



State of GDAL

GDAL 3.1, 3.2, 3.3...

Even Rouault
SPATIALYS



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/X Open Source license (permissive)

GDAL 3.1: COG driver

- New COG (Cloud Optimized GeoTIFF) driver / more efficient COG generation:
 - `gdal_translate -of COG my_source_dataset my_cog.tif`
- Reading done through the classic GeoTIFF driver, with performance improvements to reduce the number of HTTP GET requests
- “Cloud optimized formats for raster and vectors explained” (Pirmin Kalberer):
<https://callforpapers.2021.foss4g.org/foss4g2021/talk/ZJYRAS/>

GDAL 3.1: Multidimensional API

- New API to read & write hierarchical and multidimensional arrays (3D, 4D): netCDF, HDF4, HDF5, GRIB, Memory, VRT
- Available in C, C++ and Python
- Two new command line utilities:
 - gdalmdiminfo
 - gdalmdimtranslate
- ZARR support available in 3.4

GDAL 3.1 other improvements:

- **New raster drivers:**
 - EXR: read/write driver (OpenEXR library)
 - ISG: for geoid models of the International Service for the Geoid
 - RDB: read-only, for RIEGL Database .mpx RDB 2 files (prop SDK)
- **New vector drivers:**
 - FlatGeoBuf: read-support and creation. Good candidate for a Cloud-optimized vector format
 - MapML: read/write driver for experimental web spec
- **Utilities:**
 - gdalwarp: can work with any format that has some write capabilities
 - gdal_viewshed: viewshed / intervisibility computation

gdal_viewshed output



GDAL 3.1 other improvements:

- RFC76: add capability of writing (read-only) vector drivers in Python
- OAPIF driver (renamed from WFS3): updated to OGC API - Features 1.0 core spec
- GTiff: improve performance of internal overview creation
- GTiff: GeoTIFF 1.1 support
- Shapefile driver: add read/creation/update support for .shz and .shp.zip
- netCDF vector: read/write support for CF-1.8 Encoded Geometries

GDAL 3.2:

- **New GDAL drivers:**

- OGCAPI: tiles/maps/coverage raster/vector experimental driver
- ESRIC: ESRI bundle cache read-only driver
- HEIF: read-only driver for HEIF/HEIC file. Requires libheif
- TGA: read/only driver to read TGA image file format

- **New OGR driver:**

- LVBAG: read-only support for Dutch LVBAG/Kadaster 2.0 vector format

- **New utility:**

- gdal_create: to create/initialize a new raster file

GDAL 3.2: other improvements

- Multi-threaded overview computation (if GDAL_NUM_THREADS set)
- Faster GTIFF Deflate compression/decompression through libdeflate
- COG driver: TILING_SCHEME creation option
- OpenFileGDB driver: add support for using spatial indexes
- BAG driver: multiple improvements
- FITS driver: multiple improvements (MEF and binary table support)
- NITF driver: support for SNIP (Spectral NITF profile) TREs
- OGRFieldDefn: support UNIQUE constraint
- OGRFieldDefn: support a AlternativeName (alias) property
- Removed functionality:
 - Python bindings: old-style "import gdal" is no longer available. Use "from osgeo import gdal" instead

GDAL 3.3: additions

- STACTA: raster driver to read Spatio-Temporal Asset Catalog Tiled Assets (<https://github.com/stac-extensions/tiled-assets>)
- Add /vsiadls/ virtual file system for Azure Data Lake Storage Gen2
- Add automatic loading of configuration options from a file (/etc/gdal/gdalrc, ~/.gdal/gdalrc). See <https://gdal.org/user/configoptions.html#gdal-configuration-file>
- Enumerated, constraint and glob field domains in MEM, FileGDB/OpenFileGDB and GeoPackage drivers
- Add a gdal-utils Python package

GDAL 3.3: deprecation and removals

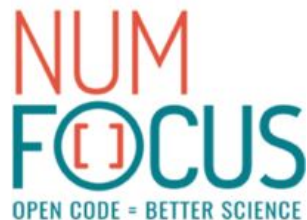
- Drop Python 2 support in favor of Python 3.6
- Removal of BNA, AeronavFAA, HTF, OpenAir, SEGUKOOA, SEGY, SUA, XPlane, BPG, E00GRID, EPSILON, IGNFHeightASCIIGrid, NTV1 drivers. Moved to (unsupported) <https://github.com/OSGeo/gdal-extra-drivers> repository.
- **Deprecation:**
 - Disable by default raster drivers DODS, JPEG2000(Jasper), JPEGLS, MG4LIDAR, FUJIBAS, IDA, INGR
 - and vector driver ARCGEN, ArcObjects, CLOUDANT, COUCHDB, DB2, DODS, FME, GEOMEDIA, GTM, INGRES, MONGODB, REC, WALK at runtime
 - ...unless the `GDAL_ENABLE_DEPRECATED_DRIVER_{drivename}` configuration option is set to YES. Those drivers are planned for removal in GDAL 3.5
 - Perl bindings are deprecated. Removal planned for GDAL 3.5. Use `Geo::GDAL::FFI` instead

GDAL 3.4 preview

- STACIT driver (Spatio-Temporal Asset Catalog Items), using the projection extension specification. Uses VRT internally
- ZARR read/write driver:
 - Zarr V2 and experimental V3 specifications
 - Classic 2D and multidimensional API

Sponsorship program

- GDAL: large code base (1.5 MLOC) with significant drive-by contributions but small pool of people with maintainer role ⇒ not sustainable
- Setup of a sponsorship program to fund maintenance activities: <https://gdal.org/sponsors/>
- Using NumFOCUS (<https://numfocus.org/>), a 501(c)(3) US charity, under the “Grantor-Grantee” fiscal sponsorship model. For other purposes, GDAL remains a OSGeo project



Sponsorship program

- ~ 300k USD / year raised
- Funding several co-maintainers (2 currently)
 - ⇒ Bug triaging and fixing, pull requests reviews, maintenance of continuous integration setups, security fixes, release management
- Funding of contributors on ad-hoc topics (in priority for “infrastructure” / non-feature oriented work)
- Uptream projects like PROJ, libgeotiff, libtiff will benefit from it
- Governance and use of those funds defined in:
 - https://gdal.org/development/rfc/rfc80_numfocus_relationship.html
 - https://gdal.org/development/rfc/rfc83_use_of_project_sponsorship.html

Current sponsors

- Gold level:



- Silver level:



- Bronze level:



- Supporter level:

Myles Sutherland

Umbra

Space Intelligence



Questions?

Links:

<http://gdal.org/>

Contact: even.rouault@spatialys.com

