1.

1:250 000 Scale Colour Raster™ Free OS OpenData

Get the regional view of towns and villages, roads and places of interest.

Coverage: All of Great Britain
Data structure: Raster
Supply format: TIFF-LZW
Version Date: 2022-06

Documentation

Product information

Download 1:250 000 Scale Colour Raster™

All OS OpenData can be freely downloaded under the Open Government Licence.

1 Specify an area — All of Great Britain

2 Data format — TIFF-LZW
Size: 135.12 MB

3 Download — 1:250 000 Scale Colour Raster™ - TIFF-LZW GB (135.12 MB)

2.

raster2pgsql.exe -x 3763 -N -32767 -t 100x100 -I -C -M -d D:\SQL_code\ras250_gb\data*.tif uk_250k | psql -d lab7 -h localhost -U postgres -p 5432

uk_250k 1 × Output

SELECT rid, rast FROM public.uk_250k limit 10 Enter a SQL expression to filter results (use Ctrl+Space)

Grid Text

	rid	rast
1	1	0100000100000000000000039400000000000039C000000000006A1841000000020D633410000000000
2	2	0100000100000000000000039400000000000039C000000000109118410000000020D633410000000000
3	3	0100000100000000000000039400000000000039C000000000020B818410000000020D633410000000000
4	4	0100000100000000000000039400000000000039C000000000030DF18410000000020D633410000000000
5	5	0100000100000000000000039400000000000039C000000000400619410000000020D633410000000000
6	6	0100000100000000000000039400000000000039C000000000502D19410000000020D633410000000000
7	7	0100000100000000000000039400000000000039C000000000605419410000000020D633410000000000

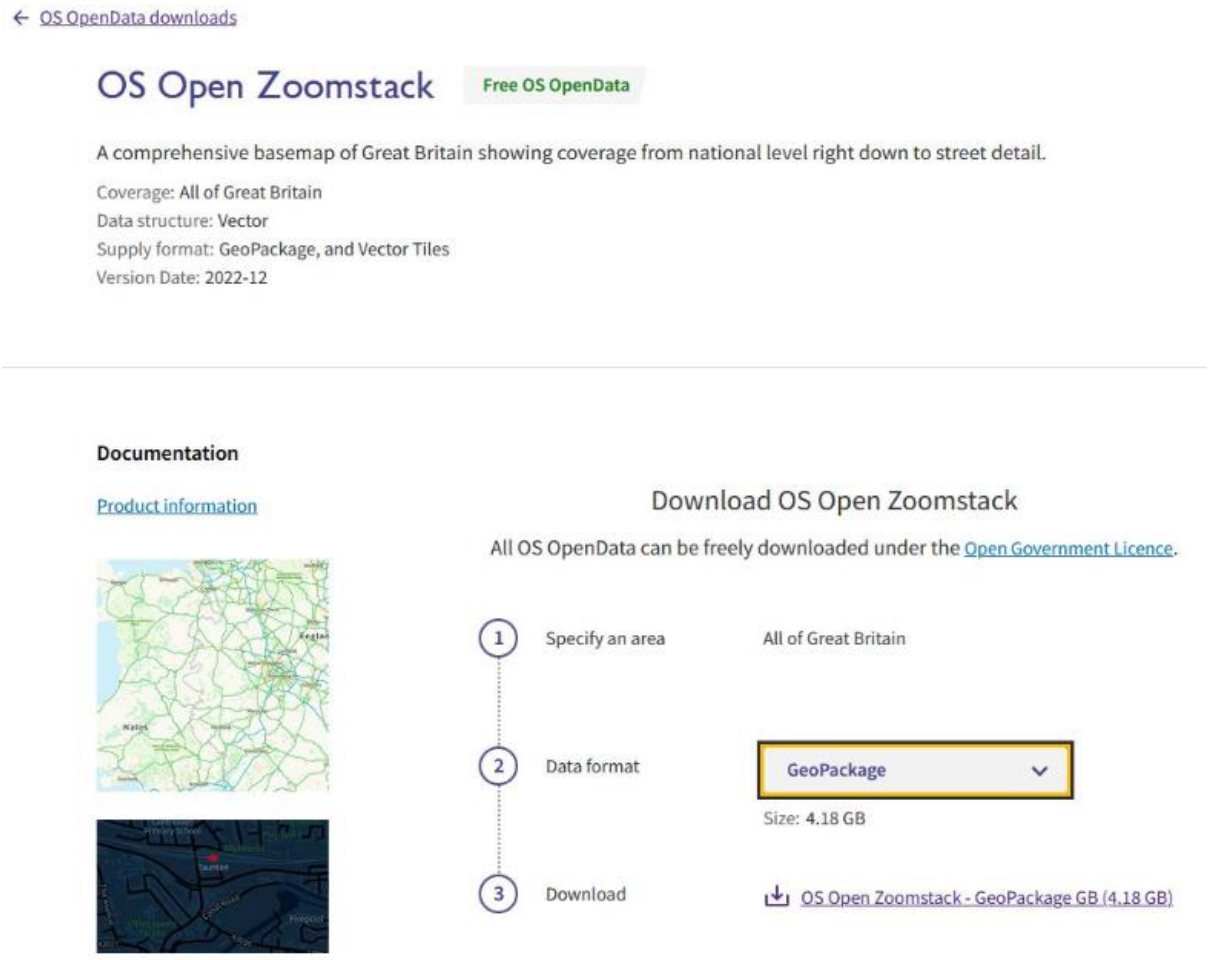
3.

```
CREATE INDEX idx_rast_gist ON uk_250k USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('public'::name,'uk_250k'::name,'rast'::name);

SELECT st_union(rast) FROM uk_250k
```

SQL Error [53200]: ERROR: out of memory
Detail: Failed on request of size 20000 in memory context "ExprContext".

Error position:

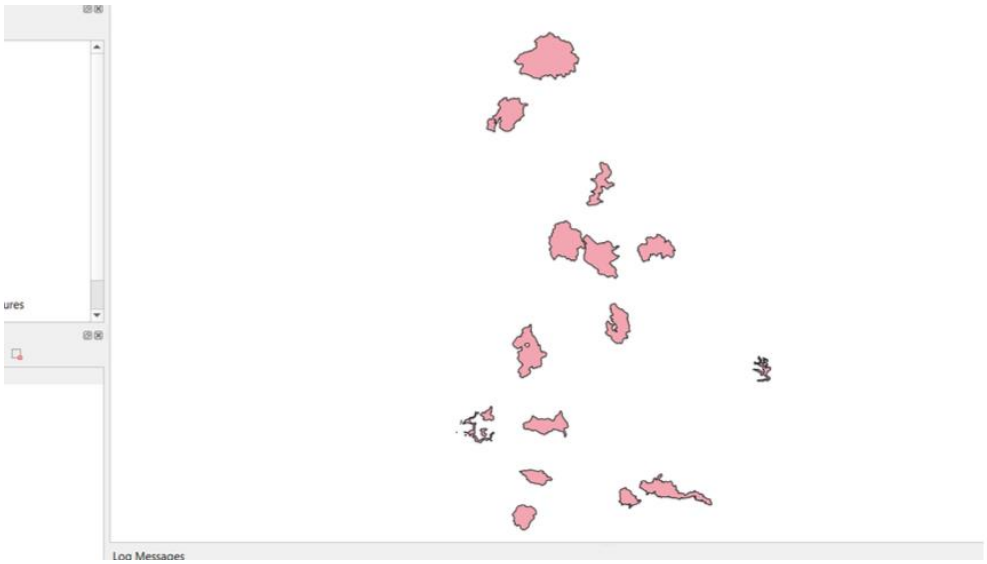
4.



5.

ogr2ogr.exe D:\SQL_Code D:\SQL_code\OS_Open_Zoomstack.gpkg

shp2pgsql -s 27700 D:\SQL_code\gb\national_parks.shp national_parks | psql -U postgres -h localhost -p 5432 -d lab7



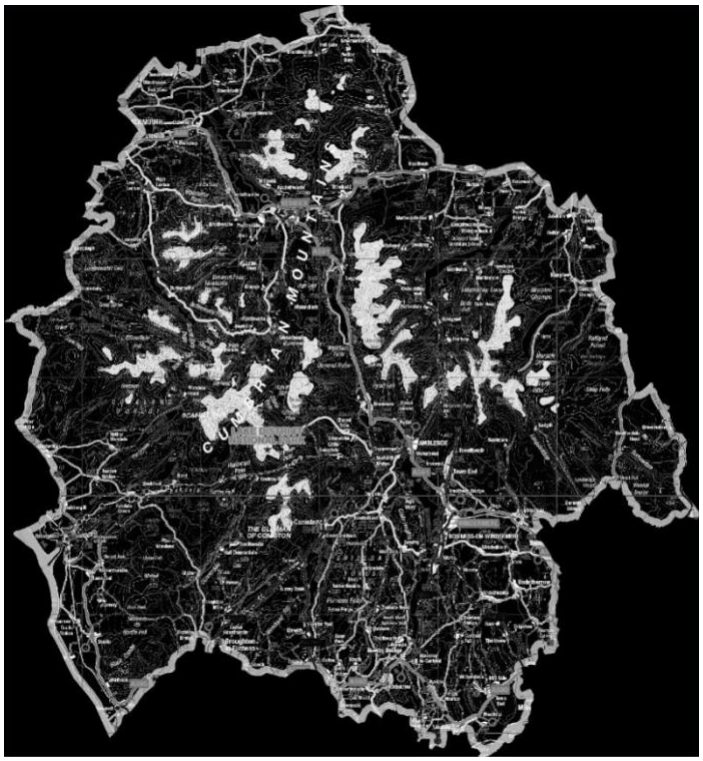
6.

```
SELECT UpdateGeometrySRID('national_parks','geom',3763);
```

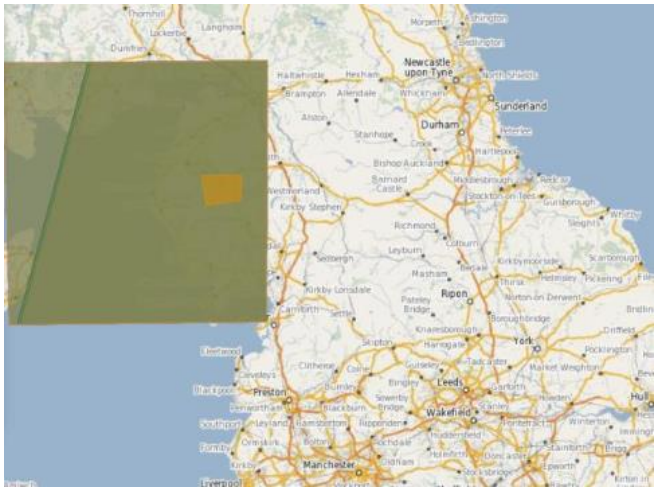
```
CREATE TABLE uk_lake_district AS
SELECT uk.rid,ST_Clip(a.rast, b.geom,true) AS rast FROM uk_250k AS uk,
national_parks AS np WHERE np.gid = 1 AND ST_Intersects(np.geom,uk.rast);
```

7.

```
CREATE TABLE tmp_out AS
SELECT lo_from_bytea(0, ST_AsGDALRaster(ST_Union(rast),
'GTiff',ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=2', 'PZLEVEL=9'])) AS loid
FROM uk_lake_district;

SELECT lo_export(loid, 'D:\SQL_code\gb\raster.tiff') FROM tmp_out;

SELECT lo_unlink(loid) FROM tmp_out;
```

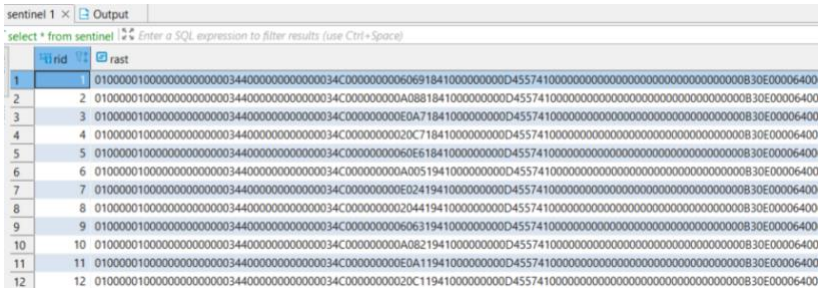


8.



9.

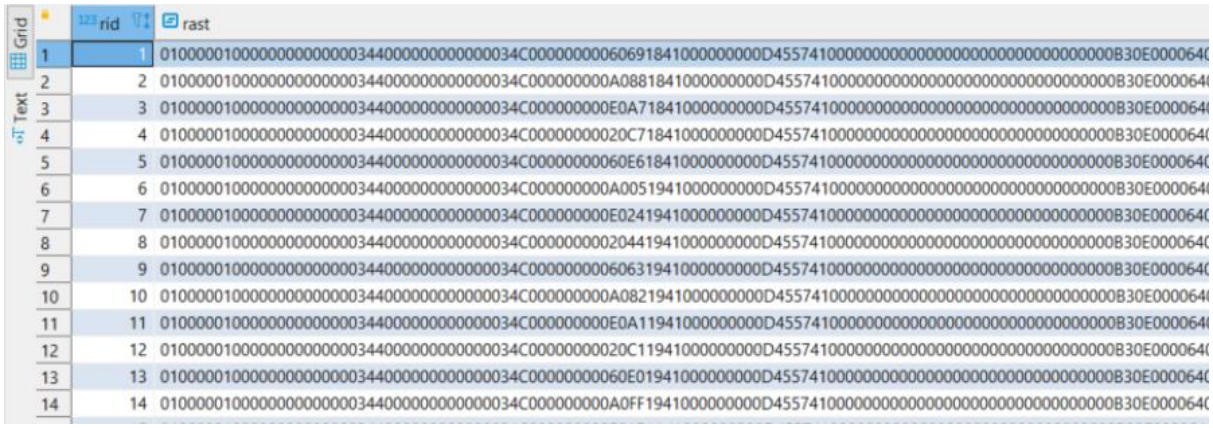
raster2pgsql.exe -s 3763 -N -32767 -t 100x100 -I -C -M -d D:\SQL_code\sentinel2\S2B_MSIL1C_20221213T113449_N0509_R080_T30UVF_20221213T121423.SAFE\GRANULE\L1C_T30UVF_A030136_20221213T113448\IMG_DATA\T30UVF_20221213T113449_B07.jp2 sentinel| psql -d lab7 -h localhost -U postgres -p 5432



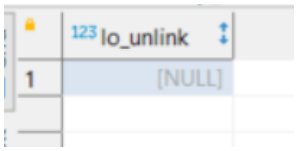

```
CREATE INDEX idx_rast_gist ON sentinel USING gist (ST_ConvexHull(rast));

SELECT AddRasterConstraints('public'::name,'sentinel'::name,'rast'::name);
```

10.

```
CREATE OR REPLACE FUNCTION ndvi(value double precision [] [] [],
pos INTEGER [][],VARIADIC userargs text [])
RETURNS DOUBLE PRECISION AS
$$
BEGIN
RETURN (value [2][1][1] - value [1][1][1])/(value [2][1][1]+value [1][1][1]);
END;
$$
LANGUAGE 'plpgsql' IMMUTABLE COST 1000;

CREATE TABLE ndvi AS WITH n AS (SELECT *FROM sentinel)

SELECT n.rid,ST_MapAlgebra(n.rast, ARRAY[1,4],
'ndvi(double precision[], integer[],text[])'::regprocedure,'32BF'::text ) AS rast
FROM n;
```




```
CREATE TABLE uk_lake_district_sentinel AS
SELECT a.rid,ST_Clip(a.rast, b.geom,TRUE) AS rast FROM ndvi AS a, np AS b WHERE
b.gid = 1 AND ST_Intersects(b.geom,a.rast);
```



11.

```
CREATE TABLE tmp_out AS
SELECT lo_from_bytea(0,ST_AsGDALRaster(ST_Union(rast), 'GTiff',
ARRAY['COMPRESS=DEFLATE', 'PREDICTOR=2', 'PZLEVEL=9'])) AS l
FROM uk_lake_district_sentinel;

SELECT lo_export(l, 'D:\SQL_code\sentinel-2\raster.tiff') FROM tmp_out;

SELECT lo_unlink(l) FROM tmp_out;
```