

**FOSS4G**

Prizren, 2023

# State of GDAL

## GDAL 3.6 & 3.7

Even Rouault  
*SPATIALYS*

June 28th 2023



# GDAL/OGR : Introduction

- GDAL? Geospatial Data Abstraction Library. The swiss army knife for geospatial.
- Read and write Raster (GDAL) and Vector (OGR) datasets
- 250 (mainly) geospatial formats and protocols.
- Widely used



(> 100 <http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal>)

- MIT Open Source license (permissive)

# *CMake* **build system**

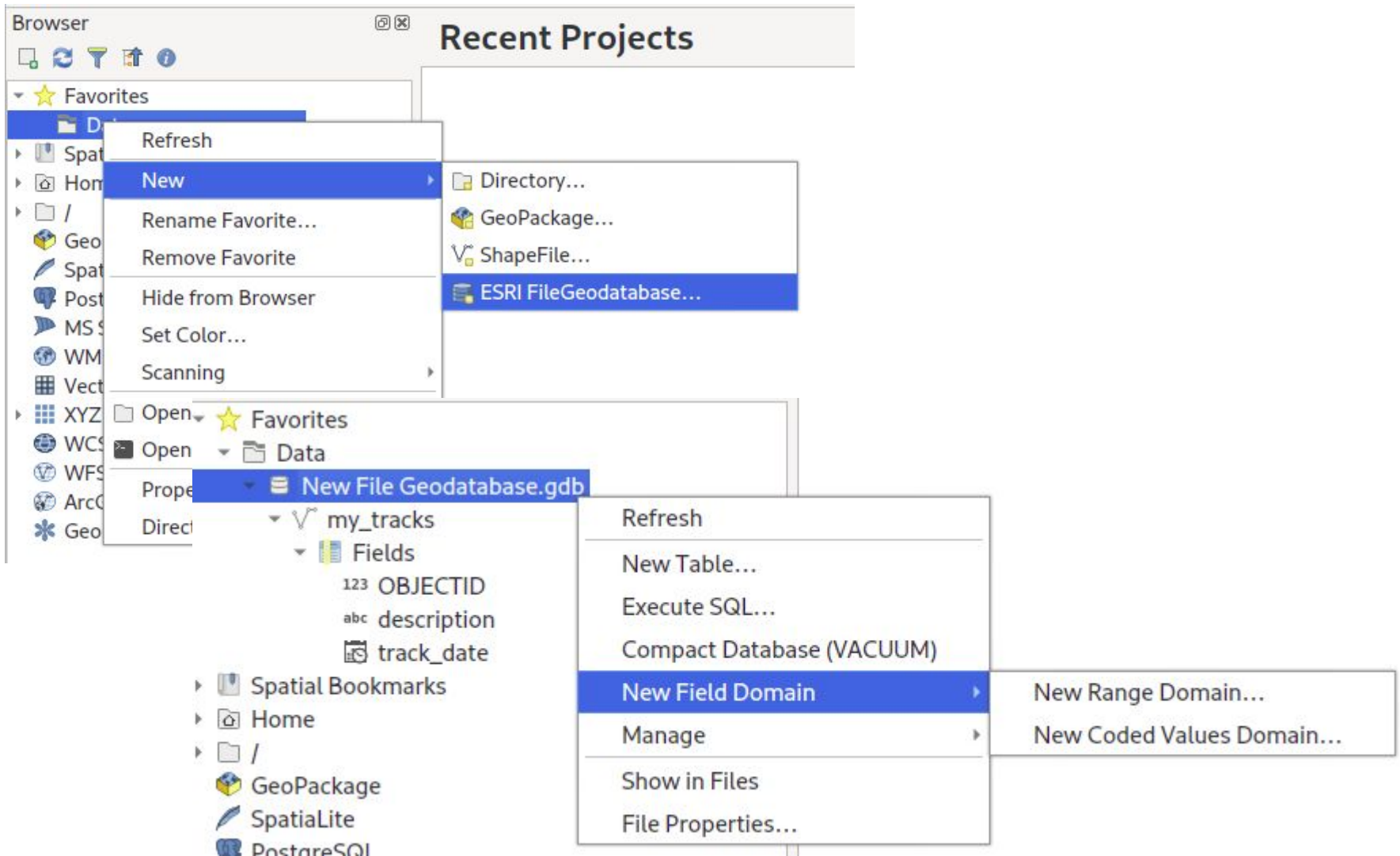
- In GDAL 3.5, CMake as a new build system
- to replace long time existing autotools-based and Visual Studio's nmake ones
- Since GDAL 3.6, CMake is now only the build system supported
- Many integrations available with build systems such as vcpkg or conda-forge
- Extensive documentation of all the options at [https://gdal.org/development/building\\_from\\_source.html](https://gdal.org/development/building_from_source.html)

# (Open)FileGeodatabase enhancements

- Full write and update support for FileGeodatabase vector, in core GDAL 3.6:
  - Creation of new datasets
  - Addition/deletion of vector layers
  - Addition/update/deletion of attributes
  - Addition/update/deletion of features
  - Creation and update of spatial index
  - Creation and update of attribute index
  - Creation/management of field domains
  - Creation and management of layer relationships
  - Database compaction



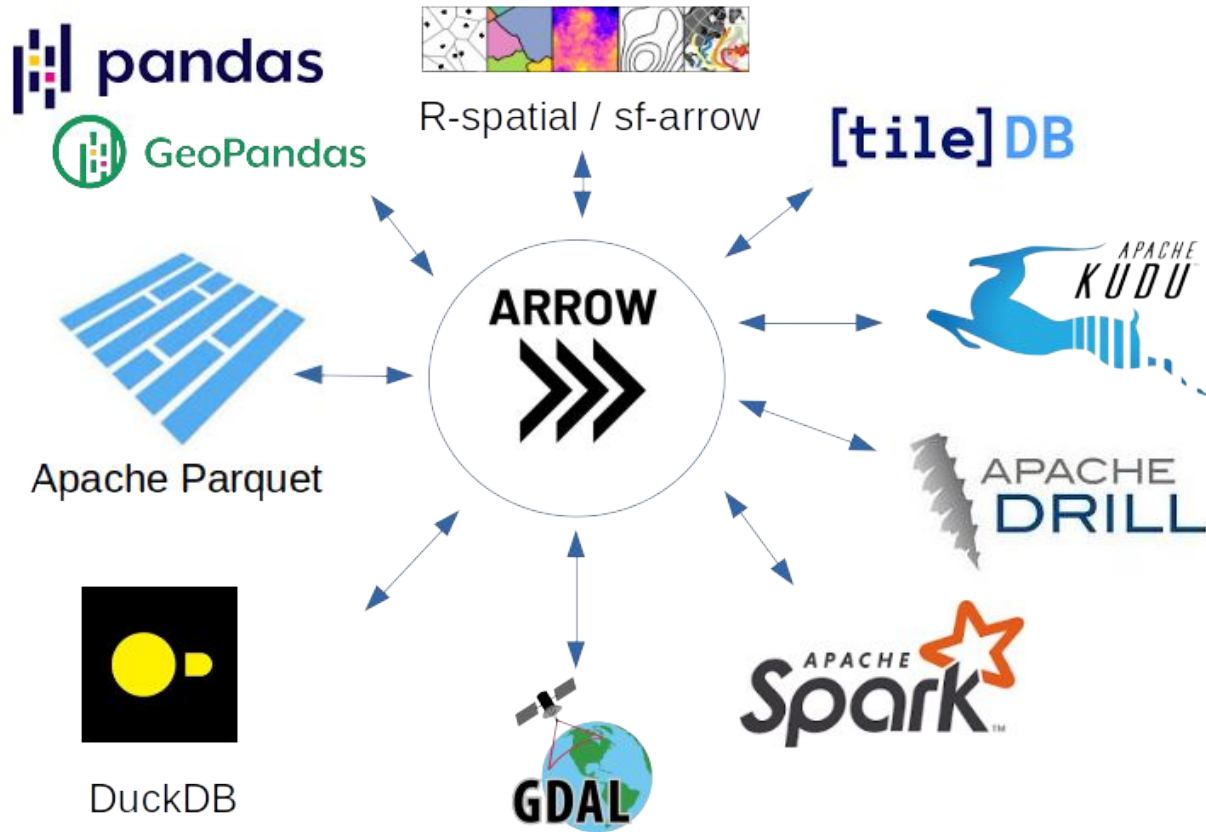
# OpenFileGDB integration in QGIS 3.28



# (Open)FileGeodatabase enhancements

- Prior O.S. effort: ArcRaster Rescue (R. Barnes)
- Read support for FileGeodatabase v10 raster datasets, in core GDAL 3.7
  - Read CRS and georeferencing information
  - Tile-based reading
  - Expose overviews/pyramids
  - Compression methods: uncompressed, LZ77 (Deflate), JPEG and JPEG2000
  - Value attribute tables
- Read support for FileGDB v9 for GDAL 3.8

# Arrow-based columnar oriented read API



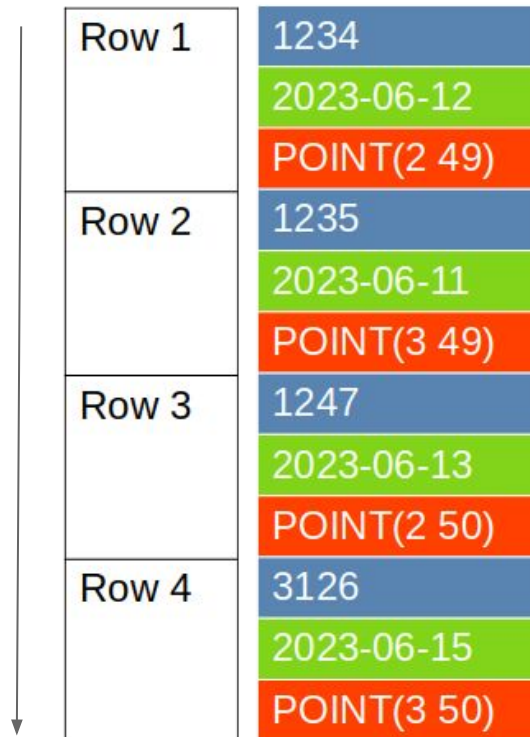
Columnar formats:

- Parquet
- Arrow
- TileDB

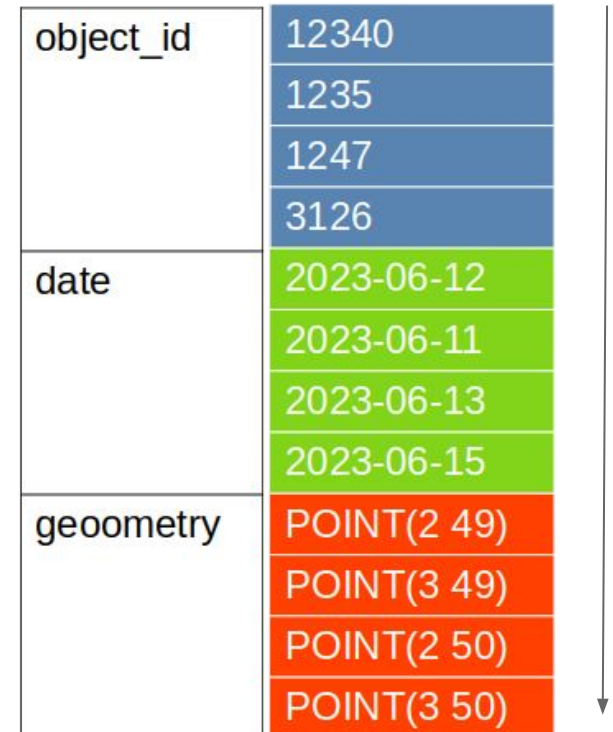
# Column-oriented = information for a given attribute is grouped by many rows

	object_id	date	geometry
Row 1	1234	2023-06-12	POINT(2 49)
Row 2	1235	2023-06-11	POINT(3 49)
Row 3	1247	2023-06-13	POINT(2 50)
Row 4	3126	2023-06-15	POINT(3 50)

Row/feature memory buffer



Arrow columnar memory buffer





# Arrow-based columnar oriented read API

- [https://gdal.org/development/rfc/rfc86\\_column\\_oriented\\_api.html](https://gdal.org/development/rfc/rfc86_column_oriented_api.html)
- `OGRLayer::GetArrowStream()` → `ArrowStream`
- Arrow Stream C interface:
  - `get_schema(stream)` → `ArrowSchema`
  - `get_next(stream)` → `ArrowArray`
- `ds = ogr.Open("my.gpkg")`  
`lyr = ds.GetLayer(0)`  
for batch in `lyr.GetArrowStreamAsPyArrow()`:  
... do something with batch ...  
for batch in `lyr.GetArrowStreamAsNumPy()`:  
... do something with batch ...

# Arrow-based columnar oriented read API

- Generic implementation for all other OGR drivers
  - uses feature-based iteration underneath (`GetNextFeature()`)
  - of course adds some overhead
- Efficient implementation in drivers:
  - (Geo)Parquet/(GeoArrow)
  - TileDB (vector side added in GDAL 3.7)
  - GeoPackage
  - FlatGeoBuf

# Arrow-based columnar oriented read API

- Benchmarks to load a 3.2 million rows (footprint polygons) of 13 attributes each

	GeoParquet	TileDB	GeoPackage	FlatGeoBuf	Shapefile
<b>File size</b>	0.43 GB	1.1 GB	1.67 GB	1.8 GB	2.92 GB
<b>Feature-based iteration</b>	6.2 s	6.0 s	6.6 s	5.0 s	10.3 s
<b>Arrow/Columnar (1 thread)</b>	1.6 s (x 3.9 speedup)	---	2.2 s (x 3 speedup)	3.0 s (x 1.7)	16.8 s (generic impl.) (x 0.61)
<b>Arrow/Columnar (4 threads)</b>	1.0 s (x 6.2 speedup)	1.3 s (x 4.6)	0.7 s (x 9.4)	--	--

## New drivers (GDAL 3.6)

- BASISU and KTX2 read/write raster drivers:
  - Basis Universal Textures: GPU-optimized
  - Khronos Texture Container 2.0
  - Require (forked) basisu open source library
- JPEG-XL standalone read/write driver:
  - JPEG-XL: competitor to AVIF/HEIF/WEBP
  - Lossless and lossy profiles
  - Already available as a GeoTIFF codec
  - Requires open source libjxl library

# New drivers (GDAL 3.7)

- NSIDCbin raster read-only driver:
  - NSIDC=National Snow and Ice Data Centre
  - Sea Ice Concentrations
  - Raw binary format
  - Daily and monthly maps in the north and south hemispheres
- GTFS vector read-only driver:
  - General Transit Feed Specification
  - Common format for public transportation schedules and associated geographic information
  - Expose text tables augmented with geometries

## **ogrinfo : API**

- Available through the GDALVectorInfo() C API call
- `gdal.VectorInfo( dataset / filename )` in Python

# ogrinfo: ... and JSON output !

```
$ ogrinfo -json poly.shp
```

```
{
  "layers":[
    {
      "name":"poly",
      "metadata":{
      },
      "geometryFields":[
        {
          "name":"",
          "type":"Polygon",
          "nullable":true,
          "extent":[
            478315.53125,
            4762880.5,
            481645.3125,
            4765610.5
          ],
          "coordinateSystem":{
            [...]
          }
        }
      ],
    }
  ],

```

```
"featureCount":10,
"fields":[
  {
    "name":"AREA",
    "type":"Real",
    "width":12,
    "precision":3,
    "nullable":true,
    "uniqueConstraint":false
  },
  {
    "name":"EAS_ID",
    "type":"Integer64",
    "width":11,
    "nullable":true,
    "uniqueConstraint":false
  },
  {
    "name":"PRFEDEA",
    "type":"String",
    "width":16,
    "nullable":true,
    "uniqueConstraint":false
  }
]
```

# Miscellaneous

- ogr\_layer\_algebra.py: promoted as a supported utility
- OGRLayer::UpsertFeature() and UpdateFeature()
- PNG decompression: 1.7 to 2.0 times faster when building GDAL against libdeflate
- Direct access to compressed raster data: extraction of JPEG/JPEGXL/WEBP/etc. Tiles:
  - Implemented in GeoTIFF, VRT, JPEG, JPEGXL and WEBP drivers
  - [https://gdal.org/development/rfc/rfc90\\_read\\_compressed\\_data.html](https://gdal.org/development/rfc/rfc90_read_compressed_data.html)



# GeoTIFF multi-threaded decompression

- Enabled if GDAL\_NUM\_THREADS configuration option/env. variable is set
- 3-band Byte tiled 10,000 x 5,000 large file:

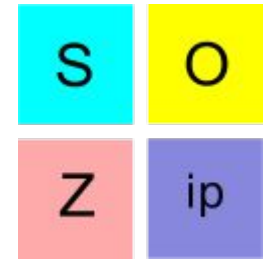
<b>Compression</b>	<b>Time (ms) single-threaded</b>	<b>Time (ms) 12 threads</b>	<b>Speed-up factor</b>	<b>File size (MB)</b>
<b>None</b>	210	112	1.9	157
<b>Deflate</b>	310	137	2.3	11
<b>JPEG (lossy)</b>	312	138	2.3	1.1
<b>LZW</b>	390	142	2.7	12
<b>WebP lossy</b>	411	148	2.8	0.5
<b>WebP lossless</b>	558	157	3.6	7.7
<b>JPEG-XL lossy</b>	1415	368	3.8	0.7
<b>JPEG-XL lossless</b>	6810	1368	5.0	8.6

- 11977 x 8745 x 8 bands UInt16 tiled deflate compressed file

Deflate	5700	1109	5.1	990
---------	------	------	-----	-----

# SOZIP: Seek-optimized ZIP format

- Open-specification on top of venerable .zip format
- Random access in large compressed files directly from a .zip file without prior decompression
- Fully backwards compatible
- Use cases: GeoPackage, FlatGeoBuf, Shapefile, ...
- Available in GDAL 3.7 / “sozip” command line utility



⇒ More at 4h30 today at “SOZip: using directly (geospatial) large compressed files in a ZIP archive!” session

# GDAL 3.8 preview

- New vector driver to read and write the OGC Feature and Geometries JSON (JSON-FG) candidate standard
- New vector driver to read and write PMTiles dataset with vector tile. PMTiles: cloud-friendly tile container
- Multi-dimensional API in TileDB driver

# Thanks to GDAL sponsors!

- Gold level:



- Silver level:



- Bronze level:



- Supporter level:





# Questions?

Links:

<http://gdal.org/>

Contact: [even.rouault@spatialys.com](mailto:even.rouault@spatialys.com)

