

GeoPackage Hackathon

BEV, 06.05.2019

GeoPackage in GDAL

GeoPackage in GDAL

Allgemeines zu GDAL

- GDAL - Geospatial Data Abstraction Library
 - freie Programmbibliothek für die Übersetzung räumlicher Rasterdaten
- OGR Simple Features Library (OGR)
 - freie Programmbibliothek für die Übersetzung räumlicher Vektordaten
- seit GDAL 2.0 sind Raster und Vektor nicht mehr so strikt getrennt
- Kernfunktionen
 - kommandozeilen-basierte Hilfsprogramme zur Datenumwandlung, Bildanpassung, Generierung von Bildausschnitten und für zahlreiche andere gängige Bildbearbeitungsaufgaben
 - Hocheffizienter Zugriff auf Rasterdaten unter Nutzung der Vorteile von Kachelung und Übersichten
 - Unterstützung großer Dateien - größer 4GB
 - Zugriff auf die Bibliothek über Python, Java, C#, Ruby, VB6 und Perl
 - Die Umrechnung in andere Koordinatenreferenzsysteme baut auf die PROJ.4 und OGC Well Known Text Beschreibung auf
- <https://www.gdal.org/>

GeoPackage in GDAL

Vorbereitung

- Download GDAL Binaries
 - <https://trac.osgeo.org/gdal/wiki/DownloadingGdalBinaries>
 - z.B. MS4W, OSGeo4W oder GISInternals
- Download GDAL Source
 - <https://trac.osgeo.org/gdal/wiki/DownloadSource>
 - <https://github.com/OSGeo/gdal>
 - muss selbst gebildet werden
- Für heute Verwendung von `\Software\release-1900-gdal-2-4-0-mapserver-7-2-2\`
 - ausführen von `SDKShell hideoci`
 - `gdalinfo --version`

GeoPackage in GDAL

Verwendung von ogr2ogr

Usage: ogr2ogr [--help-general] [-skipfailures] [-append] [-update]
[-select field_list] [-where restricted_where|\@filename]
[-progress] [-sql <sql statement>|\@filename] [-dialect dialect]
[-preserve_fid] [-fid FID] [-limit nb_features]
[-spat xmin ymin xmax ymax] [-spat_srs srs_def] [-geomfield field]
[-a_srs srs_def] [-t_srs srs_def] [-s_srs srs_def] [-ct string]
[-f format_name] [-overwrite] [[-dsco NAME=VALUE] ...]
dst_datasource_name src_datasource_name
[-lco NAME=VALUE] [-nln name]
[-nlt type|PROMOTE_TO_MULTI|CONVERT_TO_LINEAR|CONVERT_TO_CURVE]
[-dim XY|XYZ|XYM|XYZM|2|3|layer_dim] [layer [layer ...]]

siehe <https://www.gdal.org/ogr2ogr.html>

GeoPackage in GDAL

ogr2ogr Grundlagen

- `ogrinfo --formats`
- Kommandoaufruf: `ogr2ogr -f "Ausgabeformat" -Optionen ZIEL QUELLE`
 - http://www.gdal.org/ogr_formats.html
 - https://www.gdal.org/drv_geopackage.html
 - Optionen:
 - `-lco LayerCreation Options`
 - `-dco DatabaseCreation Options`
 - `-lco ENCODING=UTF-8` → Zeichencodierung setzen
 - `-lco SPATIAL_INDEX=YES` → Räumlichen Index erzeugen
 - `-s_srs Source-Koordinatensystem` → Definition durch EPSG-Code oder proj-String
 - `-t_srs Target-Koordinatensystem` → Definition durch EPSG-Code oder proj-String
- `gdal_translate` für Rasterdatenkonvertierung
 - https://www.gdal.org/gdal_translate.html

GeoPackage in GDAL

ogr2ogr Beispiele

▪ SHP to ORACLE und vice versa

- `ogr2ogr -lco INDEX=OFF -lco SRID=31287 -lco LAUNDER=YES -lco DIM=2 -nln "TestTableName" -f "OCI" OCI:ORAUSER/ORAPW@ORADBDNAME "C:\Pfad\zu\Eingangsdatensatz.shp" -progress`
- `ogr2ogr -f "ESRI Shapefile" -lco SPATIAL_INDEX=YES C:\Pfad\zu\Outputdatensatz.shp "OCI:ORAUSER/ORAPW@ORADBDNAME:TestName" -progress`

▪ GeoPackage to PostgreSQL und vice versa

- `ogr2ogr -f "PostgreSQL" PG:"host=111.222.333.444 port=5432 user=testuser password=testpasswort dbname=testdb" C:\Pfad\zu\GeoPackage.gpkg "TestTableName" -t_srs EPSG:31287 -nlt "POLYGON" -nln "TestTableName"`
- `ogr2ogr -f "GPKG" -update -append -nln "TestTableName" C:\Pfad\zu\GeoPackage.gpkg PG:"host=111.222.333.444 port=5432 user=testuser password=testpasswort dbname=testdb" "TestTableName"`

GeoPackage in GDAL

GeoPackage LayerCreationOptions

- Layer Creation Options
 - GEOMETRY_NAME: Column to use for the geometry column. Default to "geom". Note: option was called GEOMETRY_COLUMN in releases before GDAL 2
 - GEOMETRY_NULLABLE: (GDAL >=2.0) Whether the values of the geometry column can be NULL. Can be set to NO so that geometry is required. Default to "YES"
 - FID: Column name to use for the OGR FID (primary key in the SQLite database). Default to "fid"
 - OVERWRITE: If set to "YES" will delete any existing layers that have the same name as the layer being created. Default to NO
 - SPATIAL_INDEX: (GDAL >=2.0) If set to "YES" will create a spatial index for this layer. Default to YES
 - PRECISION: (GDAL >=2.0) This may be "YES" to force new fields created on this layer to try and represent the width of text fields (in terms of UTF-8 characters, not bytes), if available using TEXT(width) types. If "NO" then the type TEXT will be used instead. The default is "YES".
 - TRUNCATE_FIELDS: (GDAL >=2.0) This may be "YES" to force truncated of field values that exceed the maximum allowed width of text fields, and also to "fix" the passed string if needed to make it a valid UTF-8 string. If "NO" then the value is not truncated nor modified. The default is "NO".
 - IDENTIFIER=string: (GDAL >=2.0) Identifier of the layer, as put in the contents table.
 - DESCRIPTION=string: (GDAL >=2.0) Description of the layer, as put in the contents table.
 - ASPATIAL_VARIANT=GPKG_ATTRIBUTES/OGR_ASPATIAL/NOT_REGISTERED: (GDAL >=2.2) How to register non spatial tables. Defaults to GPKG_ATTRIBUTES in GDAL 2.2 or later (behaviour in previous version was equivalent to OGR_ASPATIAL). Starting with GeoPackage 1.2, non spatial tables are part of the specification. They are recorded with data_type="attributes" in the gpkg_contents table. This is only compatible of GDAL 2.2 or later. Priorly, in OGR 2.0 and 2.1, the "aspatial" extension had been developed for similar purposes, so if selecting OGR_ASPATIAL, non spatial tables will be recorded with data_type="aspatial" and the "aspatial" extension was declared in the gpkg_extensions table. It is also possible to use the NOT_REGISTERED option, in which case the non spatial table is not registered at all in any GeoPackage system tables.

GeoPackage in GDAL

ogr2ogr eBOD Testdaten: Konvertierung

- Wechsel ins Work-Verzeichnis mittels „pushd“
- Ausgangsdaten GeoJSON → Convert to GeoPackage
 - `ogr2ogr -f "GPKG" -lco SPATIAL_INDEX=YES -update -append -s_srs EPSG:4326 -t_srs EPSG:31287 -nln "bodenart" C:\Pfad\zu\ebod_test.gpkg ebod_bodenart_test.geojson -progress`
 - `ogr2ogr -f "GPKG" -lco SPATIAL_INDEX=YES -update -append -s_srs EPSG:4326 -t_srs EPSG:31287 -nln "bodentyp" ebod_test.gpkg ebod_bodentyp_test.geojson -progress`
- Select mittels SQL
 - `ogr2ogr -f "GPKG" -lco SPATIAL_INDEX=YES -update -append -s_srs EPSG:4326 -t_srs EPSG:31287 -nln "bodenart_lehm" ebod_test.gpkg ebod_bodenart_test.geojson -sql "SELECT * FROM ebod_bodenart_test WHERE code = 'L'" -progress -skipfailures`

GeoPackage in GDAL

Tipps und Tricks

- Arbeiten mit ORACLE
 - SET TNS_ADMIN=c:\Pfad\zu\tnsnames.ora
 - setzt den Pfad zu der ORACLE-Config Datei, falls nicht automatisch gesetzt
 - SET NLS_LANG=GERMAN_GERMANY.UTF8
 - setzt Zeichenkodierung in der Shell auf korrektes UTF8, falls nicht automatisch gesetzt
 - -lco SRID=31287
 - setzen des CRS explizit für ORACLE

GeoPackage in GDAL

Skripting in GDAL

- Skripting mit Batch- bzw. Shell Scripts

- ```
for %%i in (*.geojson) do ogr2ogr -f "GPKG" [...] -nln "%~ni"
C:\Pfad\zu\ebod_test.gpkg %%i -progress
```

- Verwendung von GDAL in Python

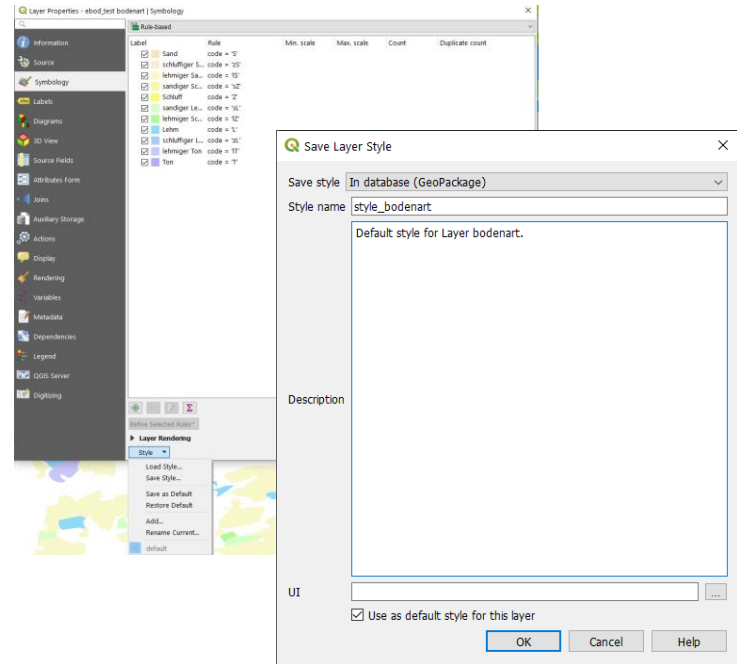
- direkter Import in Python (`from osgeo import ogr, osr, gdal`)
- Anaconda (<https://www.anaconda.com>)
  - einfach Verwendung auch mit Jupyter Noteboo

# Styles in GeoPackage

# Styles in GeoPackage

## eBOD Testdaten: Stile

- Bodenart in QGIS öffnen
- Laden des SLD Files
- Speichern als Default Stil
- Erstellen von Symbology für Bodentyp und Speichern als Default-Stil



# Styles in GeoPackage

## VGD Testdaten: Stile

- Öffnen von austria Daten und eBOD in DB Browser oder DBeaver
- Speicherung der QGIS-Projekt-Styles in GeoPackage
  
- Stile dzt. nur in Verbindung mit QGIS automatisch ausgelesen
- GeoServer kann SLD Styles lesen wenn man sie hineinkopiert
  - Evtl. Erweiterung für direktes auslesen in Sicht
- ArcGIS → ?

# GeoPackage in GeoServer

# GeoPackage in GeoServer

## GeoServer Grundlagen

- seit 2.15 (?) als Input-Standard in jeder GeoServer Installation
- GeoPackage Extension (<https://docs.geoserver.org/latest/en/user/community/geopkg/index.html>)
  - GeoPackage WMS Output Format
    - format=geopackage
    - tileset\_name: name to be used for tileset in geopackage file (default is name of layer(s)).
    - min\_zoom, max\_zoom, min\_column, max\_column, min\_row, max\_row: set the minimum and maximum zoom level, column, and rows
    - gridset: name of gridset to use (otherwise default for CRS is used)
  - GeoPackage in WFS Output Format
    - format=geopackage
    - GeoPackage mit einem single features layer.
  - GeoPackage WPS Process



# GeoPackage in GeoServer

## Einrichtung von GeoPackage als Datenquelle

1. Öffnen cmd
2. Wechseln ins Verzeichnis von [GeoServer]\bin (mit pushd)
3. Start des GeoServer mittels startup.bat
4. Öffnen eines Browser und Eingabe der URL <http://127.0.0.1:8080/geoserver/>
5. Login mittels admin/geoserver
6. Anlegen eines neuen Workspaces
7. optional: Kopieren der Daten nach [GeoServer]\data\vgd\_test\
8. Einrichten eines Datenspeichers
9. Publizierung der Layer BL, PB, PG und KG
10. Erstellen der Stile

# GeoPackage in GeoServer

## GeoPackage als Datenquelle in LFRZ INSPIRE Services

- Testinstanz <https://portal.lfrz.at/at.lfrz.geoinspire/999901/web/>
- Umbenennen Input-File in geoinspire\_vector.gpkg
- Upload im Datamanager im Tab GeoPackage
- Layer hinzufügen und Metadaten ergänzen

# GeoPackage in Office Anwendungen

# GeoPackage in Office Anwendungen

## Einrichten einer ODBC Verbindung unter Windows

- Download SQLite ODBC Treiber (32bit & 64 bit)
  - <http://www.ch-werner.de/sqliteodbc/>
- Installation beider ODBC Treiber
  - Install SQLite2 nicht anhaken
- Spatialite DLLs downloaden und entpacken
  - <http://www.gaia-gis.it/gaia-sins/windows-bin-x86/>
  - mod\_spatialite-x.x.x-win-x86.7z
- DLLs kopieren in C:\Windows\SysWOW64 (oder alternativ in C:\Windows\System32)
- Neue ODBC Verbindung anlegen
  - Windows 10: Suche "ODBC Datenquellen einrichten"
    - Benutzer-DSN oder System DSN
    - Hinzufügen
    - SQLite3 ODBC Driver
    - Data Source Name: beliebigen sprechenden Namen eingeben
    - Database Name: Link zur .gpkg Datei
    - Load Extension: mod\_spatialite.dll

# GeoPackage in Office Anwendungen

## GeoPackage in Access

1. Reiter "externe Daten"
2. Importieren Sie [...] oder Erstellen Sie eine Verknüpfung [...]
3. Computerdateiquellen
4. (erstellte ODBC Verbindung auswählen)
5. gewünschte Tabellen importieren

Herzlichen Dank  
für Ihre Aufmerksamkeit.

**Manuel Illmeyer**

Experte GIS / Projektleitung

[manuel.illmeyer@lfrz.gv.at](mailto:manuel.illmeyer@lfrz.gv.at)

+43 1 / 33176 - 414