GTiff2Tiles.Console

GTiff2Tiles.Console is a simple console application, that implements methods from **GTiff2Tiles.Core** to create tiles. The app is available to download from GitHub Releases Page and Docker Images are available on Docker Hub and GitHub Packages Feed.

Supports **only GeoTIFF** as input data and creates **geodetic or mercator** tiles on output in **tms** or **non-tms** (*Google maps like*) structure. Any **GeoTIFF** (with less, than **5 bands**) on input is supported, if it's not **EPSG:4326** or **EPSG:3857**, it'll be converted to your selected target coordinate system and saved inside **temp** directory before cropping.

Requirements

Application runs on **Linux x64** (*tested on Ubuntu 18.04+*) and **Windows x64** (*tested on Win 7 SP1+*) operating systems.

If you're using **Windows 7 SP1**, you can experience weird error with **GDAL** packages. It's recommended to install KB2533623 to fix it. You can read about this Windows update on MSDN.

Usage

Short	Long	Description	Required?
-i	--input	Path to input file	Yes
-o	--output	Path to output directory	Yes
	--minz	Minimum cropped zoom	Yes
	--maxz	Maximum cropped zoom	Yes
	--threads	Threads count, calculates auto by default	No
	--extension	Extension of ready tiles, `.png` by default	No
-t	--temp	Path to temp directory, current directory by default	No
	--tms	Do you want to create tms-compatible tiles? `true` by default	No
-c	--coordinates	Target tiles coordinate system, `geodetic` by default	No
	--interpolation	Interpolation of ready tiles, `lanczos3` by default	No
-b	--bands	Count of bands in ready tiles, 4 by default	No
	--tilecache	How much tiles would you like to store in memory cache? 100 by default	No
-m	--memcache	Maximum size of input files to store in RAM, `2147483648` by default	No
	--progress	Do you want to see the progress? `true` by default	No
	--timeleft	Do you want to see estimated time left? `false` by default	No
	--tilesize	Ready tile's size, 256 by default	No
	--version	Current version	
	--help	Message about command line options	

Minimal example looks like this: `./GTiff2Tiles.Console -i "D:/Examples/Input.tif" -o "D:/Examples/Output" --minz 0 --maxz 12`

Take a look at Start.ps1 **PowerShell** script for automating and more examples of the work. Note, that running this script requires installed **PowerShell** or **PowerShell Core** (also available on **Linux**/**OSX** systems!).

Detailed options description

-i/--input is `string`, representing full path to input **GeoTIFF** file. Please, specify the path in double quotes (`“like this”`) if it contains spaces.

-o/--output is `string`, representing full path to directory, where tiles in will be created. Please, specify the path in double quotes (`"like this"`) if it contains spaces. **Directory should be empty.**

--minz is `int` parameter, representing minimum zoom, which you want to crop.

--maxz is `int` parameter, representing maximum zoom, which you want to crop.

--threads is `int` parameter, representing threads count. By default (if not set) uses calculates automatically, based on your PC.

--extension is a `string`, representing ready tiles extension. By default is set to `.png`. Currently supported extensions are: `.webp`, `.jpg`, `.png`.

-t/--temp is `string`, representing full path to temporary directory. Please, specify the path in double quotes (`"like this"`) if it contains spaces. Inside will be created directory, which name is a **timestamp** in format `yyyyMMddHHmmssfff`. By default – the same directory, where application is located.

--tms is `string`, which shows if you want to create tms-compatible or non-tms-compatible tiles on output. Can have values `true` or `false`. By default is `true`.

-c/--coordinates is a `string`, representing ready tile's coordinate system. By default is `geodetic` (*EPSG:4326*). Supported values: `geodetic`, `mercator`.

--interpolation is a `string`, representing ready tile's interpolation. By default is `lanczos3`. Supported values: `nearest`, `linear`, `cubic`, `mitchell`, `lanczos2`, `lanczos3`.

-b/--bands is `int` parameter, representing count of bands in ready tiles. By default is `4`.

--tilecache is `int` parameter, representing count of tiles to store in RAM to crop them faster (*that's vips stuff*). `1000` by default.

--memcache is `long` parameter, representing maximal size (*in bytes*) of input file to store in RAM to crop it faster. By default is `2147483648` (*which equals to 2Gb*).

--progress is `bool` parameter. If it's set to `true` – you'll see cropping progress in your command line. `true` by default.

--timeleft is a `bool` parameter. If it's set to `true` – you'll see estimated time left before end of cropping after each tile is cropped. `false` (*beware, too much output can slow app down*) by default.

--tilesize is `int` parameter, representing the size of one side (**tiles should be a square, so specifying 2 side's sizes is redundant*) of ready tiles. `256` by default.

Offline docs

Offline docs are also available as pdf and distributed alongside the application.

Build dependencies

- GTiff2Tiles.Core;
- CommandLineParser – 2.8.0;