GTiff2Tiles.Benchmarks

GTiff2Tiles.Benchmarks is a benchmarking project for **GTiff2Tiles.Core**.

The following benchmarks were run at **25.08.2020**.

Requirements

- Docker
- Linux x64/Win 10+ x64

Build dependencies

- GTiff2Tiles.Core;
- BenchmarkDotNet – 0.12.1;
- CommandLineParser – 2.8.0;

Running by yourself

To run the benchmarks by yourself you should build the app **in Release x64 configuration** first. See main and core pages to learn how to build the soultion.

Benchmarks uses the installed docker to pull the **latest** images of **maptiler/engine** and **osgeo/gdal** and then runs the **GTiff2Tiles.Core** benchmarks agains them.

Short	Long	Description	Required?
-i	--input	Path to input file	No
	--version	Current version	
	--help	Message about console options	

-i/--input is `string`, representing path to input **GeoTIFF** file. Please, specify the path in double quotes (`“like this”`) if it contains spaces. The path is **optional**, by default app uses the `(repo_home)/Examples/Input/Benchmarks.tif`. **USE ONLY EPSG:4326 GEOTIFF AS INPUT**!

Simple example (for **pwsh with admin rights**) looks like this: `./GTiff2Tiles.Benchmarks`

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

And last but not the least: if you're changing the `SimpleJob` arguments, not overdo it. **BenchmarkDotNet** can create **VERY BIG** log file in process (*I had a 130Gb file on my system drive while running it and I had to stop*). I also don't recommend to add memory/threading attributes, since they don't analyze inner docker processes and it's kind of useless for this app.

Offline docs

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

Results

Docker Desktop version is an **edge release 2.3.5.0**; uses **WSL2** features.

Used **MapTiler Engine** version is **10.3**, used **gdal2tiles.py** version is **GDAL 3.2.0dev-38e9587ed7fc34d8e145b03a86ca0a2ec655fcce, released 2020/08/25**, used **GTiff2Tiles.Core** version is **2.0.0.589**.

Benchmarks create the **geodetic png 256x256 non-tmscompatible** tiles from **EPSG:4326** input GEOTIFF, resampling is **cubic**, zooms **0-15**, process counter **8**.

maptiler was running with the following arguments: `-srs EPSG:4326 -preset geodetic -resampling cubic -zoom 0 15 -P 8 -f png32 -o outDir in.tif`.

gdal2tiles.py was running with the following arguments: `-s EPSG:4326 -p geodetic -r cubic -z 0-15 --processes 8 in.tif outDir`.

```
BenchmarkDotNet=v0.12.1, OS=Windows 10.0.19041.450 (2004/?/20H1)
Intel Core i7-6700K CPU 4.00GHz (Skylake), 1 CPU, 8 logical and 4 physical cores
.NET Core SDK=5.0.100-preview.7.20366.6
  [Host]     : .NET Core 5.0.0 (CoreCLR 5.0.20.36411, CoreFX 5.0.20.36411), X64 RyuJIT
  Job-GYLFGH : .NET Core 5.0.0 (CoreCLR 5.0.20.36411, CoreFX 5.0.20.36411), X64 RyuJIT

IterationCount=10  LaunchCount=10  WarmupCount=10
```

Method	Mean	Error	StdDev	Median
RunGTiff2Tiles	2.255 s	0.0233 s	0.0672 s	2.250 s
RunGdal2Tiles	12.989 s	0.1660 s	0.4711 s	12.885 s
RunMaptiler	4.948 s	0.0958 s	0.2716 s	4.840 s