

QGIS Application - Bug report #21819

gdal2tiles very slow compared to QGIS 2.18

2019-04-10 10:43 AM - Karsten Tebling

Status:	Feedback	
Priority:	Normal	
Assignee:		
Category:	Processing/GDAL	
Affected QGIS version:	3.6.1	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution:
Crashes QGIS or corrupts data:	No	Copied to github as #: 29634
Description		
<p>I tried to generate tiles for zoom levels 10-11 for a roughly 2GB compressed DOP, with QGIS 3.6.1 it took about 581 minutes to finish. I also tried it with QGIS 2.18.28 and it only took around 8 seconds for the same DOP.</p>		

History

#1 - 2019-04-10 06:18 PM - Giovanni Manghi

- Category changed from GDAL Tools to Processing/GDAL

- Status changed from Open to Feedback

Interesting:

same exact command generated in both versions?

can you see any difference in running the command directly from the command line?

#2 - 2019-04-12 10:22 AM - Karsten Tebling

Giovanni Manghi wrote:

Interesting:

same exact command generated in both versions?

can you see any difference in running the command directly from the command line?

2.18.28:

```
gdal2tiles.bat -s EPSG:25832 -z 10-11 -w leaflet "dop20rgb.tif" "[temporäre Dateij]"
```

3.6.1:

```
python3 -m gdal2tiles -p mercator -z 10-11 -w leaflet -r average -s EPSG:25832 -a 0.0 "dop20rgb.tif"
```

```
C:/Users/User/AppData/Local/Temp/processing_1dd1e8de74f7470f9aaf10bd70148/a360afbda0ae4404a981ef0a8db4242d/OUTPUT
```

I have not tried running from the command line, OS is Windows 7 64bit. I don't have python3 installed on the PC, so I think command line won't work unless I set all the environment variables to use the one bundled with QGIS... but I don't know how to do this - can you tell me how I can test this? I installed QGIS via standalone setup, I didn't use osgeo4w.

#3 - 2019-04-12 02:42 PM - Giovanni Manghi

Karsten Tebling wrote:

Giovanni Manghi wrote:

Interesting:

same exact command generated in both versions?

can you see any difference in running the command directly from the command line?

2.18.28:

gdal2tiles.bat -s EPSG:25832 -z 10-11 -w leaflet "dop20rgb.tif" "[temporäre Dateij]"

3.6.1:

python3 -m gdal2tiles -p mercator -z 10-11 -w leaflet -r average -s EPSG:25832 -a 0.0 "dop20rgb.tif"

C:/Users/User/AppData/Local/Temp/processing_1dd1e8de74f7470bfd9aaf10bd70148/a360afbda0ae4404a981ef0a8db4242d/OUTPUT

you must compare the **actual** (gdal) commands that are run, you can see them at the top of the Processing/tool log, after you run the tool

I have not tried running from the command line, OS is Windows 7 64bit. I don't have python3 installed on the PC, so I think command line won't work unless I set all the environment variables to use the one bundled with QGIS... but I don't know how to do this - can you tell me how I can test this? I installed QGIS via standalone setup, I didn't use osgeo4w.

any QGIS installation comes with an installation of Python.

In your QGIS entry in the start menu (assuming you are on Windows) you have a shortcut to a "OSgeo4W shell", from there you can run any gdal utilities program.

If you want to help troubleshoot this you should grab the (gdal) commands as they are created/run by QGIS and try them directly in the shell, and see if the difference of performance is still a thing.

If you are uncomfortable with all this then please attach a sample project with data and show us a screenshot of the tool options you are using.

#4 - 2019-04-12 03:03 PM - Karsten Tebling

Giovanni Manghi wrote:

Karsten Tebling wrote:

Giovanni Manghi wrote:

Interesting:

same exact command generated in both versions?

can you see any difference in running the command directly from the command line?

2.18.28:

gdal2tiles.bat -s EPSG:25832 -z 10-11 -w leaflet "dop20rgb.tif" "[temporäre Dateij]"

3.6.1:

```
python3 -m gdal2tiles -p mercator -z 10-11 -w leaflet -r average -s EPSG:25832 -a 0.0 "dop20rgb.tif"
```

```
C:/Users/User/AppData/Local/Temp/processing_1dd1e8de74f7470fbdf9aaf10bd70148/a360afbda0ae4404a981ef0a8db4242d/OUTPUT
```

you must compare the **actual** (gdal) commands that are run, you can see them at the top of the Processing/tool log, after you run the tool

I have not tried running from the command line, OS is Windows 7 64bit. I don't have python3 installed on the PC, so I think command line won't work unless I set all the environment variables to use the one bundled with QGIS... but I don't know how to do this - can you tell me how I can test this? I installed QGIS via standalone setup, I didn't use osgeo4w.

any QGIS installation comes with an installation of Python.

In your QGIS entry in the start menu (assuming you are on Windows) you have a shortcut to a "OSgeo4W shell", from there you can run any gdal utilities program.

If you want to help troubleshoot this you should grab the (gdal) commands as they are created/run by QGIS and try them directly in the shell, and see if the difference of performance is still a thing.

If you are uncomfortable with all this then please attach a sample project with data and shows us a screenshot of the tool options you are using.

I will try it on Monday, but I can't share the files because I don't have the rights to do that.

#5 - 2019-04-18 08:10 AM - Karsten Tebling

It is displaying the same commands or I'm looking at the wrong windows. I also opened the processing window, but all I could find there was in 2.18:

GDAL execution console output

Generating Base Tiles:

0...10...20...30...40...50...60...70...80...90...100

Generating Overview Tiles:

0...10...20...30...40...50...60...70...80...90...100

#6 - 2019-04-18 02:02 PM - Giovanni Manghi

Karsten Tebling wrote:

It is displaying the same commands or I'm looking at the wrong windows. I also opened the processing window, but all I could find there was in 2.18:

GDAL execution console output

Generating Base Tiles:

0...10...20...30...40...50...60...70...80...90...100

Generating Overview Tiles:

0...10...20...30...40...50...60...70...80...90...100

have you tried run the commands in the consoles (the osgeo4w console if you are on Windows), and see if there is any difference?