GDAL 2.2
What's new ?

Even Rouault - *SPATIALYS*

July 19th 2017

Plan

- Introduction to GDAL/OGR
- Community
- GDAL 2.2 : new features
- Future directions

GDAL/OGR : Introduction

- GDAL? Geospatial Data Abstraction Library. The swiss army knife for geospatial.
- Read and write Raster (GDAL) and Vector (OGR) datasets
- More than 200 (mainly) geospatial formats and protocols.

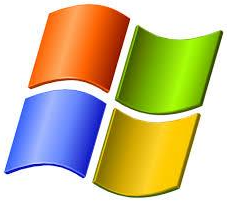
Widely used (FOSS & proprietary)

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

GDAL/OGR : Introduction

- Started in 1998 by Frank Warmerdam
- A project of OSGeo since 2008
- MIT/X Open Source license (permissive)
- > 1M lines of code for library + utilities, ...
- > 150K lines of test in Python

- Multi-OS:


Main features

- Utilities for format conversion, reprojection, subsetting, mosaicing, interpolating, indexing, tiling…
- Support datasets of arbitrary size with limited resources
- C++ library with C API
- Language bindings:

- Can work with local, remote (/vsicurl, /vsis3), compressed (/vsizip/, /vsigzip/, /vsitar), in-memory (/vsimem) files

Main features

- Algorithms: rasterization, vectorization (polygon and contour generation), DEM algorithms (hillshading, slope, etc..), null pixel interpolation, filters
- For vectors, SQL capabilities
  - OGR SQL or SQLite for all formats
  - SQL pass-through for RDBMS

Community activity

- 58 developers with SVN write access
  - 14 active in 12 last months
- 60 occasional contributors active in 12 last months
- 2280 subscribers to gdal-dev. 2160 messages / 12 last months
- ~ 390 tickets created / 12 last months (6960 total). 626 opened
- 2 GSoC students in 2016

GDAL/OGR 2.1: in a nutshell

- 2.1.0 in May 2016. 2.1.4 release in June 2017
- 6 RFCs implemented in 2.1 cycle
  - Geographical Network Model
  - Utilities as library functions
  - Management of measured geometries(M dimension)
- 7 new raster drivers (WMTS, Sentinel2)
- 5 new vector drivers (mongoDB)
- New virtual file system: /vsis3/

GDAL/OGR 2.2

- 2.2.0 in May 2017. 2.2.1 release in June 2017 (available in OSGeo4W and Debian testing/sid)
- Very intense cycle: 3731 commits
- 4 new raster drivers:
  - DERIVED driver: read-support. Expose subdatasets in a new metadata domain, DERIVED_SUBDATASETS, for derived quantities for complex (real+imaginary) data types: amplitude, intensity, log-amplitude, phase, …
  - JP2Lura: read/create support for JPEG2000, through (commercial) Luratech SDK. Trick for Float32 encoding

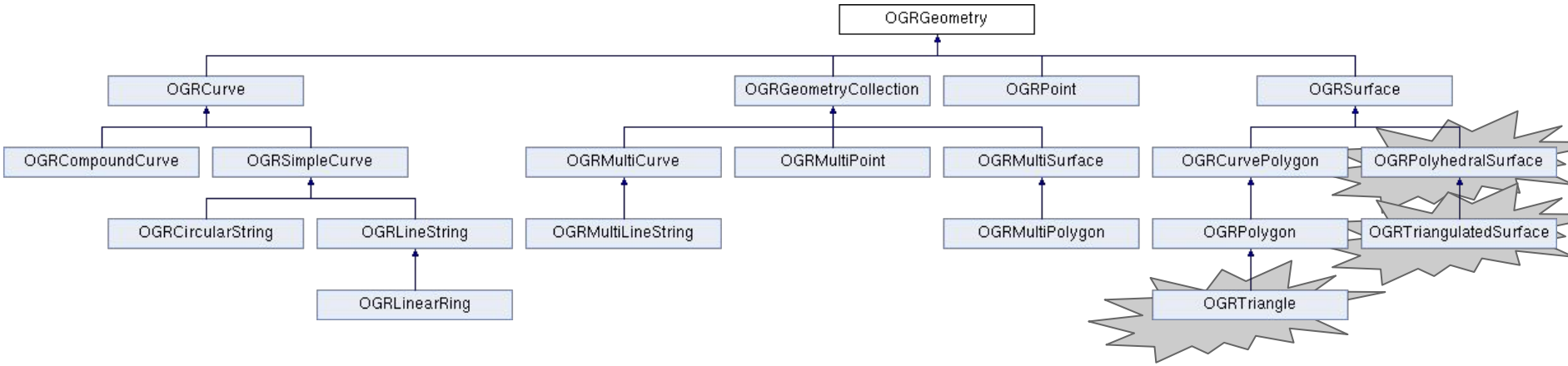
GDAL/OGR 2.2

- Raster drivers (cont'd)
  - PRF: read support for internal format of Racus PHOTOMOD software
  - RRASTER: read support for .grd/.gri files of the 'raster' module of the R language.

- 3 new vector drivers:
  - CAD/DWG driver based on libopencad (X/MIT). Restricted to DWG R2000/AC1015. (GSoc 2016)
  - DGNv8 driver: read-write support for DGN 8.0 format (using Teigha ODA libraries)
  - GMLAS driver: read-write support. XML/GML driver driven by Application Schemas

RFC 64: Triangle, Polyhedral surface and TIN (Triangulated Irregular Networks) (GSoc 2016)

- Geometry model fully supporting ISO SQL/MM Part 3



- Implemented in Shapefile, PostGIS, GML and DXF
- Used SFCGAL for 3D operations

- Backward compatibility impacts: see
  - https://svn.osgeo.org/gdal/branches/2.2/gdal/MIGRATION_GUIDE.TXT
  - https://trac.osgeo.org/gdal/wiki/rfc64_triangle_polyhedralsurface_tin

GMLAS: support for schemas of complex features

Context:

- Inspire, GeoSciML, GroundWaterML2, etc.. : datasets using GML Complex Features schemas
- Complex Features: multiple geometries, nesting of elements, repeated elements, etc...
- Server-side support (GeoServer, Deegree), but little client-side support
- Existing OGR GML driver restricted to simple features

GMLAS: support for schemas of complex features

⇒ New GMLAS (GML Application Schema) read/write driver:

- Analyze the schema to create a relational model, that can be consumed by target databases (PostGIS, Spatialite, etc…)
- Arbitrary big documents can be read and converted (potentially GB)
- Tunable behaviour
- Write side: can regenerate a GML/XML from an imported database that has been modified

Related work: QGIS GML Application Schema toolbox: https://github.com/BRGM/gml_application_schema_toolbox

VRT pixel functions in Python

```
<VRTDataset rasterXSize="20" rasterYSize="20">
<SRS>EPSG:26711</SRS><GeoTransform>440720,60,0,3751320,0,-60</GeoTransform>
<VRTRasterBand dataType="Byte" band="1" subClass="VRTDerivedRasterBand">
        <PixelFunctionType>multiply</PixelFunctionType>
        <PixelFunctionLanguage>Python</PixelFunctionLanguage>
        <PixelFunctionArguments factor="1.5"/>
        <PixelFunctionCode><![CDATA[
import numpy as np
def multiply(in_ar, out_ar, xoff, yoff, xsize, ysize, raster_xsize,
                raster_ysize, buf_radius, gt, **kwargs):
factor = float(kwargs['factor'])
out_ar[:] = np.round_(np.clip(in_ar[0] * factor,0,255))
]]>
        </PixelFunctionCode>
<SimpleSource><SourceFilename relativeToVRT="1">byte.tif</SourceFilename></SimpleSource>
</VRTRasterBand></VRTDataset>
```

See http://www.gdal.org/gdal_vrttut.html#gdal_vrttut_derived_python

Other RFCs

- RFC 63: Sparse dataset improvements

  Add GDALRasterBand::GetDataCoverageStatus() and implement it in GTiff and VRT drivers

- RFC 66: OGR random layer read/write capabilities
  - OSM, GMLAS drivers

- RFC 67: add null field state for OGR features, in addition to unset fields

{ "type": "Feature", "properties": { "foo": null, "bar": "baz" } } ⇒ foo null

{ "type": "Feature", "properties": { "bar": "baw" } } ⇒ foo unset

  - Impacts GeoJSON / GML
  - Backward compatibility impacts: see
    - https://svn.osgeo.org/gdal/branches/2.2/gdal/MIGRATION_GUIDE.TXT

Other driver changes

- GeoPackage: support for v1.2 standard
  - Attributes data type for geometry-less tables
  - Tiled gridded elevation extension (has been removed from the v1.2 standard)
- GeoJSON: support for the IETF RFC 7946 revision
- FileGDB/OpenFileGDB: add support to read curve geometries
- ISIS3 (planetary datasets): add write support and improve read support

Other changes

- Upgrade to EPSG database v9.0
- ogrmerge.py: to merge several vector datasets into a single one
- /vsigs/: read support for Google Cloud Storage
- Python bindings: Global Interpreter Lock (GIL) released before entering GDAL native code
- GNM built by default

GDAL 2.3 (likely) preview

- More robustness: GDAL is now integrated in OSS-Fuzz. 400 issues fixed in the last 6 weeks
- PDS4 (Planetary datasets) raster driver
- Multi-core gdal2tiles (ongoing work by Gregory Bataille)
- JPEG2000 driver for Comprimato (propr., GPU accel.)
- GRIB2 write support
- More cloud-based virtual filesystems:
  - extra ways of providing credentials for AWS S3
  - write support for Google Storage,
  - read/write for Microsoft Azure blobs,
  - read/write for AliCloud Object Storage Service

GDAL 2.3 (likely) preview

- Fast lossless TIFF compression/decompression with ZStd
- Capability to write GDAL/OGR drivers in Python
  - Prototype for read-only vector drivers at

    https://github.com/rouault/gdal2/tree/pythondrivers
- C++11 compatible compiler required

Potential future directions

- OpenFileGDB write support
- OpenFileGDB raster read support
- GeoJSON driver compatible of arbitrary large files on reading
- Improvement in spatial reference system management: guessing of EPSG codes, proposing appropriate datum shifts according to location, …
- CRS WKT 2 / ISO 19162 standard management
- Alternative geometry engine : Boost::Geometry
- New drivers, performance improvements, ...

Potential future directions

- CMake build system
  - Unified build systems for Unix&Windows
  - Out-of-tree builds, correct header dependency
  - https://github.com/nextgis-borsch/lib_gdal
- GNM
  - Add more network drivers (pgRouting, OSRM, …)
  - Conversions between network formats (PGRouting, Spatialite, …)
- Planar topology:
  - New abstraction based on related ISO SQL/MM Part3 modeling
  - Topology primitives: nodes, edges, faces
  - TopoGeometry build on primitives / hierarchical TopoGeometry
  - Building of topology from geometries
  - Geometry ←→ TopoGeometry conversions
  - Interface with PostGIS, GRASS, Oracle, GML, Spatialite, TopoJSON
  - Conversion : topo2topo

Questions?

Links:

http://www.gdal.org/

https://trac.osgeo.org/gdal/wiki/Release/2.2.0-News

https://trac.osgeo.org/gdal/wiki/RfcList

Contact: even.rouault@spatialys.com