## Create tiles of a GeoTiff map using gdal2tiles

GDAL2Tiles is distributed together with [GDAL library](http://www.gdal.org/).

One of the problems with high-resolution raster data, though, is that it takes a lot of memory to display it. And if you're pushing it out over the net, you have bandwidth concerns as well. To address that problem, you have to create tiles.

[GDAL2Tiles](http://www.klokan.cz/projects/gdal2tiles/) is a command line tool that allows easy publishing of raster maps on the Internet. The raster image is converted into a directory structure of small tiles which you can copy to your webserver.

Simple web pages with viewers based on Google Maps and OpenLayers are generated as well - so anybody can comfortably explore your maps on-line and you do not need to install or configure any special [software](http://www.gdal.org/gdal2tiles.html) and the map displays very fast in the web browser. You only need to upload the generated directory onto a web server.

GDAL2Tiles also creates the necessary metadata for Google Earth (KML SuperOverlay), in case the supplied map uses EPSG:4326 projection.

You can use any image, but you need to know the boundaries of the image, the latitude and longitude of each of the corners of the image.

Once you've installed the GDAL libraries and selected the image, the first thing you have to do is to get some information about the image so that you can georeference it. Specifically, you need the pixel and line positions of each corner of the image.

GDAL provides a handy utility, gdalinfo, for capturing this information. At the command line, simply type

🡪 gdalinfo name\_of\_your\_file

You will get something like this:

Driver: GTiff/GeoTIFF

Files: mundo\_creado.tif

Size is 16200, 8100

Coordinate System is `'

GCP Projection =

GEOGCS["WGS 84",

DATUM["WGS\_1984",

SPHEROID["WGS 84",6378137,298.257223563,

AUTHORITY["EPSG","7030"]],

AUTHORITY["EPSG","6326"]],

PRIMEM["Greenwich",0],

UNIT["degree",0.0174532925199433],

AUTHORITY["EPSG","4326"]]

GCP[ 0]: Id=1, Info= (0,0) -> (-180,90,0)

GCP[ 1]: Id=2, Info= (16200,0) -> (180,90,0)

GCP[ 2]: Id=3, Info= (16200,8100) -> (180,-90,0)

Metadata:

AREA\_OR\_POINT=Area

EXIF\_ColorSpace=1

EXIF\_DateTime=2007:06:27 19:49:19

EXIF\_Orientation=1

EXIF\_PixelXDimension=16200

EXIF\_PixelYDimension=8100

EXIF\_ResolutionUnit=2

EXIF\_Software=Adobe Photoshop CS2 Macintosh

EXIF\_XResolution=(72)

EXIF\_YResolution=(72)

Image Structure Metadata:

INTERLEAVE=PIXEL

Corner Coordinates:

Upper Left ( 0.0, 0.0)

Lower Left ( 0.0, 8100.0)

Upper Right (16200.0, 0.0)

Lower Right (16200.0, 8100.0)

Center ( 8100.0, 4050.0)

Band 1 Block=16200x1 Type=Byte, ColorInterp=Red

Band 2 Block=16200x1 Type=Byte, ColorInterp=Green

Band 3 Block=16200x1 Type=Byte, ColorInterp=BlueDriver: GTiff/GeoTIFF

The important information for this case are the Upper Left, Lower Left, Upper Right, Lower Right lines. These tell you the pixel and line values of each corner. In this case: the Upper Left is at 0,0; the Lower Left is at 0,8100; the Upper Right is at 16200,0; and the Lower Right is at 16200,8100

The next step will be georeferencing the image. Georeferencing in this case means to create metadata describing the geographic position of each of the corners of the image. Using the information gained before and gdal\_translate, you can assign georeference information to the file.

🡪 gdal\_translate -a\_srs EPSG:4326 -gcp 0 0 -180 90 -gcp 16200 0 180 90 -gcp 16200 8100 180 -90 name\_of\_your\_file.jpg new\_name.tif

-a\_srs assigns a spatial reference system to the file. That tells any application consuming it what coordinate system is being used. In this case, it is using EPSG:4326, which is the same as WGS84.

-gcp, or ground control point, assigns coordinates to positions in the file. For -gcp, define the gcp by setting the pixel and then line number, and then the longitude and latitude. Each of those is separated by a space.

The last two parameters are the origin file and the target file.

There are more options you can use, just type gdal\_translate and you can see them.

The original image wasn't created for a round globe, it was created to appear to lie flat. In GIS terms, it is [projected](http://en.wikipedia.org/wiki/Map_projection" \t "_blank), which means that it is a two-dimensional representation of a three-dimensional object. Projection requires distorting the image so that it appears how you would expect a flat image of the Earth to look.

In order to get it to look right, you have to warp the image it to fit the globe. Fortunately GDAL provides a great tool for that too. Simply type

🡪 gdalwarp -t\_srs EPSG:4326 new\_name.tif new\_name \_4326.tif

-t\_srs indicates the spatial reference system we are going to define. This will create a new file, which provides metadata about the warping procedure.

There are more options you can use, just type gdalwarp and you can see them.

Finally, you just need to use GDAL2Tiles to generate the map tiles:

🡪 gdal2tiles new\_name \_4326.tif name\_of\_folder

There are a lot of options you can use here:

gdal2tiles [-p profile] [-r resampling] [-s srs] [-z zoom]

[-e] [-a nodata] [-v] [-h] [-k] [-n] [-u url]

[-w webviewer] [-t title] [-c copyright]

[-g googlekey] [-b bingkey] input\_file [output\_dir]

-s SRS, --s\_srs=SRS: The spatial reference system used for the source input data.

-z ZOOM, --zoom=ZOOM: Zoom levels to render (format:'2-5' or '10').

-w WEBVIEWER, --webviewer=WEBVIEWER: Web viewer to generate (all, google, openlayers, none) - default 'all'.

-t TITLE, --title=TITLE: Title of the map.

-c COPYRIGHT, --copyright=COPYRIGHT: Copyright for the map.

-k, -n, -u are options only for KML (Google Earth).