# Instructions For gdal2tiles_parallel.py

Steven D. Lander, Reinventing Geospatial

January 26, 2015

## 1 Purpose

This document covers installation of dependencies, in-depth explanations of command-line arguments, and usage examples for gdal2tiles_parallel.py.

The file gdal2tiles_parallel.py is a Python script that converts GDAL-supported raster imagery files into a folder of tiles in TMS format (z/x/y). This version improves upon the standard gdal2tiles.py file included with GDAL by adding multiprocessing improvements to obtain even better tile generation performance.

## 2 Installing Dependencies

Gdal2tiles_parallel.py was written for Python 2.7.x and has not been tested on any other version. It will run on either 32 or 64 bit systems with no issues.

The script relies on Python 2.7.x, GDAL core, and the GDAL Python bindings. Some imagery formats such as MrSID will require additional driver installers to work correctly.

### 2.1 Windows

In order to run gdal2tiles_parallel.py on a Windows environment, install the following packages. For convenience, they have been included in all PythonGeopackage releases and are viewable in source under Dependencies.

- The latest version of Python 2.7 for either x86 (32 bit) or x86_64 (64 bit): `https://www.python.org/downloads/windows/`.
- The lastest stable release of the Geospatial Data Abstraction Library (GDAL) in either x86 (32 bit) or x86_64 (64 bit): `http://www.gisinternals.com/sdk`

### 2.2 Linux

In order to run gdal2tiles_parallel.py on a Linux environment, instructions will differ slightly by distribution. Most newer Linux distributions have Python 2.7 as either an option or the default python version in their package repository, but others such as CentOS only have older versions. In that case, the user will need to find out how to get Python 2.7 from a reputable source or compile it themselves. The following packages are needed:

- The latest version of Python 2.7 for your system. For Debian-based distributions such as Ubuntu and RedHat type the following into a command line:

  ```
  sudo apt-get install python2.7 python2.7-dev python-pip base-devel
  ```

  This will install the Python 2.7 environment plus PIP, a Python module manager.

- The latest GDAL binaries and python bindings for your system. For Debian-based distributions such as Ubuntu and RedHat, type the following into a command line:

  ```
  sudo apt-get install gdal-bin python-gdal
  ```

# 3 Usage

## 3.1 Command Line Arguments

Gdal2tiles_parallel.py supports additional functionality via command line arguments provided to the script at the time it is executed. Following is a outline of the important flags:

| | |
|---|---|
| -h, –help | Print the listing of commands available for the script. |
| -p, –profile | Specify the tiling profile you would like these tiles to be created in. Valid options are mercator or geodetic. |
| -e, –resume | Instruct the script to not overwrite tiles that have already been created. This is a **mandatory** flag when using default multiprocessing. |
| -z, –zoom | The zoom levels to create. Allows the tiler to make tiles past the default zoom level that GDAL detects. (Format: '2-5' or '10') |
| -a, –srcnodata | Specify the RGB value that gdal2tiles should convert to transparency. Typical value should be '0,0,0'. |

## 3.2 Examples

- Create a folder of tiles in the mercator projection based on a GeoTiff image named WhiteHorse.tif and name the folder 'WhiteHorse_tiles':

```
python gdal2tiles_parallel.py
   -p mercator -e
   /data/raw/WhiteHorse.tif /data/tiles/mercator/WhiteHorse_tiles
```

- Create a folder of tiles for zoom level 15 in the mercator projection based on a GeoTiff image named WhiteHorse.tif and name the folder 'WhiteHorse_tiles':

```
python gdal2tiles_parallel.py
   -p mercator -e -z 15
   /data/raw/WhiteHorse.tif /data/tiles/mercator/WhiteHorse_tiles
```

- Create a folder of tiles in the geodetic projection based on a MrSID image named FortBelvoir_201307_A6.sid and name the folder belvoir_tiles. Also, assigns the NODATA transparency to the RGB color value of 0,0,0:

```
python gdal2tiles_parallel.py
   -p geodetic -e -a 0,0,0
   /data/raw/FortBelvoir_201307_A6.sid /data/tiles/belvoir_tiles
```

- Create a folder of tiles in the geodetic projection based on a MrSID image named FortBelvoir_201307_A6.sid and name the folder belvoir_tiles. Also, assigns the NODATA transparency to the RGB color value of 255,0,0:

```
python gdal2tiles_parallel.py
   -p geodetic -e -a 255,0,0
   /data/raw/FortBelvoir_201307_A6.sid /data/tiles/belvoir_tiles
```

- Create a folder of tiles for zoom levels 10 through 13 in the geodetic projection based on a MrSID image named FortBelvoir_201307_A6.sid and name the folder belvoir_tiles. Also, assigns the NODATA transparency to the RGB color value of 0,0,0:

```
python gdal2tiles_parallel.py
```

```
-p geodetic -e -a 0,0,0
-z 10-13
/data/raw/FortBelvoir_201307_A6.sid /data/tiles/belvoir_tiles
```

# 4 Caveats & Known Issues

- Gdal2tiles_parallel.py currently only outputs tiles as full-color PNGs. This is so that transparency can be preserved. When used in conjunction with tiles2gpkg_parallel.py a user can make a geopackage with either PNGs, JPEGs, or a mixture of both.
- Since this script utilizes multiprocessing, it does not exit cleanly when interrupted. For example, if Ctrl+D (KeyboardInterrupt) while the script is executing, it will not completely stop but instead produce profuse error messages. To kill the process, close the terminal windows on Windows or, on Linux, send the process to the background (Ctrl+Z) then kill the job with `kill -9 $(jobs -p)`.
- Gdal2tiles_parallel.py only creates tiles with a lower-left tile origin.