Command Prompt

FOSS4GNL 2026

WORKSHOP

GDAL COMMAND LINE INTERFACE

Hans van der Kwast
QWAST-GIS
hans@qwast-gis.com

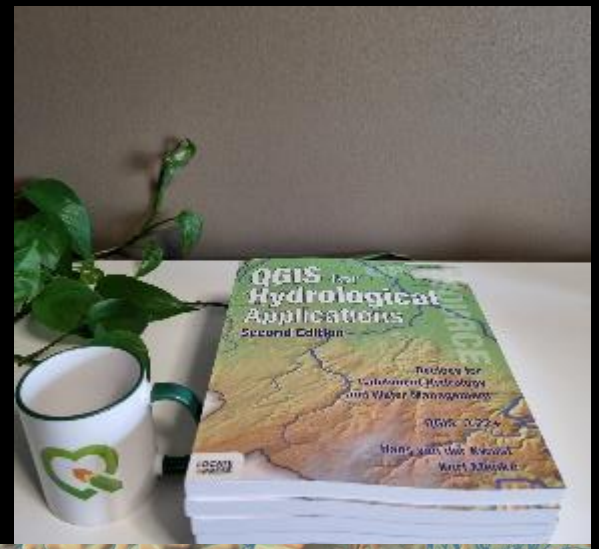
Command Prompt

IHE DELFT Institute for Water Education under the auspices of UNESCO

QWAST

ABOUT ME

- Physical Geographer, Associate Professor of Open Science and Digital Innovation at IHE Delft
- MSc, PhD at Utrecht University, the Netherlands
- Researcher at the Flemish Institute for Technological Research (VITO, Belgium)
- Board member of Dutch QGIS User Group, OSGeo Charter member
- Owner of QWAST-GIS and member of QCooperative
- Co-author of QGIS for Hydrological Applications with Kurt Menke (Septima)
- GIS OpenCourseWare



LinkedIn @hansakwast @hansvanderkwast YouTube @hansvanderkwast

Command Prompt

WHAT IS GDAL?

- Geospatial Data Abstraction Library
- GDAL is a translator library for raster and vector geospatial data formats
- GDAL (pronounced ‘gee-dal’ or ‘goodle’) is a collection of software that helps with the translation of data from different file formats, data types, and map projections. It’s used in free software like QGIS and GRASS, and even some proprietary applications

Command Prompt

WHY A NEW GDAL CLI?

- Powerful and comprehensive
- 20+ years of backwards compatibility
- But many inconsistencies:
  - Underscore or not? gdal_translate vs gdalwarp
  - Dataset order: gdal_translate in.tif out.tif vs ogr2ogr out.gpkg in.gpkg
  - Argument naming:gdalwarp -ts width height vs gdal_translate -outsize width height ○Dash vs double dash: gdal_calc --type vs gdal_translate -ot
  - Lots (too many) options ⇒ ogr2ogr: 76 switches!

Command Prompt

WHAT'S BEHIND THE NEW CLI?

- git -style sub-commands
- Break apart huge CLIs
  - gdal_translate and ogr2ogr
  - gdalwarp and gdaldem
- Unified CLI framework
- Programmatic discoverability
- GDALG pipelining

Command Prompt

WHAT'S NEW?

- Short options ⇒ single dash:
  -i my.tif
- Long options ⇒ double dash:
  --update
- Two possibilities to pass values:
  --input=my.tif or --input my.tif
- Input(s) first, output last:
  $ gdal raster convert in.gpkg out.tif
- Always use hyphen (no more _'s)
- Never overwrite output without -overwrite

- Detection of typos such as letter inversion

```
$ gdal raster convret ERROR 1: Algorithm 'convret' is unknown. Do you mean 'convert'?
```

```
$ gdal raster convert --creattionoption ERROR 5: convert: Option '--creattionoption' is unknown. Do you mean '--creation-option'?
```

Command Prompt

FIRST LEVEL COMMANDS

```
D:\>gdal
ERROR 1: gdal: Missing command name.
Usage: gdal <COMMAND> [OPTIONS]
where <COMMAND> is one of:
  - convert:  Convert a dataset (shortcut for 'gdal raster convert' or 'gdal vector convert').
  - dataset:  Commands to manage datasets.
  - driver:   Command for driver specific operations.
  - info:     Return information on a dataset (shortcut for 'gdal raster info' or 'gdal vector
info').
  - mdim:     Multidimensional commands.
  - pipeline: Process a dataset applying several steps.
  - raster:   Raster commands.
  - vector:   Vector commands.
  - vsi:      GDAL Virtual System Interface (VSI) commands.

Try 'gdal --help' for help.

'gdal <FILENAME>' can also be used as a shortcut for 'gdal info <FILENAME>'.
And 'gdal read <FILENAME> ! ...' as a shortcut for 'gdal pipeline <FILENAME> ! ...'.

For more details, consult https://gdal.org/programs/index.html

WARNING: the gdal command is provisionally provided as an alternative interface to GDAL and OGR
command line utilities.
The project reserves the right to modify, rename, reorganize, and change the behavior of the utility
until it is officially frozen in a future feature release of GDAL.
```

Command Prompt

GDAL RASTER

```
- aspect:        Generate an aspect map  (gdaldem aspect)
- calc:          Perform raster algebra
- clean-collar:  Clean the collar, removing noise. (nearblack)
- clip:          Clip a raster dataset   (gdal_translate -projwin)
- color-map:     Generate a RGB or RGBA dataset from a single band, using
                 a color map (gdaldem color-relief or gdal_translate -expand rgb)
- contour:       Creates a vector contour from a raster elevation model (DEM)
- convert:       Convert a raster dataset (gdal_translate)
- create:        Create a new raster dataset (gdal_create)
- edit:          Edit a raster dataset (gdal_translate -mo/-a_nodata/-a_srs)
- fill-nodata:   Fill nodata raster regions by interpolation from edges.
                 (gdal_fillnodata.py)
- footprint:     Compute the footprint of a raster dataset (gdal_footprint)
- hillshade:     Generate a shaded relief map (gdaldem hillshade)
- index:         Create a vector index of raster datasets (gdaltindex)
- info:          Return information on a raster dataset (gdalinfo)
- mosaic:        Build a mosaic, virtual (VRT) or materialized. (gdalbuildvrt)
- overview:      Manage overviews of a raster dataset (gdaladdo)
- pipeline:      Process a raster dataset.
- pixel-info:    Return information on a pixel (gdallocationinfo)
- polygonize:    Create a polygon feature dataset from a band (gdal_polygonize)
- reclassify:    Reclassify values in a raster dataset
- reproject:     Reproject a raster dataset (gdalwarp)
```

Command Prompt

GDAL RASTER (CONTINUED)

- resize: Resize a raster dataset without changing the georeferenced extents *(gdal_translate -outsize)*
- roughness: Generate a roughness map *(gdaldem roughness)*
- scale: Scale the values of the bands *(gdal_translate -scale)*
- select: Select a subset of bands *(gdal_translate -b)*
- set-type: Modify the data type of bands *(gdal_translate -ot)*
- sieve: Remove small polygons *(gdal_sieve.py)*
- slope: Generate a slope map *(gdaldem slope)*
- stack: Combine together input bands into a multi-band output, virtual (VRT) or materialized *(gdalbuildvrt -separate)*
- **tile**: Generate tiles in separate files from a raster dataset. **(enhanced gdal2tiles)**
- tpi: Generate a Topographic Position Index (TPI) map *(gdaldem TPI)*
- tri: Generate a Terrain Ruggedness Index (TRI) map *(gdaldem TRI)*
- unscale: Convert scaled values of a raster dataset into unscaled Values. *(gdal_translate -unscale)*
- viewshed: Compute the viewshed of a raster dataset *(gdal_viewshed)*

GDAL RASTER TILE

- C++ port of gdal2tiles.py
- Enhancements
  - Handle non-Byte data types (UInt16, Float32…)
  - More tiling schemes
    - consecutive overview levels with resolution != 2 (ie NZTM2000)
    - alternative origin
    - different tile size
  - 3x to 6x faster
  - Unified progress report

Command Prompt

GDAL VECTOR

```
- clip:      Clip a vector dataset (ogr2ogr -clipsrc / -clipdst)
- concat:    Concatenate vector datasets (ogr_merge.py)
- convert:   Convert a vector dataset (ogr2ogr)
- edit:      Edit metadata of a vector dataset (ogr2ogr -a_srs/-mo)
- filter:    Filter a vector dataset (ogr2ogr -spat/-where)
- geom:      Geometry operations on a vector dataset (ogr2ogr …)
- grid:      Create a regular grid from scattered points (gdal_grid)
- info:      Return information on a vector dataset (ogrinfo)
- pipeline:  Process a vector dataset
- rasterize: Burns vector geometries into a raster (gdal_rasterize)
- reproject: Reproject a vector dataset (ogr2ogr -t_srs)
- select:    Select a subset of fields   (ogr2ogr -select)
- sql:       Apply SQL statement(s) to a dataset (ogr2ogr -sql)
```

Command Prompt

GDALG - GENERATING PIPELINES

- Some raster or vector algorithms accept streamed input and generate streamed outputs ⇒ processing pipelines

```
$gdal raster pipeline read in.tif ! \
reproject --dst-crs=EPSG:4326 ! \
edit --metadata TITLE=my-title ! \
write out.tif --of COG
```

```
$ gdal vector pipeline ! \
read in.gpkg ! \ select fields=name,geometry ! \
reproject --dst-crs=EPSG:4326 ! \
geom make-valid ! \
clip --box=2,49,3,50 ! \
write out.gpkg --overwrite
```

Command Prompt

ACKNOWLEDGEMENTS

Thanks to Even Rouault (SPATIALYS)