



WhereGroup

Wie kommt der Schwimmbagger ins WebGIS?



WhereGroup





WhereGroup





Where





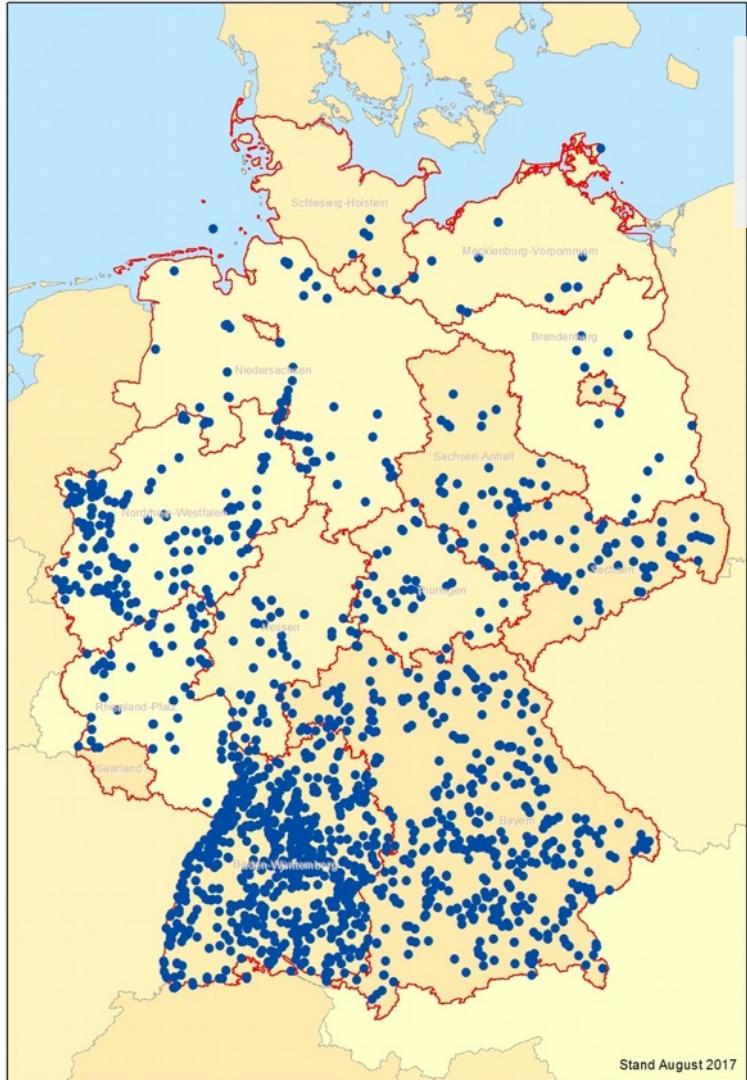
WhereGroup

GisInfoService

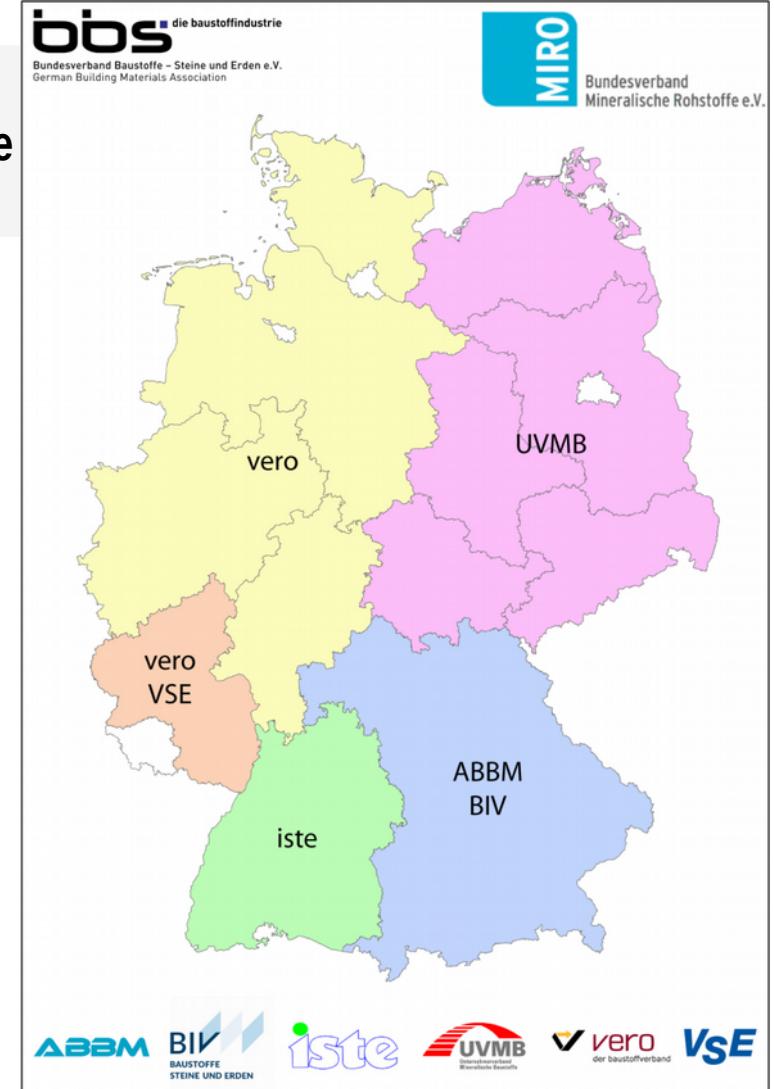
- Online-Geoinformationssystem der Landesrohstoffverbände für die Mitgliedsunternehmen
- Von überall per Internet Browser nutzbar
- Geodaten der Behörden per Webdienst
- Eigene Geodaten können ergänzt werden (z.B. Risswerke, Abbauplanung, Rekultivierungspläne)
- Grundstücksverwaltung ist möglich (Eigentümer, Pächter, Verträge, Fristen etc.)



WhereGroup

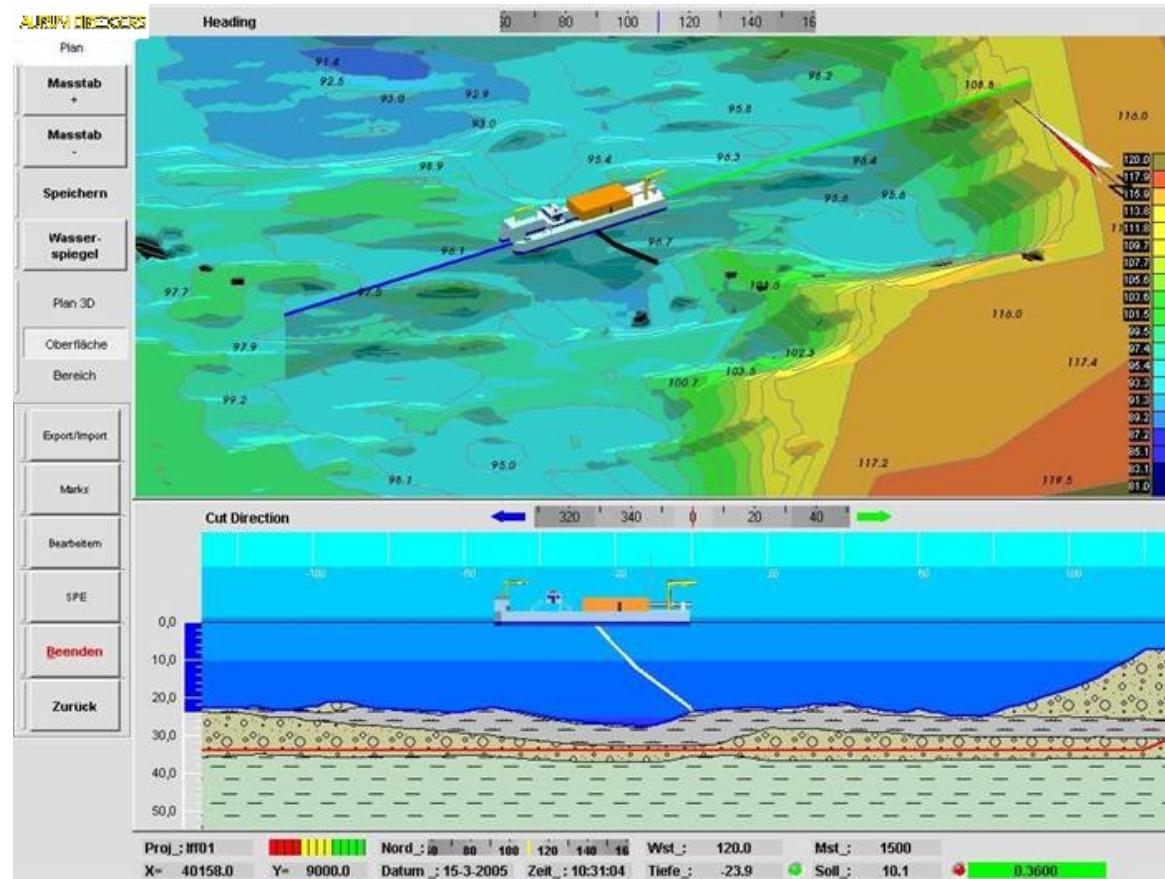


Entwicklungsstand
und registrierte Werke
~1750





WhereGroup

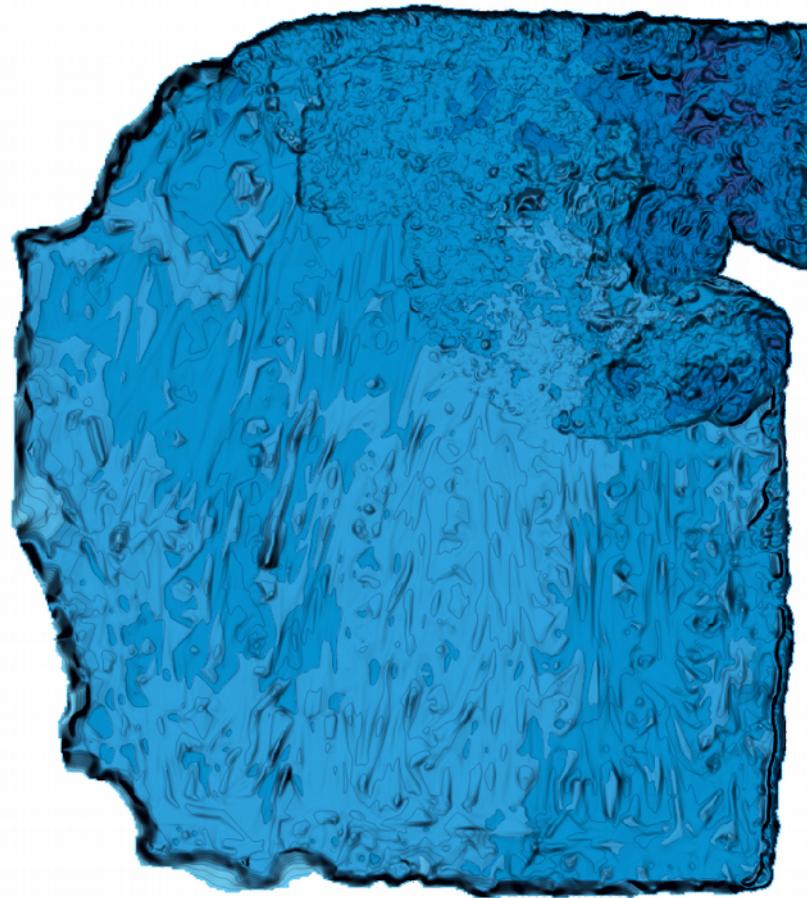




- **regelmäßig / wöchentlich neue Daten**
- **automatischer Workflow zur Verarbeitung**
- **bisher zur Integration in GisInfoService**



WhereGroup





WhereGroup



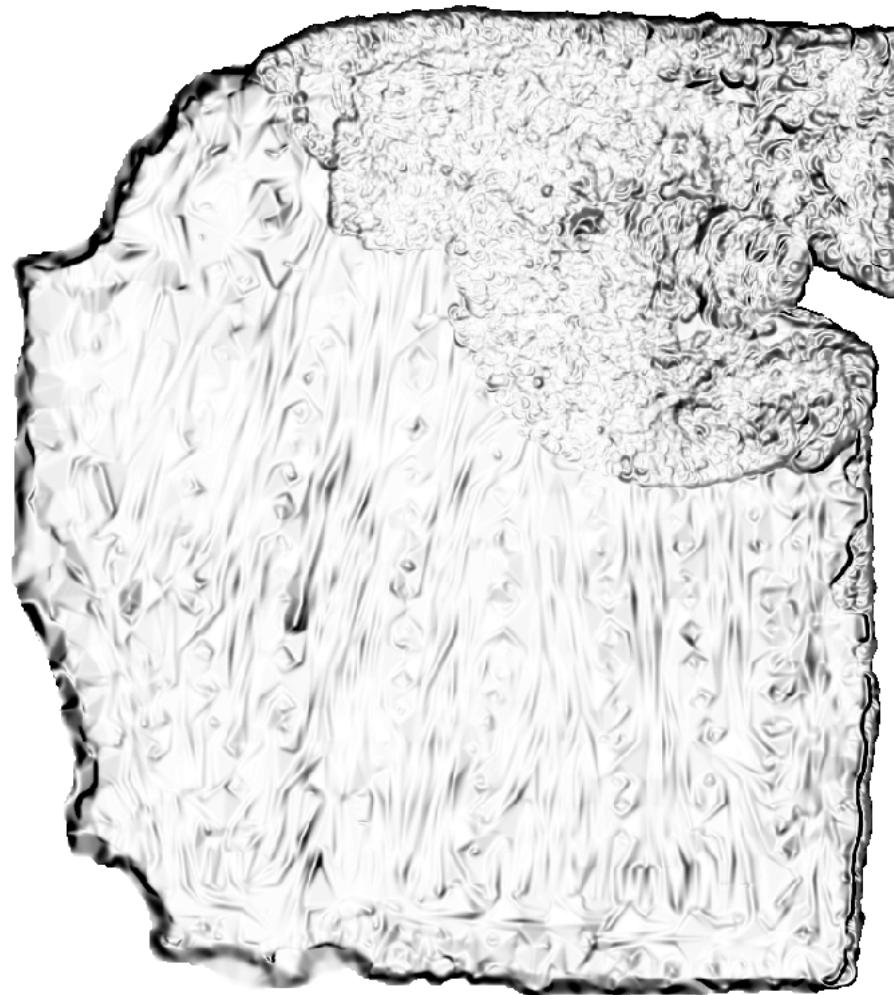


WhereGroup





WhereGroup



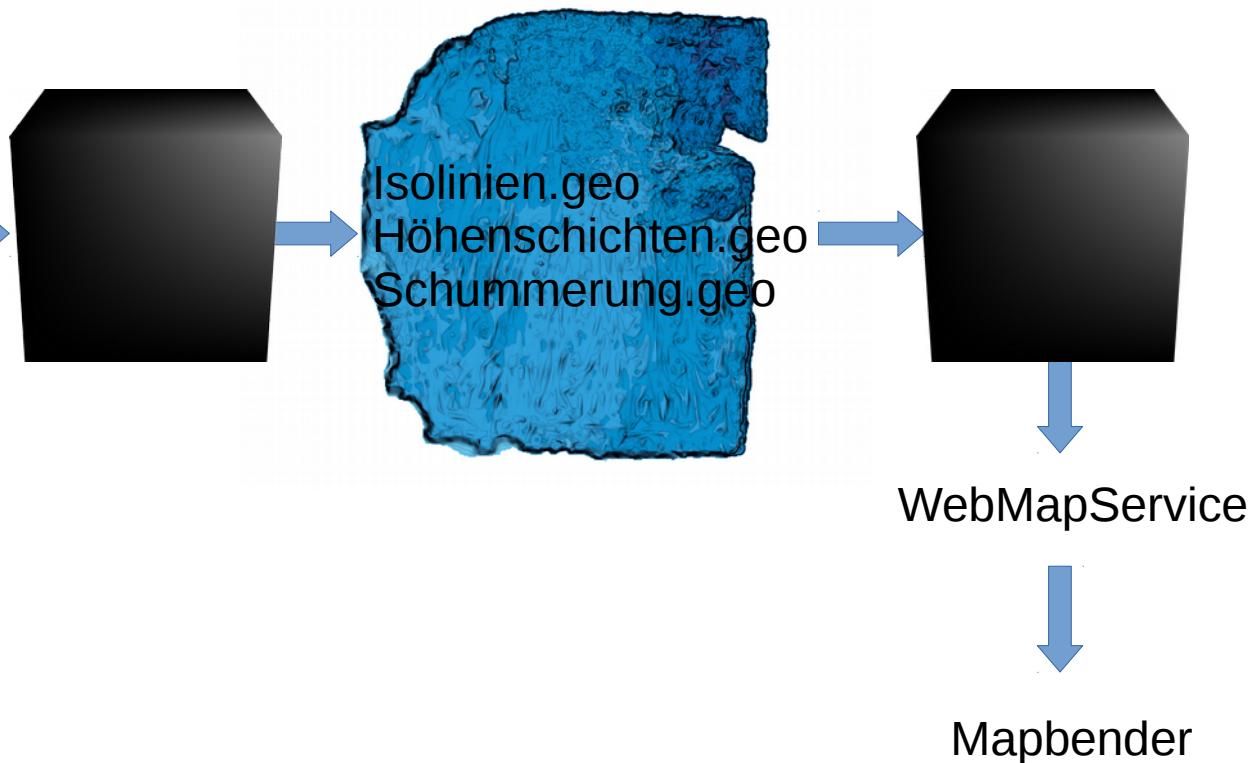


4489708.65	5752716.37	50.40
4489709.65	5752716.37	50.34
4489710.65	5752716.37	50.32
4489711.65	5752716.37	50.33
4489712.65	5752716.37	50.34
4489713.65	5752716.37	50.33
4489714.65	5752716.37	50.37
4489716.65	5752716.37	50.47
4489717.65	5752716.37	50.47
4489718.65	5752716.37	50.49
4489719.65	5752716.37	50.53
4489720.65	5752716.37	50.57
4489721.65	5752716.37	50.57
4489722.65	5752716.37	50.56
4489723.65	5752716.37	50.59



WhereGroup

4489708.65	5752716.37	50.40
4489709.65	5752716.37	50.34
4489710.65	5752716.37	50.32
4489711.65	5752716.37	50.33
4489712.65	5752716.37	50.34
4489713.65	5752716.37	50.33
4489714.65	5752716.37	50.37
4489716.65	5752716.37	50.47
4489717.65	5752716.37	50.47
4489718.65	5752716.37	50.49
4489719.65	5752716.37	50.53
4489720.65	5752716.37	50.57
4489721.65	5752716.37	50.57
4489722.65	5752716.37	50.56
4489723.65	5752716.37	50.59





WhereGroup

```
Werde jeweils in der zu bearbeitenden Ordner
for i in ${FIND_DIRECTORIES}
do
cd $i
echo "*****Bearbeite Verzeichnis: "$i*****
echo "*****Bearbeite Verzeichnis: "$i***** >> /home/hoehendaten/GIS_data/DEM/error.txt

#Erzeuge für jede .txt-Datei in dem Ordner Schummerung, Hoehenschichten und Hoehenlinien
for f in *.txt;
do
echo "*****Bearbeite Datei: "${f##*/}*****
echo "*****Bearbeite Datei: "${f##*/}***** >> /home/hoehendaten/GIS_data/DEM/error.txt
#Erzeugung der Schummerung
echo "----Erzeuge Schummerung----" >> /home/hoehendaten/GIS_data/DEM/error.txt
gdaldem hillshade "$f" "${OUTPUT_PATH}${i##*/}/${f##*/}_schummerung.tif" -b 1 -z 3 -s 1 -az 315 -alt 85 \
>/dev/null 2>> /home/hoehendaten/GIS_data/DEM/error.txt

#Erzeugung der Hoehenschichten
echo "----Erzeuge Hoehenschichten----" >> /home/hoehendaten/GIS_data/DEM/error.txt
gdaldem color-relief "$f" "${FARBRELIEF_DEFINITION}" "${OUTPUT_PATH}${i##*/}/${f##*/}_farbreleif.tif" -b 1 -nearest_color_entry \
>/dev/null 2>> /home/fabian/hoehendaten_schwenk/GIS_data/DEM/error.txt

#Erzeugung der Hoehenlinien
echo "----Erzeuge Hoehenlinien----" >> /home/hoehendaten/GIS_data/DEM/error.txt
gdal_contour -a HOEHE -i 0.5 "$f" "${OUTPUT_PATH}${i##*/}/${f##*/}.shp" \
>/dev/null 2>> /home/hoehendaten/GIS_data/DEM/error.txt

done
done
```



WhereGroup

```
Werde jeweils in den zu bearbeitenden Ordner
for i in ${FIND_DIRECTORIES}
do
cd $i
echo "*****Bearbeite Verzeichnis: "$i*****
echo "*****Bearbeite Verzeichnis: "$i***** >> /home/hoehendaten/GIS_data/DEM/error.txt

#Erzeuge für jede .txt-Datei in dem Ordner Schummerung, Hoehenschichten und Hoehenlinien
for f in *.txt;
do
echo "*****Bearbeite Datei: ".$f##*/*****"
echo "*****Bearbeite Datei: ".$f##*/*****" >> /home/hoehendaten/GIS_data/DEM/error.txt
#Erzeugung der Schummerung
echo "----Erzeuge Schummerung----" >> /home/hoehendaten/GIS_data/DEM/error.txt
gdaldem hillshade "$f" "${OUTPUT_PATH}${i##*/}/${f##*/}_schummerung.tif" -b 1 -z 3 -s 1 -az 315 -alt 85 \
>/dev/null 2>> /home/hoehendaten/GIS_data/DEM/error.txt

#Erzeugung der Hoehenschichten
echo "----Erzeuge Hoehenschichten----" >> /home/hoehendaten/GIS_data/DEM/error.txt
gdaldem color-relief "$f" "${FARBRELIEF_DEFINITION}" "${OUTPUT_PATH}${i##*/}/${f##*/}_farbreleif.tif" -b 1 -nearest_color_entry \
>/dev/null 2>> /home/fabian/hoehendaten_schwenk/GIS_data/DEM/error.txt

#Erzeugung der Hoehenlinien
echo "----Erzeuge Hoehenlinien----" >> /home/hoehendaten/GIS_data/DEM/error.txt
gdal_contour -a HOEHE -i 0.5 "$f" "${OUTPUT_PATH}${i##*/}/${f##*/}.shp" \
>/dev/null 2>> /home/hoehendaten/GIS_data/DEM/error.txt

done
done
```

gdal



<http://gdal.org/>

GDAL

Main Page	Related Pages	Classes	Files	Download	Issue Tracker	
-----------	---------------	---------	-------	----------	---------------	--

GDAL - Geospatial Data Abstraction Library

Select language: [English][Russian][Portuguese][French/Francais]

 GDAL is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single raster abstract data model and single vector abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing. The [NEWS](#) page describes the November 2017 GDAL/OGR 2.2.3 release.



Traditionally GDAL used to design the raster part of the library, and OGR the vector part for Simple Features. Starting with GDAL 2.0, both sides have been more tightly integrated. You can still refer to the documentation of GDAL 1.X if needed.

Master: <http://www.gdal.org>

Download: <http://download.osgeo.org>



The following utility programs are distributed with GDAL.

- [gdalinfo](#) - Report information about a file.
- [gdal_translate](#) - Copy a raster file, with control of output format.
- [gdaladdo](#) - Add overviews to a file.
- [gdalwarp](#) - Warp an image into a new coordinate system.
- [gdaltindex](#) - Build a MapServer raster tileindex.
- [gdalbuildvrt](#) - Build a VRT from a list of datasets.
- [gdal_contour](#) - Contours from DEM.
- [gdaldem](#) - Tools to analyze and visualize DEMs.
- [rgb2pct.py](#) - Convert a 24bit RGB image to 8bit palettes.
- [pct2rgb.py](#) - Convert an 8bit palettes image to 24bit RGB.
- [gdal_merge.py](#) - Build a quick mosaic from a set of images.
- [gdal2tiles.py](#) - Create a TMS tile structure, KML and simple web viewer.
- [gdal_rasterize](#) - Rasterize vectors into raster file.
- [gdaltransform](#) - Transform coordinates.
- [nearblack](#) - Convert nearly black/white borders to exact value.
- [gdal_retile.py](#) - Retiles a set of tiles and/or build tiled pyramid levels.
- [gdal_grid](#) - Create raster from the scattered data.
-



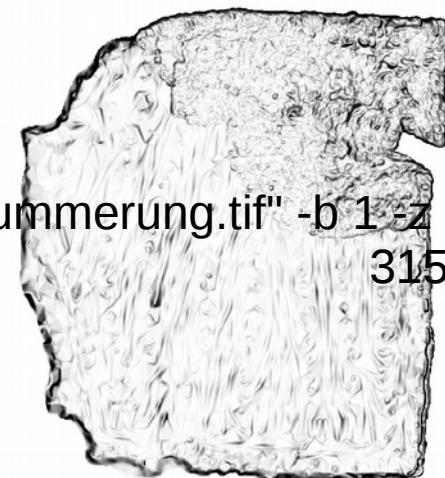
WhereGroup

To generate a shaded relief map from any GDAL-supported elevation raster :

```
gdaldem hillshade input_dem output_hillshade  
[-z ZFactor (default=1)] [-s scale* (default=1)]"  
[-az Azimuth (default=315)] [-alt Altitude (default=45)]  
[-alg ZevenbergenThorne] [-combined | -multidirectional]  
[-compute_edges] [-b Band (default=1)] [-of format] [-co "NAME=VALUE"]* [-q]
```

```
gdaldem hillshade see1.txt see1_schummerung.tif" -b 1 -z 3 -s 1 -az 315 -alt 85
```

```
gdaldem hillshade "$f" "${OUTPUT_PATH}${i##*/}/${f##*/}_schummerung.tif" -b 1 -z 3 -s 1 -az  
315 -alt 85
```



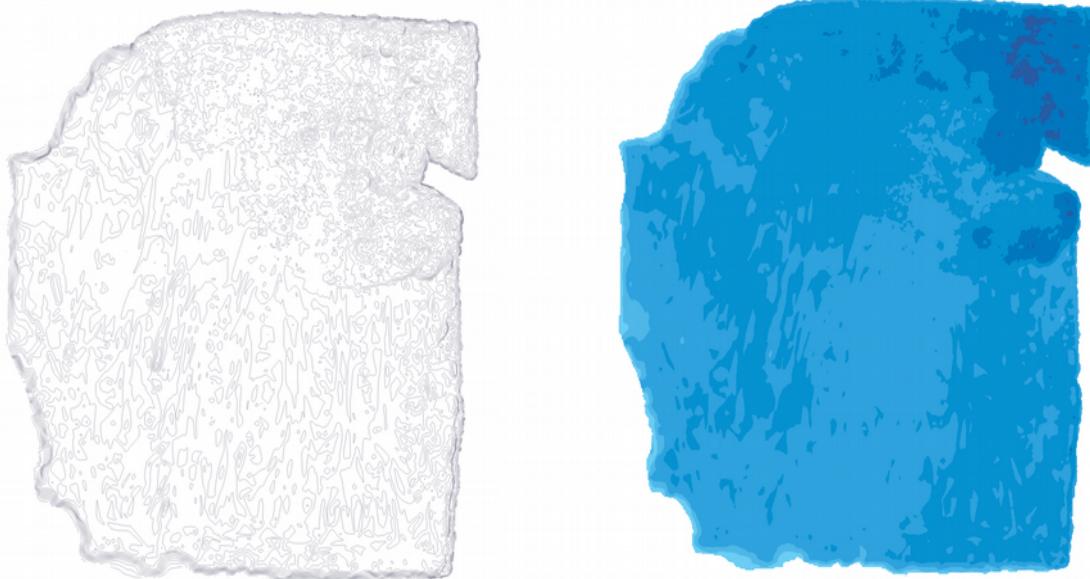


#Erzeugung der Hoehenschichten

```
gdaldem color-relief "$f" "${FARBRELIEF_DEFINITION}" "${OUTPUT_PATH}${i##*/}/${f##*/}_farbrelied.tif" -b 1 -nearest_color_entry
```

#Erzeugung der Hoehenlinien

```
gdal_contour -a HOEHE -i 0.5 "$f" "${OUTPUT_PATH}${i##*/}/${f##*}.shp" \
```





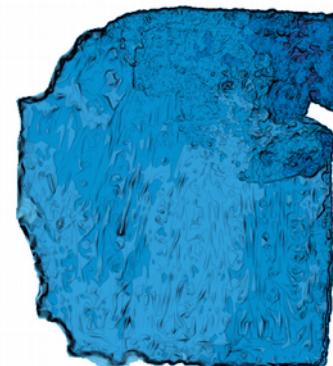
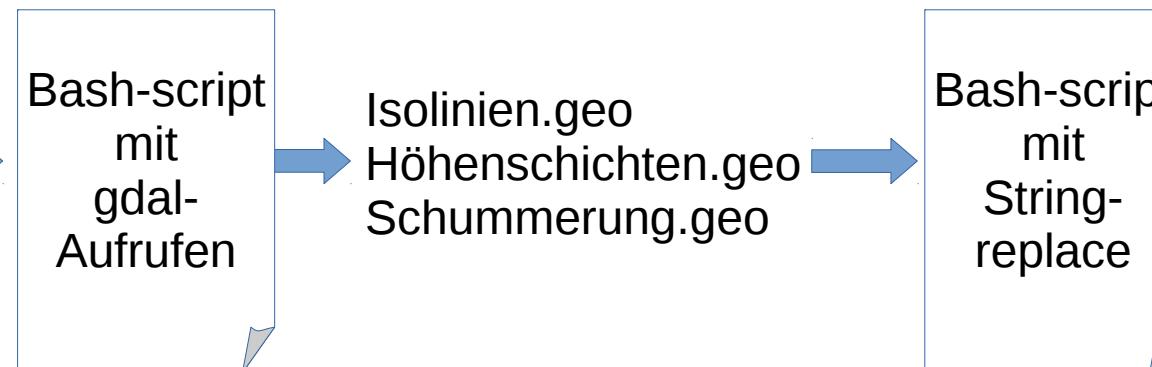
Erzeugung der WMS

- Auf Basis MapServer / mapfiles
- Templates (Grundgerüst) für Mapfiles erstellt
- Einfügen der neuen Geodaten in die Mapdatei über sh-Script



WhereGroup

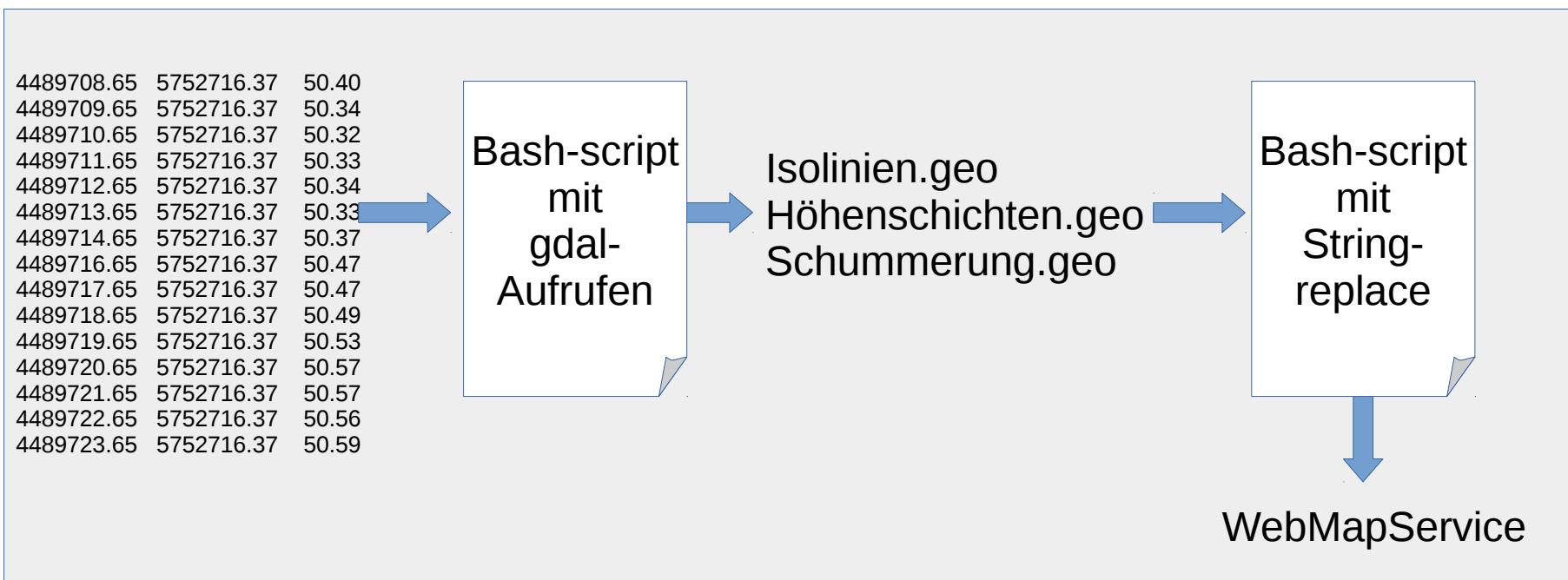
```
4489708.65 5752716.37 50.40
4489709.65 5752716.37 50.34
4489710.65 5752716.37 50.32
4489711.65 5752716.37 50.33
4489712.65 5752716.37 50.34
4489713.65 5752716.37 50.33
4489714.65 5752716.37 50.37
4489716.65 5752716.37 50.47
4489717.65 5752716.37 50.47
4489718.65 5752716.37 50.49
4489719.65 5752716.37 50.53
4489720.65 5752716.37 50.57
4489721.65 5752716.37 50.57
4489722.65 5752716.37 50.56
4489723.65 5752716.37 50.59
```





crontab

```
0 3 * * 1 /home/dredger/import_asciis.sh
```





WhereGroup

Integration insWebGIS

ja, aber



WhereGroup

Input: ascii-daten

Output: wms

**Alles nur mit ein paar Zeilen bash-script und
der mächtigen gdal-Bibliothek**



WhereGroup

Herzlichen Dank!

