Subject: Re: GDAL/OGR library 'import' now in sandbox
Posted by Tom1 on Mon, 21 Dec 2015 15:01:24 GMT
View Forum Message <> Reply to Message

Hi Mirek,

I have initially tried to connect to your GDAL encapsulation for reading raster maps.

I'm missing at least access to:

```
Band::GetColorTable();
Band::GetColorTable()->GetColorEntryCount();
Band::GetColorTable()->GetColorEntry();
Band::GetColorTable()->GetPaletteInterpretation();
```

and also a way to read the data block directly in the native buffer format. Some rasters are really big and in one byte per pixel format. If these are readily converted to int or double, it is easy to run out of memory on 32 bit systems.

--

Before we proceed with this any further, may I ask why the encapsulation? While rewriting my calls to match Gdal:: instead of original GDAL interface, I found similar complexity but with partially unfamiliar interface. I could better understand this encapsulation if we had e.g. a GeoImage:: class and even a TiledGeoImage:: class, which were automatically loaded from raster files using Gdal:: and representing the raster content as an Image:: that has georeferencing data directly available. (Of course the 32 bits per pixel RGBA data storage of Image makes this less than optimal in memory for large files, but a very nice solution in principle.) Maybe storing the raster data in original pixel depth plus a palette in a GeoImage:: and then adding Image generation for selected area with desired transform would be a better solution.

Best regards,

Tom