Instructions For tiles2gpkg_parallel.py

Steven D. Lander, Reinventing Geospatial

December 15, 2014

1 Purpose

This document covers installation of dependencies, in-depth explanations of command-line arguments, and usage examples for tiles2gpkg_parallel.py.

The file tiles2gpkg_parallel.py is a Python script that accepts a folder of tiles in TMS format (z/x/y) and outputs an Open Geospatial Consortium (OGC)- compliant Geopackage. The script leverages as much hardware capability as possible in order package this data.

For more information about the OGC Specification for Geopackage, please visit the Open Geospatial Consortium Website: `http://www.opengeospatial.org/standards/geopackage`.

2 Installing Dependencies

Tiles2gpkg_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 the Python Imaging Library (PIL) in order to allow fine-grain control of the MIME type (PNG/JPEG) of each individual tile that resides within the output Geopackage. Without PIL, the script will simply detect the image type of the input tile and maintain that MIME type when it is stored to the Geopackage.

2.1 Windows

In order to run tiles2gpkg_parallel.py on a Windows environment, install the following:

- The latest version of Python 2.7 for either x86 (32 bit) or x86_64 (64 bit): `https://www.python.org/downloads/windows/`.
- The Python Imaging Library pre-compilied binary installer for your correct Python version (2.7) AND architecture (32/64 bit): `http://www.lfd.uci.edu/~gohlke/pythonlibs/#pillow`.

  NOTE: PIL has been deprecated in favor of Pillow, a continuation of the project. They are 100% interchangable but prefer Pillow whenever you can.

2.2 Linux

In order to run tiles2gpkg_parallel.py on a Linux environment, instructions will differ slightly by distribution. Most newer Linux distributions have Python 2.7 as an option in their default 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. Install the following:

- 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. PIP is necessary for the next step.

- The Python Imaging Library python module. On Linux, PIP will download the source code for PIL or Pillow and then install it on your system. For Debian-based distributions type the following into a command line:

  ```
  sudo pip install Pillow
  ```

  This will install Pillow, the successor to the deprecated version of the Python Imaging Library. PIP will attempt to compile the Pillow source code for your system using the python2.7-dev headers and the compilation tools contained within the base-devel package.

3 Usage

3.1 Command Line Arguments

Tiles2gpkg_parallel.py supports additional functionality via command line arguments provided to the script at the time it is executed. Following is a outline of each one and their purpose:

-h	Print the listing of commands available for the script.
-tileorigin	Specify the origin of the tiles contained within the input data folder. Gdal2tiles.py creates tiles referenced by the bottom-left corner which follows TMS convention. Other tile providers can create tiles with a tile origin of upper-left. Valid options are ul, ll, nw, or sw. The default option is ll for lower-left.
-srs	Specify the spatial reference system of the tiles contained within the input data folder. This could also be called the tile grid profile. Valid options are 3857 (mercator), 4326 (geodetic), and 3395 (global geodetic). The default value for this field is 3857.
-imagery	Convert the MIME type of the tiles on-disk to a new type when they are stored in the Geopackage. Valid options are source, mixed, png, and jpeg. Specifying mixed mode will convert all tile images in the source folder that do not have transparency to JPEG with compression enabled for space savings. Specifying source mode will preserve the file type of the input tile images. The default value for this field is source.
-q	When the -imagery flag is set to either mixed or jpeg, this flag specifies the jpeg quality value. Acceptable values are from 1-100 inclusive. Lower numbers result in smaller size images but greatly reduced image quality.
-T	By default, tiles2gpkg_parallel.py takes advantage of all the processors available to the hardware that it is executed on. This can mean that other computing tasks on the machine may suffer depending on the size of the packaging job. The -T flag disables this behavoir and only uses a single-core process to execute the job.

3.2 Examples

- Create a geopackage from a folder of tiles named WhiteHorse in the geodetic tile profile, and name the new Geopackage whitehorse.gpkg:

  ```
  python tiles2gpkg_parallel.py
     -srs 4326
     /data/tiles/geodetic/WhiteHorse /data/geopackage/whitehorse.gpkg
  ```

- Create a geopackage from a folder of tiles named belvoir in the mercator profile and changing the tile images to a mix of PNG and JPEG images:

  ```
  python tiles2gpkg_parallel.py
     -srs 3857 -imagery mixed
     /data/tiles/mercator/belvoir /data/geopackage/belvoir-3857.gpkg
  ```

- Create a geopackage from a folder of tiles named gnc in the world mercator profile with a tile origin of upper-left and also converting the tile images to JPEG with 50% quality:

```
python tiles2gpkg_parallel.py
   -srs 3395 -tileorigin ul -imagery jpeg -q 50
   /data/tiles/world-mercator/gnc /data/geopackage/gnc-wm.gpkg
```

4 Caveats & Known Issues

- Currently it is possible to provide tiles2gpkg_parallel.py with a folder of tiles with different types of data in it. For example, an input folder of tiles could have tiles for the world at very high zoom level (0-3) but also tiles at a very low zoom level (18-19). The script would only describe these tiles for the world and thus confuse a Geopackage viewer about where the rest of the tiles in the package are located.
- Currently the script does not support making a Geopackage with multiple raster tiles tables.
- The script only creates raster tiles at the moment and does not support vector features nor vector tiles. Plans for inclusion of vector tiles in the future are pending.