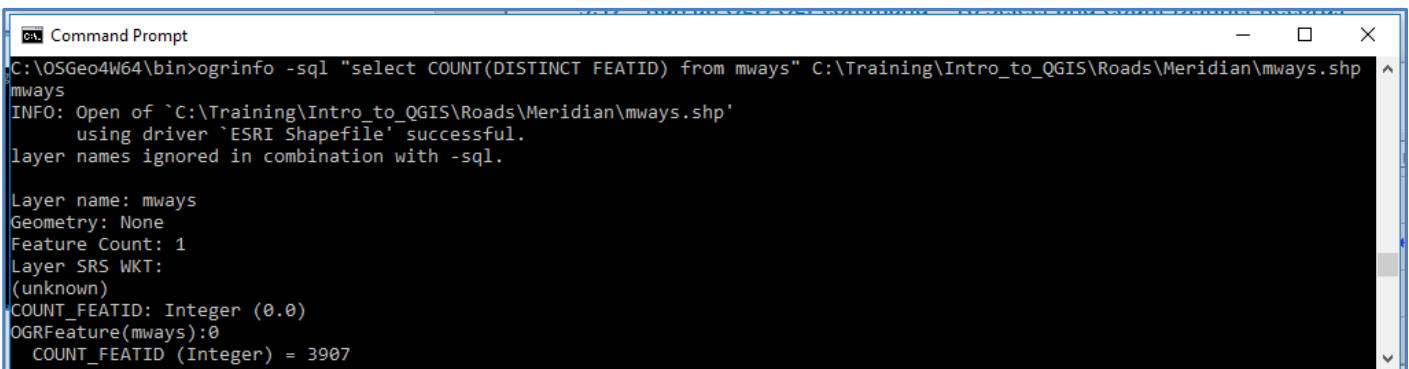
### 3.9 – Count Records in an ESRI Shapefile

#### 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
```

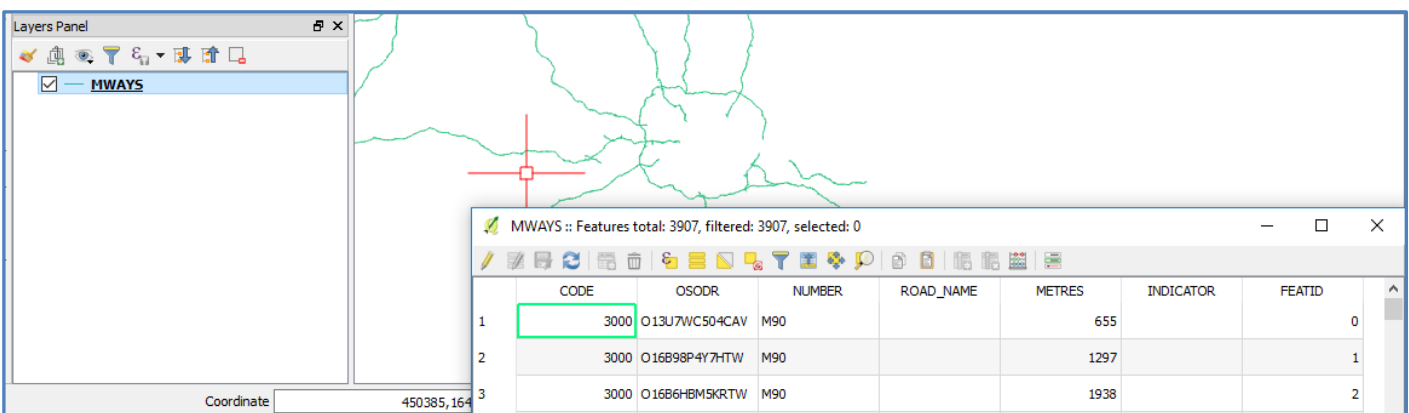


```

C:\OSGeo4W64\bin>ogrinfo -sql "select COUNT(DISTINCT FEATID) from mways" C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp
mways
INFO: Open of `C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp'
      using driver `ESRI Shapefile' successful.
layer names ignored in combination with -sql.

Layer name: mways
Geometry: None
Feature Count: 1
Layer SRS WKT:
(unknown)
COUNT_FEATID: Integer (0.0)
OGRFeature(mways):0
  COUNT_FEATID (Integer) = 3907
  
```

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 DISTINCT records using the FEATURE ID, so the result (3907) should be equal to a Count of all records.



	CODE	OSODR	NUMBER	ROAD_NAME	METRES	INDICATOR	FEATID
1	3000	O13U7WC504CAV	M90		655		0
2	3000	O16B98P4Y7HTW	M90		1297		1
3	3000	O16B6HBM5KRTW	M90		1938		2

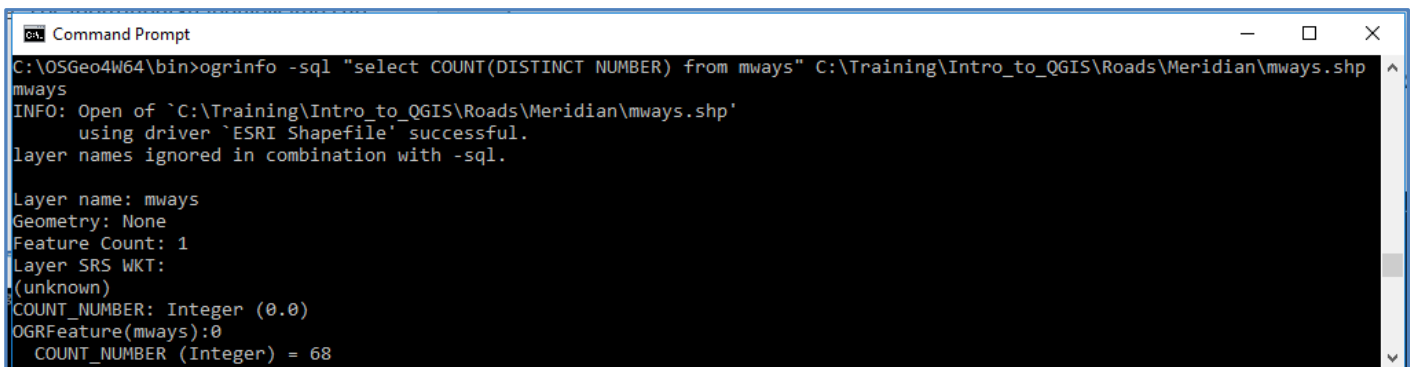
### 3.10 – Count DISTINCT Records in an ESRI Shapefile

#### 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
```



```
Command Prompt  
C:\OSGeo4W64\bin>ogrinfo -sql "select COUNT(DISTINCT NUMBER) from mways" C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp  
mways  
INFO: Open of `C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp`  
using driver `ESRI Shapefile` successful.  
layer names ignored in combination with -sql.  
  
Layer name: mways  
Geometry: None  
Feature Count: 1  
Layer SRS WKT:  
(unknown)  
COUNT_NUMBER: Integer (0.0)  
OGRFeature(mways):0  
COUNT_NUMBER (Integer) = 68
```

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 DISTINCT 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.



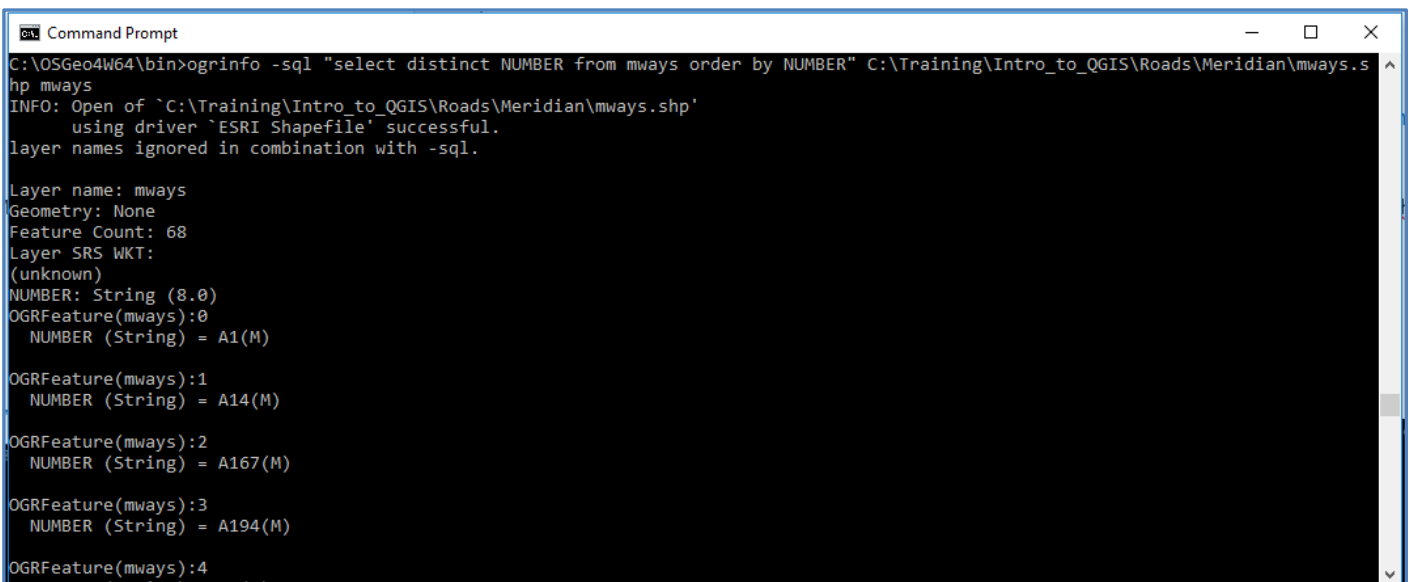
### 3.11 – List and Order DISTINCT Records in an ESRI Shapefile for a Chosen Field

#### 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
```



```

C:\OSGeo4W64\bin>ogrinfo -sql "select distinct NUMBER from mways order by NUMBER" C:\Training\Intro_to_QGIS\Roads\Meridian\mways.s
hp mways
INFO: Open of `C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp'
      using driver `ESRI Shapefile' successful.
layer names ignored in combination with -sql.

Layer name: mways
Geometry: None
Feature Count: 68
Layer SRS WKT:
(unknown)
NUMBER: String (8.0)
OGRFeature(mways):0
  NUMBER (String) = A1(M)

OGRFeature(mways):1
  NUMBER (String) = A14(M)

OGRFeature(mways):2
  NUMBER (String) = A167(M)

OGRFeature(mways):3
  NUMBER (String) = A194(M)

OGRFeature(mways):4
  NUMBER (String) = A194(M)
  
```

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

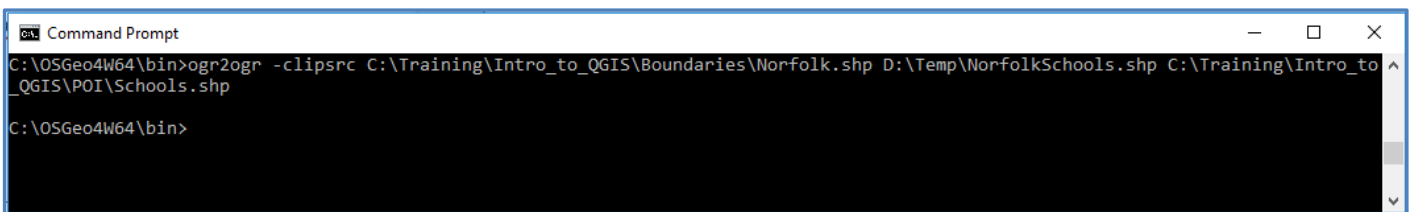
### 3.12 – Clip Spatial Data (Points in Polygon)

#### Command:

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

#### Example:

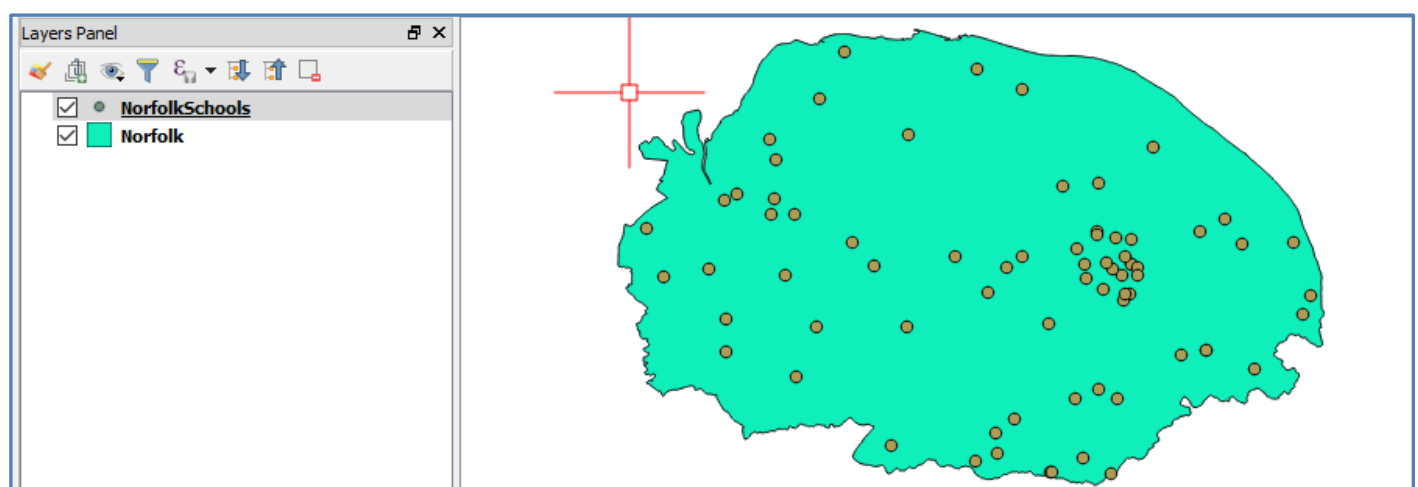
```
ogr2ogr -clipsrc C:\Training\Intro_to_QGIS\Boundaries\Norfolk.shp D:\Temp\NorfolkSchools.shp
C:\Training\Intro_to_QGIS\POI\Schools.shp
```



```

Command Prompt
C:\OSGeo4W64\bin>ogr2ogr -clipsrc C:\Training\Intro_to_QGIS\Boundaries\Norfolk.shp D:\Temp\NorfolkSchools.shp C:\Training\Intro_to_QGIS\POI\Schools.shp
C:\OSGeo4W64\bin>
  
```

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



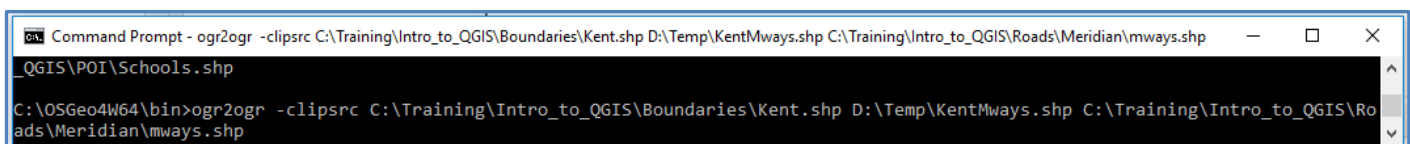
### 3.12 – Clip Spatial Data (Lines in Polygon)

#### Command:

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

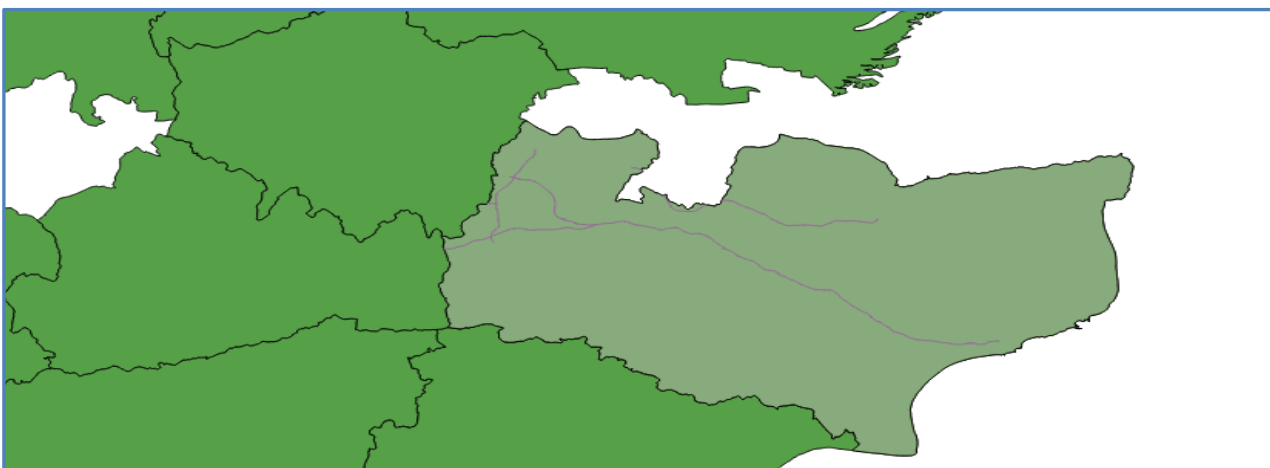
#### Example:

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



```
Command Prompt - ogr2ogr -clipsrc C:\Training\Intro_to_QGIS\Boundaries\Kent.shp D:\Temp\KentMways.shp C:\Training\Intro_to_QGIS\Roads\Meridian\mways.shp - X  
_QGIS\POI\Schools.shp  
C:\OSGeo4W64\bin>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

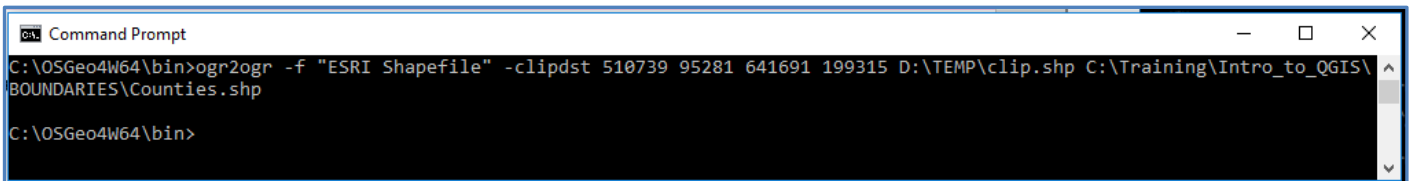
#### Command:

```
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



```

Command Prompt
C:\OSGeo4W64\bin>ogr2ogr -f "ESRI Shapefile" -clipdst 510739 95281 641691 199315 D:\TEMP\clip.shp C:\Training\Intro_to_QGIS\
BOUNDARIES\Counties.shp
C:\OSGeo4W64\bin>
    
```

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

