

PostNAS-0.3

Dokumentation



WhereGroup

August 2008

WhereGroup GmbH & Co. KG
Siemensstraße 8
D-53121 Bonn
Fon: +49 / (0)228 / 90 90 38-0
Fax: +49 / (0)228 / 90 90 38-11
<http://www.wheregroup.com>
info@wheregroup.com



Inhaltsverzeichnis

1.1 Einführung.....	3
1.2 Installation.....	3
1.2.1 Linux.....	3
1.2.2 Windows.....	4
1.3 Programme.....	5
1.3.1 ogrinfo.....	6
1.3.2 ogr2ogr.....	8
1.3.2.1 PostGIS.....	9
1.3.2.2 Shape.....	10
1.3.2.3 Oracle.....	11

1.1 Einführung

ALKIS kommt. Damit einher kommt hochspezialisierte Software zur Migration, Datenhaltung und Bearbeitung von Katasterdaten im 3A Modell. Für viele Nachnutzer und Anwendungen ist diese Software jedoch ungeeignet, funktionsüberfrachtet und nicht finanzierbar. Es gibt jedoch einen Schlüssel zu diesen Daten, das NAS-Format. Es ermöglicht die Nutzung von ALKIS-Daten außerhalb der Spezialsoftware, sowohl in bewährten Geodateninfrastrukturen als auch herkömmlichen Desktop-PC Anwendungen.

Global betrachtet hat das deutsche Format NAS jedoch einen Nischencharakter. Um die professionelle Entwicklung gewährleisten zu können ist es erforderlich eine stabile finanzielle Grundlage für die Entwicklung zu schaffen. Die Einführung des Systems soll innerhalb eines sehr kurzen Zeitraums erfolgen, da ALK und ALKIS nicht parallel betrieben werden können, sondern eine möglichst rasche Ablösung angestrebt wird. Ein erstes konkretes Ergebnis der Datenmigration (Stufe 1) sind NAS Datensätze, mit denen bereits getestet werden kann.

Die WhereGroup hat in Kooperation mit mehreren Unternehmen, Kreis- und Landesverwaltungen das Projekt PostNAS ins Leben gerufen. Die prototypische Phase der Entwicklung wurde im Juli 2008 abgeschlossen. Mit PostNAS Version 0.3. liegt eine stabile Grundlage vor. Die Software wird von Projektpartnern bereits produktiv eingesetzt, derzeit vor allem zur Qualitätssicherung und Einführung von Nachnutzungsprozessen. PostNAS wird auf Basis der höchst erfolgreichen Bibliothek GDAL/OGR implementiert, die weltweit von Unternehmen wie ESRI und Google Earth (Professional Edition) und praktisch allen professionellen Open Source GIS Produkten unterstützt wird.

1.2 Installation

1.2.1 Linux

Die PostNAS-0.3 Distribution unter Linux enthält das Modul NAS für GDAL/OGR. Der Quellcode für die GDAL / OGR Bibliothek (Version 1.6.0dev) befindet sich, inkl. PostNAS-0.3, auf der CD ('1_PostNAS-0.3/linux/gdal_incl_PostNAS-0.3.tar.gz').

Installation:

1. Kopierung der Datei '1_PostNAS-0.3/linux/gdal_incl_PostNAS-0.3.tar.gz' von der CD auf den lokalen Rechner. Die Datei entpacken und in den Basisordner ('gdal') wechseln

```
tar -xzf gdal_incl_PostNAS-0.3.tar.gz
cd gdal
```

2. Das 'configure'-Skript im GDAL Basisverzeichnis ('gdal') mit xerces ausführen:

```
$ ./configure --with-xerces
```

Ggf. müssen Pakete noch nachinstalliert werden.

3. GDAL (im 'gdal' Basisverzeichnis) inkl. NAS kompilieren

```
$ make
```

4. Installation des GDAL-Programms

```
$ sudo make install
```

5. Testen ob OGR NAS unterstützt:

In der Eingabeaufforderung den folgenden Befehl absetzen:

```
ogrinfo --formats
```

(siehe 1.3.1)

Ausgabe:

```
...
-> "CSV" (read/write)
-> "NAS" (readonly)
-> "GML" (read/write)
...
```

=> (NAS muss mit aufgeführt sein)

1.2.2 Windows

Bei der Installation von PostNAS-0.3 handelt es sich nicht wirklich um eine Installation des Programms, sondern viel mehr um ein kopieren des PostNAS-0.3 Verzeichnisses von der CD.

Installation:

1. Das auf der CD sich befindende Verzeichnis '1_PostNAS-0.3/win/PostNAS-0.3' in ein beliebiges lokales Verzeichnis kopieren, bspw. unter 'C://Programme'.

(Die einzelnen Programme (ogrinfo.exe, ogr2ogr.exe, ...) befinden sich im Verzeichnis:
'PostNAS-0.3\bin')

2. Windows Umgebungsvariable setzen

**Die Windows Umgebungsvariable GDAL_DATA muss auf das Verzeichnis
'PostNAS-0.3\data' gesetzt sein.**

3. Testen ob OGR NAS unterstützt:

In der Eingabeaufforderung den folgenden Befehl absetzen:

```
ogrinfo.exe --formats
```

(siehe 1.3.1)

Ausgabe:

```
...
-> "CSV" (read/write)
-> "NAS" (readonly)
-> "GML" (read/write)
...
```

=> (NAS muss mit aufgeführt sein)

1.3 Programme

Allgemeines zu OGR

Die OGR Simple Features Library ist eine C++ open source - Bibliothek (sowie eine Sammlung von Kommandozeilenwerkzeugen), die lesenden (und manchmal schreibenden) Zugriff auf verschiedene Vektordatenformate einschließlich ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, Mapinfo mid/mif und NAS Formate bietet.

Mehr Informationen unter:

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

Verwendete Programme für PostNAS

Für PostNAS kommen die beiden OGR Kommandozeilenprogramme ogrinfo und ogr2ogr zum tragen, wobei ogrinfo nur Informationen zum vorliegenden Vektordatenformat liefert. Das Programm ogr2ogr ist das Hauptprogramm für PostNAS, mit diesem erfolgen die eigentlichen Konvertierungen zwischen den einzelnen Vektordatenformaten (NAS => PostGIS, NAS => Shape, ...).

1.3.1 ogrinfo

Das Programm **ogrinfo** liefert eine Vielzahl von Informationen über die in OGR unterstützten Formate (Datenquellen). Das Programm selber wird über die Kommandozeile aufgerufen.

Verwendung:

```
ogrinfo [-ro] [-q] [-where restricted_where]
        [-spat xmin ymin xmax ymax] [-fid fid]
        [-sql statement] [-al] [-so] [--formats]
        datasource_name [layer [layer ...]]
```

Mehr Informationen unter:

<http://www.gdal.org/ogr/ogrinfo.html>

Beispiele mit der NAS-Datenquelle :

1.) Testen ob das Programm OGR die Datenquelle NAS unterstützt:

linux:

```
ogrinfo --formats
```

win:

```
ogrinfo.exe --formats
```

(NAS muss mitaufgeführt sein)

2.) Übersicht von allen FeatureTypes einer NAS-XML-Datei:

linux:

```
ogrinfo <Name_der_XML-Datei>
```

win:

```
ogrinfo.exe <Name_der_XML-Datei>
```

(Auflistung der FeatureTypes)

```
$ ogrinfo 2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml
Had to open data source read-only.
INFO: Open of `2008-03-31-dhk-Bestandsdatenauszug-
GK2566.xml'
      using driver `NAS' successful.
1: AX_Buchungsstelle (None)
2: AX_Anschrift (None)
...
48: AX_Gebaeude (Polygon)
49: AX_BesondereGebaeudelinie (Line String)
...
82: AX_LagebezeichnungMitHausnummer (None)
83: AP_Darstellung (None)
```

3.) Begutachtung eines FeatureTypes in der NAS-XML-Datei :

```
$ ogrinfo 2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml
AX_Gebaeude
Had to open data source read-only.
INFO: Open of `2008-03-31-dhk-Bestandsdatenauszug-
GK2566.xml'
      using driver `NAS' successful.

Layer name: AX_Gebaeude
Geometry: Polygon
```

```

Feature Count: 826

Extent: (349956.750000, 5530126.641000) - (354981.434000,
5532019.741000)

Layer SRS WKT:

(unknown)

gml_id: String (16.0)
beginnt: String (20.0)
advStandardModell: String (4.0)
anlass: Integer (0.0)
gebaeudedefunktion: Integer (0.0)
description: Integer (0.0)
name: String (25.0)
zeigtAuf: String (0.0)
lageZurErdoberflaeche: Integer (0.0)
bauweise: Integer (0.0)
zustand: Integer (0.0)
...
<Auflistung der Features>

```

1.3.2 ogr2ogr

Das Programm **ogr2ogr** ist ein Programm zur Konvertierung zwischen Vektorformaten. Beim Programmaufruf kann eine Vielzahl von Optionen mit übergeben werden, so ist es möglich dass zu exportierende Format zu beeinflussen bzw. zu modifizieren. Das Programm wird ebenfalls über die Kommandozeile aufgerufen.

Verwendung:

```

ogr2ogr [-skipfailures] [-append] [-update] [-f
format_name] [-gtn] [-select field_list] [-
where restricted_where] [-sql <sql_statement>]
[--help-general] [-spat xmin ymin xmax ymax] [-
preserve_fid] [-fid FID] [-a_srs srs_def] [-
t_srs srs_def] [-s_srs srs_def] [[-dsco
NAME=VALUE] ...] dst_datasource_name
src_datasource_name [-lco NAME=VALUE] [-nln
name] [-nlt type] [layer [layer ...]]

```


Mehr Informationen unter:

<http://www.gdal.org/ogr/ogr2ogr.html>

Anmerkung :

Bei den nachfolgenden Datenkonvertierungs-Befehlen und Beispielen handelt es sich um Kommandozeilenbefehlen auf einem Linux Rechner.

Bei der Verwendung der Befehle unter Windows ist der Gebrauch der Programme mit der Endung **.exe** notwendig. Ebenfalls ist es ggf. nötig absolute Pfade zu benutzen.

1.3.2.1 *PostGIS*

Die Benutzung von Oracle in ogr2ogr (inkl. PostNAS) setzt die zuvor installierte Datenbank PostgreSQL + PostGIS voraus. Ebenfalls muss zuvor eine Datenbank (-Instanz) mit PostGIS-Funktionalität angelegt worden sein.

Befehl allgemein:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=<DBNAME> user=<USER>  
host=<HOST> port=<PORT> password=<PASSWORT>" -a_srs  
<EPSG_NR> <NAME_DER_NAS_DATEI>
```

Beispiel 1:

Import aller FeatureTypes einer NAS-Datei in die PostgreSQL + PostGIS DB

```
ogr2ogr -f "PostgreSQL" PG:"dbname=alkis_lverm_geo_rlp  
user=postgres host=localhost port=5432 password=postgres"  
-a_srs EPSG:25832 2008-03-31-dhk-Bestandsdatenauszug-  
GK2566.xml
```

Beispiel 2:

Import eines einzelnen FeatureTypes (AX_Gebaeude) einer NAS-Datei in die PostgreSQL + PostGIS DB

```
ogr2ogr -f "PostgreSQL" PG:"dbname=alkis_lverm_geo_rlp  
user=postgres host=localhost port=5432 password=postgres"  
-a_srs EPSG:25832 2008-03-31-dhk-Bestandsdatenauszug-  
GK2566.xml AX_Gebaeude
```

Die Daten werden direkt in die Datenbank geschrieben, es erfolgt kein Umweg über sql-Dateien oder sonstiges. Ebenfalls erfolgt eine Indizierung der Geodaten und das hinterlegen von Metadaten für die einzelnen FeatureTypes bzw. Tabellen.

1.3.2.2 Shape

Aufgrund des eingeschränkten Datenmodells von ESRI Shapefiles, das keine unterschiedlichen Geometrien für ein FeatureType zulässt, ist die Konvertierung etwas beschwerlich:

1. Lösungsweg:

Die NAS-Datei muss von doppelten Geometrien für ein FeatureType gesäubert werden. Dies ist schon mit den Demodaten geschehen, das Ergebnis ist die Datei: '2_Testdaten/nas/mod_2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml' (Siehe [Beispiel 1](#))

2. Lösungsweg:

Die unterschiedlichen Features werden einzeln in die Shapefiles überführt siehe [Beispiel 2](#) (PostGIS). Ggf. kann hier ein Skript für die Stapelverarbeitung geschrieben werden.

In kommenden Versionen von PostNAS soll die Konvertierung von NAS nach Shape etwas anwendungsfreundlicher werden.

Befehl allgemein:

```
ogr2ogr -f "ESRI Shapefile" -a_srs <EPSG_NR>  
<EXPORT_VERZEICHNIS> <NAME_DER_NAS_DATEI>
```

Beispiele 1:

Konvertierung aller FeatureTypes einer NAS-Datei in Shapes

(Zuvor muss das Ziel-Verzeichnis 'shp' erstellt worden sein)

```
ogr2ogr -f "ESRI Shapefile" -a_srs "EPSG:25832" ./shp/  
mod_2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml
```

(Anmerkung in Windows : '... \shp ...')

Beispiele 2:

Konvertierung eines einzelnen FeatureTypes (AX_Gebaeude) einer NAS-Datei in ein Shape

(Zuvor muss das Ziel-Verzeichnis 'shp' erstellt worden sein)

```
ogr2ogr -f "ESRI Shapefile" -a_srs "EPSG:25832" ./shp/
mod_2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml
AX_Gebaeude
```

1.3.2.3 Oracle

Die Benutzung von Oracle in ogr2ogr (inkl. PostNAS) setzt die installierte Oracle Database Instant Client (OCI) voraus. Der Client ist von der Seite (<http://www.oracle.com/technology/tech/oci/instantclient/index.html>) herunterzuladen und zu installieren.

Vor der eigentlichen Konvertierung / Import der NAS-Daten in eine Oracle DB muss ein Datenbankbenutzer angelegt worden sein.

Befehl allgemein:

```
ogr2ogr -f OCI OCI:<DB_USER>/<PASSWORT>@<SERVICE_NAME_
oder_SID> -lco <DIMENSION_DER_DATEN> -lco <EPSG_NR>
<NAME_DER_NAS_DATEI>
```

Beispiel 2:

Konvertierung aller FeatureTypes einer NAS-Datei nach Oracle

```
ogr2ogr -f OCI OCI:lverm_geo_rlp/root@xe -lco DIM=2 -lco
SRID=25832 2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml
```

Beispiel 1:

Konvertierung eines einzelnen FeatureTypes (AX_Gebaeude) einer NAS-Datei nach Oracle

```
ogr2ogr -f OCI OCI:lverm_geo_rlp/root@xe -lco DIM=2 -lco
SRID=25832 2008-03-31-dhk-Bestandsdatenauszug-GK2566.xml
AX_Gebaeude
```

Die Daten werden direkt in die Datenbank geschrieben, es erfolgt kein Umweg über sql-

Dateien oder sonstiges. Ebenfalls erfolgt eine Indizierung der Geodaten und das hinterlegen von Metadaten für die einzelnen FeatureTypes bzw. Tabellen.

Mehr Informationen unter:

http://www.gdal.org/ogr/drv_oci.html
