Open Source Geospatial Analysis in Python: GDAL/OGR Basics

python™

OSGeo
Your Open Source Compass

GDAL

OGR

GDAL/OGR

Originally developed by Frank Warmerdam

Geospatial Data Abstraction Library (GDAL)

- A set of tools for working with raster data, mainly for **translation, projecting and some processing**. The extent of spatial analysis tools is limited, but the building blocks are there.
- GDAL **Utilities** include Projecting, Mosaicing, Polyongize/rasterize, Proximity, among others.

OpenGIS Simple Features Library (OGR)

- A set of tools for **reading and writing** vector formats.
- Large amount of **vector geometry tools**, such as: buffer, intersect, union, convexHull, project, simplify, among others.

These tools were designed for the command line interface. Since then a number of API's have been developed for **Python**, Perl, Ruby, Java, C++, and others. Working with the API is not the native environment , so some functionality is not available or harder to use.

GDAL/OGR Command Line Examples

```
gdalinfo --help
gdalinfo myFile.tif

gdalwarp -t_srs 'EPSG:4326' input.tif output.tif

ogrinfo --help
ogrinfo –so myFile.shp myFile
ogr2ogr -f 'Mapinfo File' output.tab input.shp
```

Command line example

Software that uses GDAL/OGR

Several software programs use the GDAL/OGR libraries to allow them to read and write multiple GIS formats. Such programs include:

ArcGIS – Uses GDAL for reading/writing raster formats.
Biosphere3D – Open source landscape scenery globe.
FWTools – A cross-platform open source GIS software bundle compiled by Frank Warmerdam.
gdaltokmz – A Python module translating from GDAL-supported raster graphics formats to the Google Earth KMZ format.
Google Earth – A virtual globe and world imaging program.
GRASS GIS
gvSIG
JMap
MapServer
World Wind Java – NASA's open source virtual globe and world imaging technology.
OSSIM – Libraries and applications used to process imagery, maps, terrain, and vector data.
OpenEV – Geospatial toolkit and a frontend to that toolkit; to display georeferenced images and elevation data.
Orfeo toolbox – A satellite image processing library.
Quantum GIS
R – An open source statistical software with extensions for spatial data analysis.
SAGA GIS – A cross-platform open source GIS software.
TopoQuest – Internet topographic map viewer.
Rolta Geomatica software

http://en.wikipedia.org/wiki/GDAL

Where to get started:

- http://wiki.osgeo.org/wiki/OSGeo_Python_Library - The OSGeo wiki for Python. Doesn’t contain a lot of updated resources.
- http://trac.osgeo.org/gdal/wiki/GdalOgrInPython - The GDAL/OGR wiki for Python. Useful links and installation information.
- http://www.gdal.org/gdal_tutorial.html - GDAL API tutorial. Has good documentation for a limited number of tasks in C, C++ and Python
- http://www.gis.usu.edu/~chrisg/python/ - Chris Gerard of the RS/GIS Lab at Utah State University has taught a few classes on Open Source RS/GIS and provided these resources to the public. This is a great place to get started with GDAL/OGR. Two of her lectures are on the Z drive in the readings.
- http://pcjericks.github.io/py-gdalogr-cookbook/ - Recently a GDAL/OGR Cookbook has been put together. This is the best resource at the moment, in my opinion.
- http://gdal.org/python/ - This is very useful reference resource once you grasp the basics.
- http://trac.osgeo.org/gdal/wiki/PythonGotchas - Discusses many common pitfalls and known issues.

OGR Vector Formats

Format Name	Code	Creation	Georeferencing	Compiled by default
Aeronav FAA files	AeronavFAA	No	Yes	Yes
ESRI ArcObjects	ArcObjects	No	Yes	No, needs ESRI ArcObjects
Arc/Info Binary Coverage	AVCBin	No	Yes	Yes
Arc/Info .E00 (ASCII) Coverage	AVCE00	No	Yes	Yes
Arc/Info Generate	ARCGEN	No	No	Yes
Atlas BNA	BNA	Yes	No	Yes
AutoCAD DWG	DWG	No	No	No
AutoCAD DXF	DXF	Yes	No	Yes
Comma Separated Value (.csv)	CSV	Yes	No	Yes
CouchDB / GeoCouch	CouchDB	Yes	Yes	No, needs libcurl
DODS/OPeNDAP	DODS	No	Yes	No, needs libdap
EDIGEO	EDIGEO	No	Yes	Yes
ElasticSearch	ElasticSearch	Yes (write-only)	-	No, needs libcurl
ESRI FileGDB	FileGDB	Yes	Yes	No, needs FileGDB API library
ESRI Personal GeoDatabase	PGeo	No	Yes	No, needs ODBC library
ESRI ArcSDE	SDE	No	Yes	No, needs ESRI SDE
ESRI Shapefile	ESRI Shapefile	Yes	Yes	Yes
FMEObjects Gateway	FMEObjects Gateway	No	Yes	No, needs FME

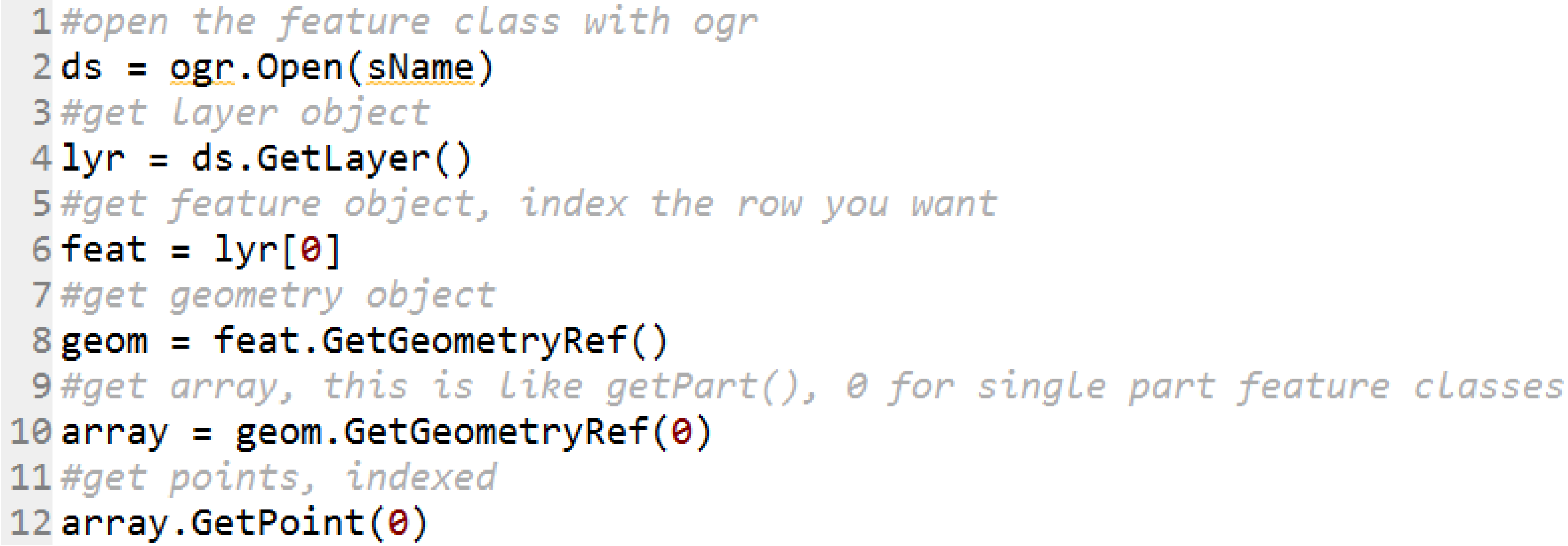
Format Name	Code	Creation	Georeferencing	Compiled by default
Géoconcept Export	Geoconcept	Yes	Yes	Yes
Geomedia .mdb	Geomedia	No	No	No, needs ODBC library
GeoRSS	GeoRSS	Yes	Yes	Yes (read support needs libexpat)
Google Fusion Tables	GFT	Yes	Yes	No, needs libcurl
GML	GML	Yes	Yes	Yes (read support needs Xerces or libexpat)
GMT	GMT	Yes	Yes	Yes
GPSBabel	GPSBabel	Yes	Yes	Yes (needs GPSBabel and GPX driver)
GPX	GPX	Yes	Yes	Yes (read support needs libexpat)
GRASS	GRASS	No	Yes	No, needs libgrass
GPSTrackMaker (.gtm, .gtz)	GPSTrackMaker	Yes	Yes	Yes
Hydrographic Transfer Format	HTF	No	Yes	Yes
Idrisi Vector (.VCT)	Idrisi	No	Yes	Yes
Informix DataBlade	IDB	Yes	Yes	No, needs Informix DataBlade
INTERLIS	"Interlis 1" and "Interlis 2"	Yes	Yes	No, needs Xerces (INTERLIS model reading needs ili2c.jar)
INGRES	INGRES	Yes	No	No, needs INGRESS
KML	KML	Yes	Yes	Yes (read support needs libexpat)

The list goes on.....

OGR Classes

GDAL/OGR are object oriented libraries. You have to work with multiple objects to complete a task, just like with arcpy.

Example:



```
#open the feature class with ogr
ds = ogr.Open(sName)
#get layer object
lyr = ds.GetLayer()
#get feature object, index the row you want
feat = lyr[0]
#get geometry object
geom = feat.GetGeometryRef()
#get array, this is like getPart(), 0 for single part feature classes
array = geom.GetGeometryRef(0)
#get points, indexed
array.GetPoint(0)
```

This accesses the hierarchy of vector geometry encoding.

OGR examples

OGR Geometry Types

Well known binary (wkb) geometry types

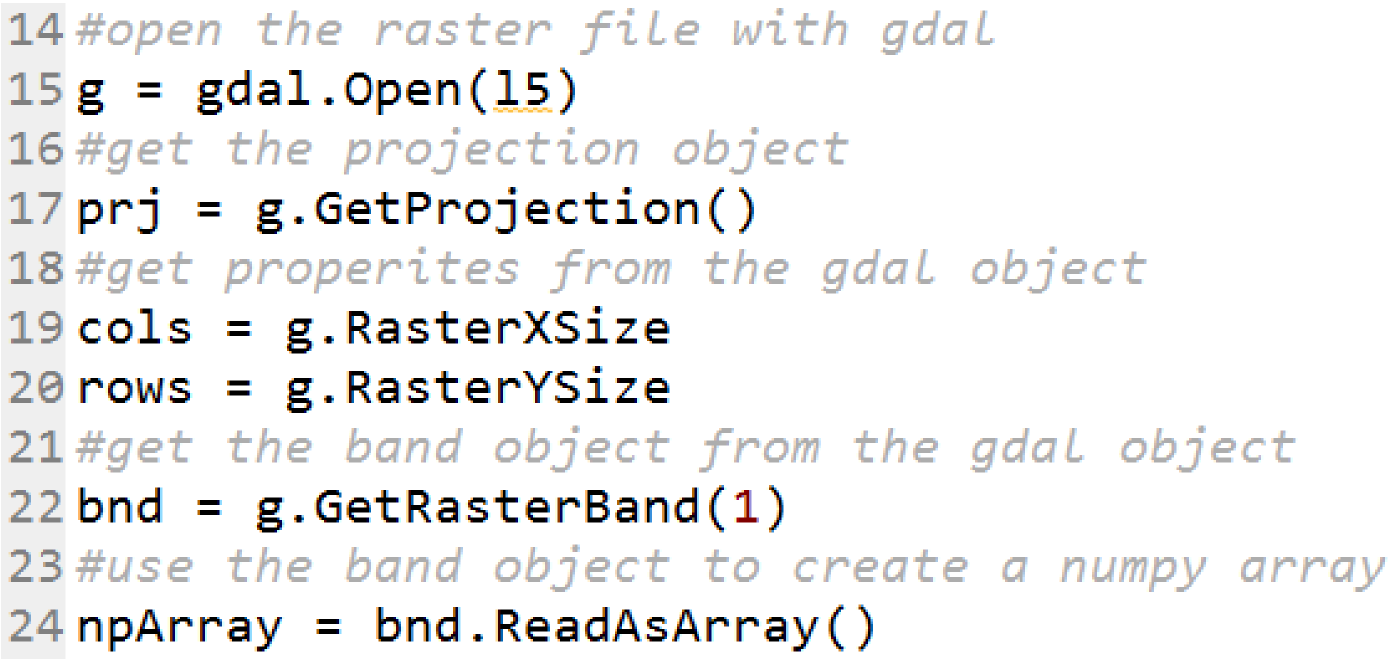
wkbUnknown	unknown type, non-standard
wkbPoint	0-dimensional geometric object, standard WKB
wkbLineString	1-dimensional geometric object with linear interpolation between Points, standard WKB
wkbPolygon	planar 2-dimensional geometric object defined by 1 exterior boundary and 0 or more interior boundaries, standard WKB
wkbMultiPoint	GeometryCollection of Points, standard WKB
wkbMultiLineString	GeometryCollection of LineStrings, standard WKB
wkbMultiPolygon	GeometryCollection of Polygons, standard WKB
wkbGeometryCollection	geometric object that is a collection of 1 or more geometric objects, standard WKB
wkbNone	non-standard, for pure attribute records
wkbLinearRing	non-standard, just for createGeometry()
wkbPoint25D	2.5D extension as per 99-402
wkbLineString25D	2.5D extension as per 99-402
wkbPolygon25D	2.5D extension as per 99-402
wkbMultiPoint25D	2.5D extension as per 99-402
wkbMultiLineString25D	2.5D extension as per 99-402
wkbMultiPolygon25D	2.5D extension as per 99-402
wkbGeometryCollection25D	2.5D extension as per 99-402

GDAL Raster Formats

Raster data format name	Code
Arc/Info ASCII Grid [1]	AAIGrid
ADRG/ARC Digitalized Raster Graphics (.gen/.thf) [2]	ADRG
Magellan BLX Topo (.blx, .xlb) [3]	BLX
Microsoft Windows Device Independent Bitmap (.bmp) [4]	BMP
VTP Binary Terrain Format (.bt) [5]	BT
Military Elevation Data (.dt0, .dt1, .dt2) [6]	DTED
ESRI .hdr Labelled [7]	EHdr
NASA ELAS [8]	ELAS
ENVI .hdr Labelled Raster [9]	ENVI
ERMapper (.ers) [10]	ERS
GeoTiff	GTiff
NOAA .gtx vertical datum shift	GTX
HF2/HFZ heightfield raster [11]	HF2
Erdas Imagine (.img) [12]	HFA
Image Display and Analysis (WinDisp) [13]	IDA
ILWIS Raster Map (.mpr,.mpl) [14]	ILWIS
Intergraph Raster [15]	INGR
USGS Astrogeology Isis cube (Version 2) [16]	ISIS2
KMLSUPEROVERLAY	KMLSUPEROVERLAY
In Memory Raster [17]	MEM
Vexcel MFF [18]	MFF
Vexcel MFF2 [19]	MFF2 (HKV)
NITF [20]	NITF
NTv2 Datum Grid Shift	NTv2
PCI Geomatics Database File [21]	PCIDSK
Raster Matrix Format (*.rsw, .mtw) [22]	RMF
Idrisi Raster [23]	RST
SAGA GIS Binary format [24]	SAGA
SGI Image Format [25]	SGI
SRTM HGT Format [26]	SRTMHGT
USGS ASCII DEM / CDED (.dem) [27]	USGSDEM
GDAL Virtual (.vrt) [28]	VRT
ASCII Gridded XYZ [29]	XYZ

GDAL Classes

GDAL allows for multiple bands. Excellent interface with numpy is provided.
Example:



```
#open the raster file with gdal
g = gdal.Open(15)
#get the projection object
prj = g.GetProjection()
#get properites from the gdal object
cols = g.RasterXSize
rows = g.RasterYSize
#get the band object from the gdal object
bnd = g.GetRasterBand(1)
#use the band object to create a numpy array
npArray = bnd.ReadAsArray()
```

Should look familiar, working with arcpy raster objects is similar

GDAL examples

GDAL Data Types (GDT)

GDT Data Type and Reference #
GDT_Unknown = 0,
GDT_Byte = 1,
GDT_UInt16 = 2,
GDT_Int16 = 3,
GDT_UInt32 = 4,
GDT_Int32 = 5,
GDT_Float32 = 6,
GDT_Float64 = 7,
GDT_CInt16 = 8,
GDT_CInt32 = 9,
GDT_CFloat32 = 10,
GDT_CFloat64 = 11,
GDT_TypeCount = 12

http://www.gdal.org/gdal_8h.html