

# OSM Backgroundprovider

---

## OpenStreetMap und NASA SRTM Höhendaten

**Torsten Armbrust**

**21.06.2009**

Diese Dokumentation beschreibt die Aufbereitung von Kartendaten aus dem OpenStreetMap Projekt und Höhenlinien auf Basis der NASA SRTM Daten in einer lokalen PostgreSQL Datenbank sowie die Integration der Geodaten zur Darstellung als Kartenhintergrund in GNavigia, einem grafisch-interaktiven Editor für Tracks, Track- und Wegpunkte. Die Dokumentation befindet sich unter <http://www.skiraware.de/GPS/GNavigia>.

<b>GNAVIGIA BACKGROUNDPROVIDER</b>	<b>4</b>
<b>MAPNIK RENDERER</b>	<b>5</b>
GEODATENQUELLE OPENSTREETMAP DATEN	6
GEODATENQUELLE HÖHENLINIEN	7
<b>INSTALLATION POSTGRESQL DATENBANK</b>	<b>8</b>
INSTALLATION BASISSYSTEM	8
POST INSTALLATION	8
<b>VERWALTUNG UND IMPORT DER OPENSTREETMAP DATEN</b>	<b>9</b>
ERSTELLEN DER LOKALEN OPENSTREETMAP-DATENBANK	9
INSTALLATION <i>INTARRAY CONTRIB MODULE</i>	10
IMPORT OPENSTREETMAP DATEN	11
OPENSTREETMAP DATEN LIZENZ	11
OSM XML DATENBANKAUSZUG	11
IMPORT DER OSM DATEN MITTELS OSM2PGSQL	12
PRIMARY KEYS AUF DEN DATENBANKTABELLEN ERSTELLEN	14
OPTIMIERUNG DER OSM DATENBANK	15
<b>VERWALTUNG UND IMPORT - HÖHENLINIENDATEN</b>	<b>17</b>
ERSTELLEN DER LOKALEN HÖHENLINIEN-DATENBANK	17
NASA SRTM DATEN LIZENZ	17
DOWNLOAD NASA SRTM DATEN	18
IMPORT DER HÖHENLINIEN	18
GDAL - GEOSPATIAL DATA ABSTRACTION LIBRARY	18
SKRIPTE	18
IMPORT	19
OPTIMIERUNG DER SRTM DATENBANK	19
<b>INSTALLATION MAPNIK</b>	<b>20</b>
INSTALLATION PYTHON INTERPRETER	20
INSTALLATION MAPNIK	20
MAPNIK FUNKTIONSTEST	21
<b>INSTALLATION UND KONFIGURATION GNAVIGIA OSM-BACKGROUNDPROVIDER</b>	<b>22</b>
FUNKTIONSBESCHREIBUNG	22
INSTALLATION	23

VERZEICHNISSTRUKTUR	23
PYTHON SKRIPT GENERATE_GNAVIGIA_IMAGE	23
OSM XML DATEI	24
<b>KONFIGURATION</b>	<b>27</b>
UMGEBUNGSVARIABLEN	27
ZUGRIFF AUF OSM BACKGROUNDPROVIDER IN GNAVIGIA	28
LOGGING	30
<b><u>ABBILDUNGSVERZEICHNIS</u></b>	<b>31</b>

---

## GNAVIGIA BACKGROUNDPROVIDER

GNavigia bietet eine COM Schnittstelle<sup>1</sup> an, über die ein Backgroundprovider angesprochen werden kann. Ein Backgroundprovider ist ein COM Server, der auf Anfrage von GNavigia für den aktuellen Kartenausschnitt ein Hintergrundbild berechnet und zurückliefert.

Dieser Backgroundprovider wird nach dem Ausführen einer Kartenoperation (z.B. Zoom oder Verschieben) automatisch durch GNavigia angesprochen. Das berechnete Hintergrundbild wird als unterste Ebene der Darstellung genutzt. GNavigia berechnet zusätzlich weitere Ebenen, die die aktuellen GPS Daten (Inhalt der GTD Datei<sup>2</sup>) und weitere Kartenattribute wie z.B. den aktuellen Darstellungsmaßstab oder den Copyright-Vermerk enthalten.

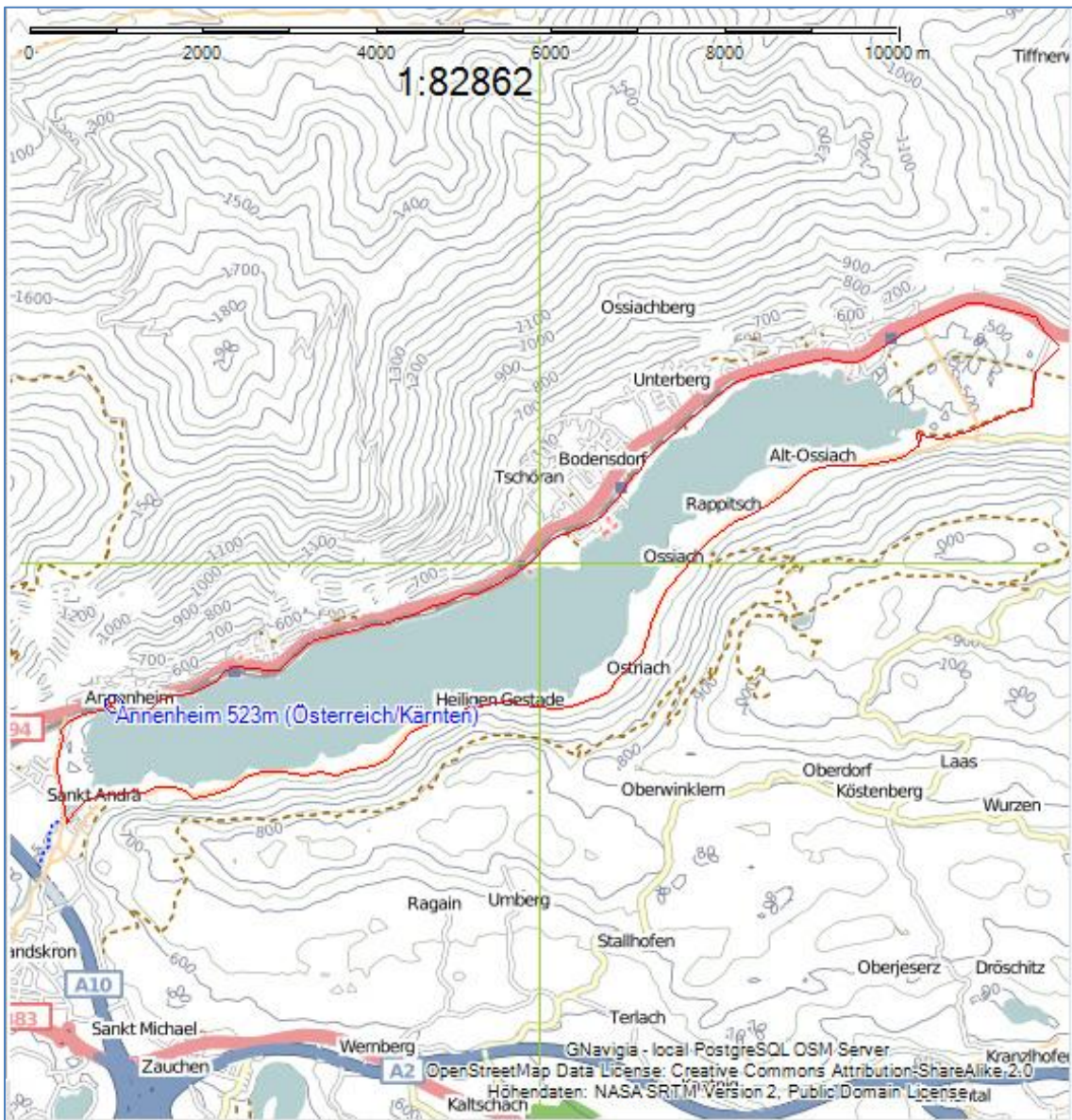


Abbildung 1: Beispieldarstellung einer Radtour mit OSM/SRTM Hintergrunddaten

<sup>1</sup> <http://www.skiraware.de/GPS/GNavigia/D/DocGNavigiaBackgroundProviderExtension.html>

<sup>2</sup> <http://www.skiraware.de/GPS/GNavigia/D/DocGNavigiaFuerFortgeschrittene.html#Datenstruktur>

## MAPNIK RENDERER

Zum Erstellen des Hintergrundbildes wird Mapnik<sup>3</sup> als Renderer genutzt. Mapnik wird über ein Python<sup>4</sup> API angesprochen und berechnet aus verschiedenen Datenquellen und den Regelsätzen zur Ausgestaltung dieser Geodaten eine Bilddatei (z.B. im PNG Format). Über die COM Schnittstelle des Backgroundproviders wird diese Bilddatei als Hintergrundkarte an GNavigia zurückgeliefert. Die folgende Abbildung zeigt ein Hintergrundbild ohne die zusätzlichen Ebenen, die GNavigia bei der Darstellung hinzufügt.



Abbildung 2: Mapnik Hintergrundkarte mit OSM Daten und Höhenlinien

Zur Erstellung der Hintergrundkarten in dieser Dokumentation werden Mapnik zwei PostgreSQL Datenbanken als Geodatenquellen zur Verfügung gestellt.

<sup>3</sup> <http://www.mapnik.org/>

<sup>4</sup> <http://www.python.org/>

## GEODATENQUELLE OPENSTREETMAP DATEN

Zur Darstellung der Kartendaten aus dem OpenStreetMap Projekt wird eine lokale PostgreSQL Datenbank erstellt und ein räumlich begrenzter Datensatz aus der OSM Primärdatenbank importiert.

Das OpenStreetMap Projekt ist ähnlich wie die Wikipedia aufgebaut und hat sich zum Ziel gesetzt, weltweit freie Kartendaten anzubieten. Jeder kann an der Vervollständigung der Kartendaten mitarbeiten. Die OSM Primärdatenbank enthält den aktuellen Geodatenbestand des Projektes. Zurzeit gibt es keinen direkten Zugriff für „Jedermann“ auf die Geodaten in der Primärdatenbank. Wöchentlich (meistens am Mittwoch) wird ein Auszug (Dump) aus der Primärdatenbank erstellt und zu freien Verfügung auf den FTP Server des Projektes abgelegt. Dieser Dump ist eine XML Datei, deren Format durch das OSM Protokoll<sup>5</sup> (aktuell Version 0.5) spezifiziert wird, das sogenannte Planet File.

Zusammen mit dem Regelsatz zur Ausgestaltung der OSM Daten kann Mapnik ein Kartenbild des aktuellen Ausschnitts wie in der folgenden Abbildung berechnen.



Abbildung 3: Teilkartenbild OpenStreetMap Datensatz (Stand 12/2008)

<sup>5</sup> [http://wiki.openstreetmap.org/wiki/OSM\\_Protocol\\_Version\\_0.5](http://wiki.openstreetmap.org/wiki/OSM_Protocol_Version_0.5)

## GEODATENQUELLE HÖHENLINIEN

Zur Darstellung der Höhenlinien wird eine lokale PostgreSQL Datenbank erstellt. Die Höhenlinien werden mit Hilfe von GDAL<sup>6</sup> (Geospatial Data Abstraction Library) aus dem NASA SRTM Datensatz errechnet und in die PostgreSQL Datenbank importiert. Zusammen mit dem Regelsatz zur Ausgestaltung der Höhenlinien erstellt Mapnik ein Kartenbild wie in der folgenden Abbildung.

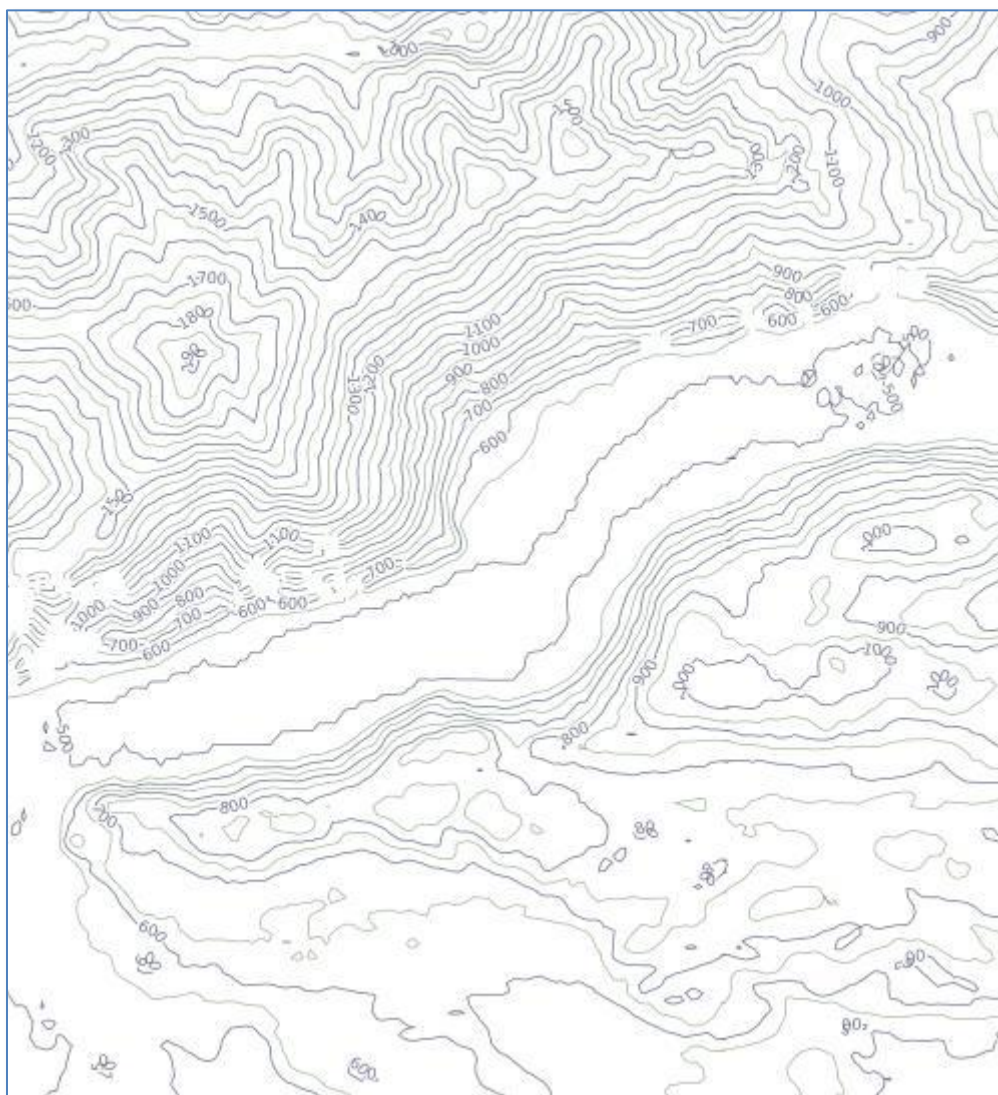


Abbildung 4: Teilkartenbild Höhenlinien aus NASA SRTM Datensatz

<sup>6</sup> <http://www.gdal.org/>

## INSTALLATION POSTGRESQL DATENBANK

In diesem Kapitel wird die Installation und Konfiguration der PostgreSQL<sup>7</sup> Datenbank beschrieben. Zusätzlich zur Standardinstallation wird zur Verwaltung der Geometriedaten die Erweiterung PostGIS<sup>8</sup> benötigt.

Basis der Installationsanleitung ist das aktuelle Dokument zur PostgreSQL Datenbank Version 8.3.\*.

### INSTALLATION BASISSYSTEM

- Installationspakete unter : <http://www.postgresql.org/download/windows>.
- Installationspaket mit Administratorenrechten starten und den Anweisungen folgen.
- Standardpakete installieren, zusätzlich benötigte Pakete werden im folgenden Kapitel (Post Installation) nachinstalliert.

### POST INSTALLATION

- Nach erfolgreicher Installation die Anwendung *EnterpriseDB Stack Builder* mit Administratoren Rechten aufrufen (*START/Alle Programme/PostgreSQL 8.3/Application Stack Builder*)
- Anmeldung am installierten PostgreSQL Datenbankserver ausführen
- Nach erfolgreicher Anmeldung muss das folgende Zusatzpaket installiert werden:
  - *PostGIS 1.3.3 for PostgreSQL 8.3 v 1.3.3*: PostGIS Erweiterung

Hinweis: Es ist empfehlenswert, nach erfolgreichem Abschluss der Installation das Unterverzeichnis *bin* im Installationsverzeichnis (z.B. *C:\Program Files\PostgreSQL\8.3\bin*) mit in die PATH Variable aufzunehmen.

---

<sup>7</sup> <http://www.postgresql.org/>

<sup>8</sup> <http://www.postgis.org/>



## VERWALTUNG UND IMPORT DER OPENSTREETMAP DATEN

Die Kartendaten des OpenStreetMap Projektes sind im Internet in Form von Dumps der OSM Primärdatenbank erhältlich. Das folgende Kapitel erläutert die Erstellung der lokalen Datenbank für OSM Daten und den Import der Dumps in die lokale OSM Datenbank.

### ERSTELLEN DER LOKALEN OPENSTREETMAP-DATENBANK

Zum Erstellen der lokalen OpenStreetMap-Datenbank wird das Administrationstool *pgAdmin III* genutzt. Nach dem Öffnen des Programms muss man sich mit dem installierten Datenbankserver verbinden.

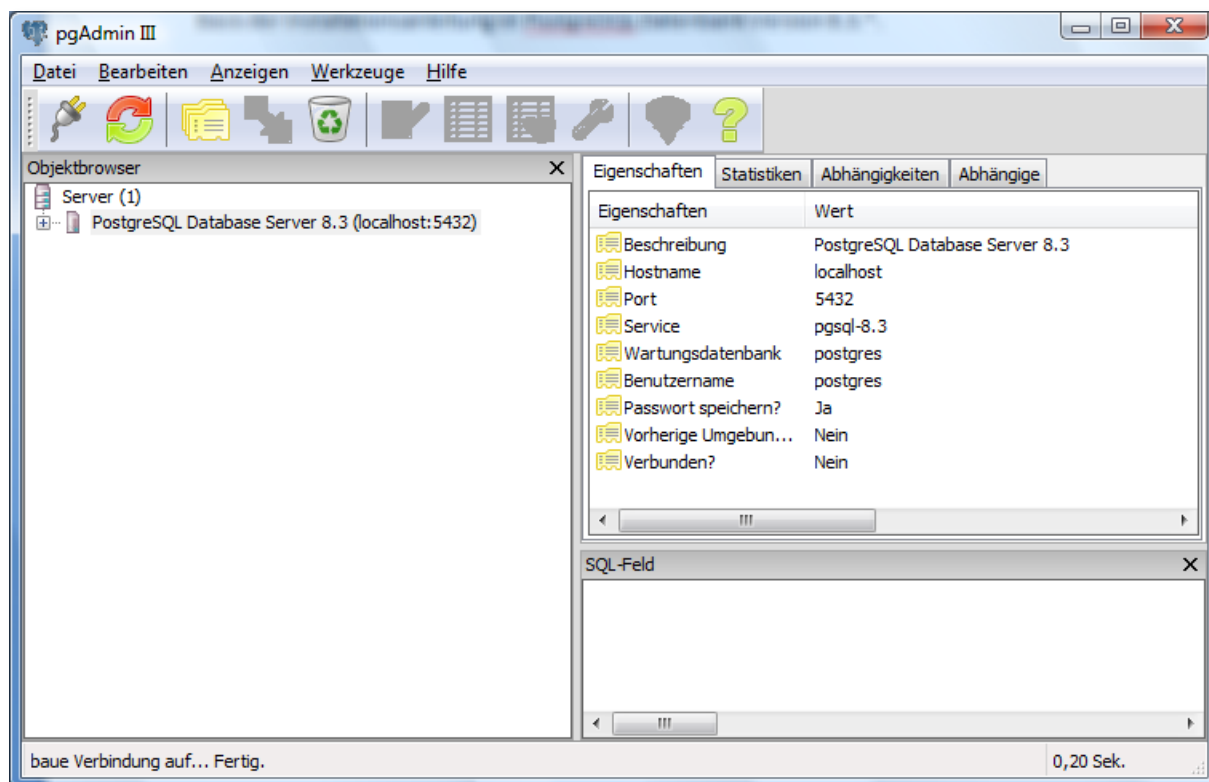


Abbildung 5: Datenbank Administrationstool

Zur Anmeldung faltet man den Knoten *PostgreSQL Database Server 8.3 ...* auf. Darunter befindet sich der Knoten *Datenbanken*. Diesen Knoten ebenfalls auffalten. Nach erfolgreicher Installation müssen sich hier die folgenden drei Datenbanken befinden:

- *postgis*
- *postgres*
- *template\_postgis*

Um eine neue Datenbank zu erstellen, den Knoten *Datenbanken* selektieren und die rechte Maustaste drücken. Im Kontextmenü den Eintrag *Neue Datenbank...* auswählen. Es öffnet sich der Dialog zum Erstellen von Datenbanken.

Der **Name** der Datenbank kann frei vergeben werden, in den folgenden Schritten der Dokumentation wird von *osm* als Datenbanknamen ausgegangen.

Als **Eigentümer** wird *postgres* ausgewählt.

Die **Kodierung** der Datenbank muss für OpenStreetMap Daten zwingend auf *UTF8* gesetzt werden.

Die **Vorlage** *template\_postgis* importiert die notwendige PostGIS Geometrie Erweiterung und ist daher ebenfalls zwingend anzugeben.

Mit dem Betätigen der Schaltfläche *OK* wird die Datenbank angelegt und unterhalb des Knotens *Datenbanken* angezeigt.

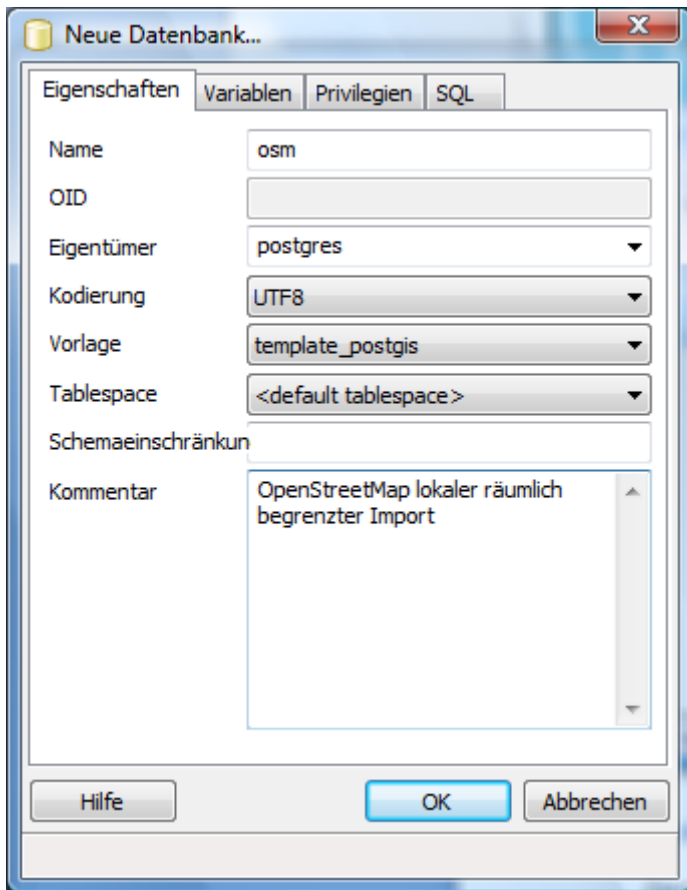


Abbildung 6: Erstellen einer neuen OSM Datenbank

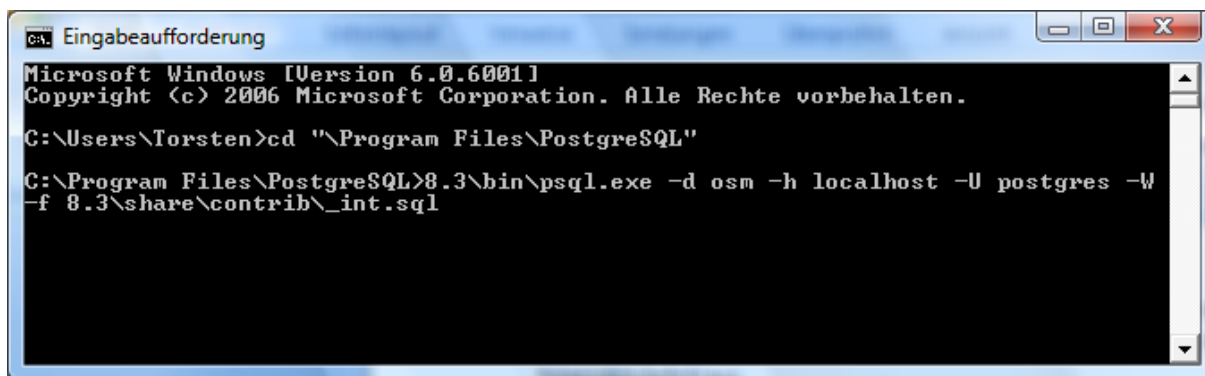
## INSTALLATION INTARRAY CONTRIB MODULE

Um später Differenzupdates von der OSM Datenbank erstellen und einspielen zu können muss zusätzlich das *Intarray Contrib Module*<sup>9</sup> installiert werden. Soll kein Differenzupdate genutzt werden, kann dieses Kapitel übersprungen werden.

Zur Installation des Modules öffnen Sie eine Eingabeaufforderung und wechseln Sie in das Installationsverzeichnis der PostgreSQL Installation (z.B. *C:\Program Files\PostgreSQL*). Führen Sie den folgenden Befehl aus.

```
8.3\bin\psql.exe -d osm -h localhost -U postgres -W -f 8.3\share\contrib\_int.sql
```

<sup>9</sup> <http://www.postgresql.org/docs/current/static/intarray.html>



```
ca. Eingabeaufforderung
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Torsten>cd "\Program Files\PostgreSQL"
C:\Program Files\PostgreSQL>8.3\bin\psql.exe -d osm -h localhost -U postgres -W
-f 8.3\share\contrib\_int.sql
```

Abbildung 7: Konsolen Screenshot Installation Intarray Contrib Module

Hinweis: Das Thema „Differenzupdate“ durchleuchte ich gerade. Sobald hier ein sinnvoller Weg gefunden ist, wird an dieser Stelle die Dokumentation erweitert.

## IMPORT OPENSTREETMAP DATEN

Zum Import der OpenStreetMap Daten wird ein Dump aus der OSM Primärdatenbank und das Programm *osm2pgsql* benötigt.

### OPENSTREETMAP DATEN LIZENZ

Die Daten aus dem OpenStreetMap Projekt unterliegen der *Creative Commons Attribution-ShareAlike 2.0 license*

Zitat:

„...Über OpenStreetMap Daten kann frei unter den Bedingungen der Creative Commons Attribution-ShareAlike 2.0 Lizenz verfügt werden.

*OpenStreetMap Software ist unter der GNU GPL lizenziert.*

*Das heißt für Dich (dieses ist kein anwaltlicher Rat), dass Du mit den Daten eigentlich machen kannst was Du willst, so lange Du den Urheber und die Lizenz angibst und jeder das selbe mit Deinen daraus resultierenden von Dir produzierten Werken tun kann....“*

(Quelle: [http://wiki.openstreetmap.org/wiki/DE:OpenStreetMap\\_License](http://wiki.openstreetmap.org/wiki/DE:OpenStreetMap_License))

### OSM XML DATENBANKAUSZUG

Das Planet File<sup>10</sup> enthält den aktuellen Stand der OSM Primärdatenbank. Diese Datei wird wöchentlich aktualisiert und als Download zur Verfügung gestellt. Allerdings nähert sich die Größe der Datei im entpackten Zustand der 100 GB Grenze (Stand 12/2008) und ist somit für einen Import von kleineren Gebieten nicht handlich. Alternativ können über den Downloadserver der GeoFabrik<sup>11</sup> kleinere Gebiete als OSM Datei heruntergeladen werden. Diese Dateien enthalten definierte Gebiete/Ausschnitte (z.B. Länder, Bundesländer,...) aus dem aktuellen Planet File.

<sup>10</sup> <http://wiki.openstreetmap.org/index.php/Planet.osm>

<sup>11</sup> <http://download.geofabrik.de/osm/>

**Achtung:** Der Import von OSM Daten von mehreren aneinander grenzenden Gebieten (z.B. Deutschland, Österreich) ist mit *osm2pgsql* nicht fehlerfrei möglich. Der Grund liegt in der Art der Aufspaltung von z.B. Linien an den Exportgrenzen. Hierbei entstehen in Dateien benachbarter Gebiete, Segmente mit identischen Schlüsseln. Dies führt beim Import des zweiten Gebietes zum Abbruch wegen der Verletzung von Unique Constraints auf dem Schlüssel. Es empfiehlt sich daher, das nächst größere Gebiet als OSM Datei herunterzuladen und beim Import mit *osm2pgsql* eine räumliche Einschränkung in Form einer Bounding Box zu definieren.

Für diese Dokumentation wird ein Gebiet von der dänischen Grenze im Norden bis Italien im Süden und vom Atlantik im Westen bis zur polnisch-deutschen Grenze im Osten (BBOX -5.0, 40.0, 18.0, 55.0 EPSG:4326) importiert. Für den Import wird der Europa Datensatz<sup>12</sup> vom GeoFabrik Download Server genutzt.

## IMPORT DER OSM DATEN MITTELS OSM2PGSQL

*Osm2pgsql*<sup>13</sup> wird primär als Linux Programm weiterentwickelt. Die aktuelle Version aus dem OpenStreetMap Entwicklungsumgebung (SVN<sup>14</sup>) bietet hierfür einen sogenannten SLIM Modus an. Mit diesem Modus, in Verbindung mit der Angabe des maximal zu nutzenden Arbeitsspeichers, kann der Bedarf an Arbeitsspeicher zu Lasten der Performance begrenzt werden.

Für Windows existiert (Stand 03/2009) eine brauchbare Portierung<sup>15</sup>, die ebenfalls ihren Bedarf an Arbeitsspeicher begrenzen kann. Die Installation beschränkt sich auf das Entpacken des ZIP Archives z.B. nach *C:\Programme\osm2pgsql*.

**Hinweis:** Es gibt für Windows noch eine ältere Portierung, die allerdings die Begrenzung des Arbeitsspeichers nicht beinhaltet. Es ist empfehlenswert, für den Import größerer OSM Dateien, die hier verlinkte Version zu verwenden.

Öffnen Sie eine Eingabeaufforderung und wechseln Sie in das *Osm2pgsql* Installationsverzeichnis. Dort rufen Sie *Osm2pgsql* mit der folgenden Parameterleiste auf.

```
osm2pgsql --bbox -5.0,40.0,18.0,55.0 -c -d osm -l -s -C 1024 -U postgres -W -H %rechnername% %OSM_DATA_DIR%/europe.osm.bz2
```

Beschreibung der Aufrufparameter:

<i>--bbox -5.0,40.0,18.0,55.0</i>	Schränkt den Import auf den angegebenen räumlichen Bereich (Bounding Box) ein.
<i>-c</i>	Create Mode, erstellt die Tabellen in der Datenbank und importiert die Daten. Bestehende Daten werden gelöscht. Sollen bestehende Daten beibehalten werden, muss hier der Parameter <i>-a</i> verwendet werden.

<sup>12</sup> <http://download.geofabrik.de/osm/europe.osm.bz2>

<sup>13</sup> <http://wiki.openstreetmap.org/index.php/Osm2pgsql>

<sup>14</sup> <http://svn.openstreetmap.org/applications/utils/export/osm2pgsql/>

<sup>15</sup> <http://wiki.openstreetmap.org/index.php/Osm2pgsql#Windows>

<i>-d osm</i>	Name der PostgreSQL Datenbank, in die der Import durchgeführt werden soll.
<i>-l</i>	OSM Daten werden beim Import in das Spatial Reference System EPSG 4326 (lat/lon) transformiert.
<i>-s</i>	SLIM Mode: Bewirkt das Erstellen von temporären Tabellen im Arbeitsspeicher.
<i>-C 1024</i>	Begrenzt den genutzten Arbeitsspeicher auf 1024 MB. (Abhängig von der Recherausstattung)
<i>-U postgres</i>	Datenbankbenutzer für den Import.
<i>-W</i>	Erzwingt die Abfrage des Passworts zum Datenbankbenutzer. Ist keine Passwortangabe für den Datenbankbenutzer notwendig, kann dieser Parameter entfallen.
<i>-H %rechnername%</i>	Notwendig, falls sich die PostgreSQL Datenbank auf einem anderen Rechner (also nicht <i>localhost</i> ) befindet.

Als Abschluss wird die zu importierende OSM Datei (*europa.osm.bz2*) angegeben.

```

C:\Program Files\osm2pgsql>osm2pgsql.exe --bbox -5.0,40.0,18.0,55.0 -c -d osm -l
-s -C 1024 -U postgres -W -H localhost \Users\Torsten\Downloads\europa.osm.bz2
osm2pgsql SUN version 0.55-20081113 $Rev: 10464 $

Password:
Using projection SRS 4326 (Latlong)
Applying Bounding box: -5.000000,40.000000 to 18.000000,55.000000
Setting up table: planet_osm_point
Setting up table: planet_osm_line
Setting up table: planet_osm_polygon
Setting up table: planet_osm_roads
Mid: postgresql, scale=100000000, cache=1024MB, maxblocks=131073*zd
Setting up table: planet_osm_nodes
*** WARNING: intarray contrib module not installed
*** The resulting database will not be usable for applying diffs.
HINWEIS: CREATE TABLE / PRIMARY KEY erstellt implizit einen Index  TTTplanet_osm
nodes_pkeyT% f|T|r Tabelle  TTTplanet_osm_nodesT%
Setting up table: planet_osm_ways
HINWEIS: CREATE TABLE / PRIMARY KEY erstellt implizit einen Index  TTTplanet_osm
ways_pkeyT% f|T|r Tabelle  TTTplanet_osm_waysT%
Setting up table: planet_osm_rels
HINWEIS: CREATE TABLE / PRIMARY KEY erstellt implizit einen Index  TTTplanet_osm
rels_pkeyT% f|T|r Tabelle  TTTplanet_osm_relsT%

Reading in file: \Users\Torsten\Downloads\europa.osm.bz2
Processing: Node(86701k) Way(8860k) Relation(63k)
Node stats: total(86701897), max(364175829)
Way stats: total(8860824), max(32353951)
Relation stats: total(63393), max(102317)

Going over pending ways
processing way (1702k)

Going over pending relations

node cache: stored: 55191738(94.22%), storage efficiency: 41.12%, hit rate: 69.8
3%

C:\Program Files\osm2pgsql>_

```

Abbildung 8: Konsolen Screenshot eines erfolgreichen Imports

**Hinweis:** Der Import kann je nach Rechnergeschwindigkeit einige Stunden dauern. Im aktuellen Fall lief der Import auf einem Rechner (Intel® Core™2 6400/2.13GHz, 2GB RAM, Windows VISTA Home Premium) ca. 12,5h.

**Hinweis:** Die fehlerhafte Ausgabe von Umlauten ist auf den Zeichensatz der Konsole zurück zu führen.

**Hinweis:** Wurde das *intarray contrib module* nicht installiert (siehe Kapitel *Installation Intarray Contrib Module*), werden beim Import der OSM Daten Warnungen angezeigt. Diese können ignoriert werden.

## PRIMARY KEYS AUF DEN DATENBANKTABELLEN ERSTELLEN

Während des Imports der OSM Daten werden in der OSM Datenbank die für die Verwaltung der OSM Daten notwendigen Datenbanktabellen angelegt (Parameter `-c`). Diese Datenbanktabellen enthalten keine Primary Keys. Viele, auf die OSM Daten aufbauenden, Anwendungen (z.B. Quantum GIS) benötigen zur Funktion Primary Keys auf den Datenbanktabellen. Aus diesem Grund kann es sinnvoll sein, nach Abschluss des Imports diese anzulegen.

Zum erstellen der Primary Keys legen Sie eine Textdatei mit der Endung *sql* (z.B. *osmpk.sql*) an. Kopieren Sie die folgenden Zeilen in die Textdatei.

```
ALTER TABLE planet_osm_point ADD COLUMN pid serial;
```

```
ALTER TABLE planet_osm_point ADD CONSTRAINT pid_pkey_point PRIMARY KEY (pid);
```

```
ALTER TABLE planet_osm_line ADD COLUMN pid serial;
```

```
ALTER TABLE planet_osm_line ADD CONSTRAINT pid_pkey_line PRIMARY KEY (pid);
```

```
ALTER TABLE planet_osm_polygon ADD COLUMN pid serial;
```

```
ALTER TABLE planet_osm_polygon ADD CONSTRAINT pid_pkey_polygon PRIMARY KEY (pid);
```

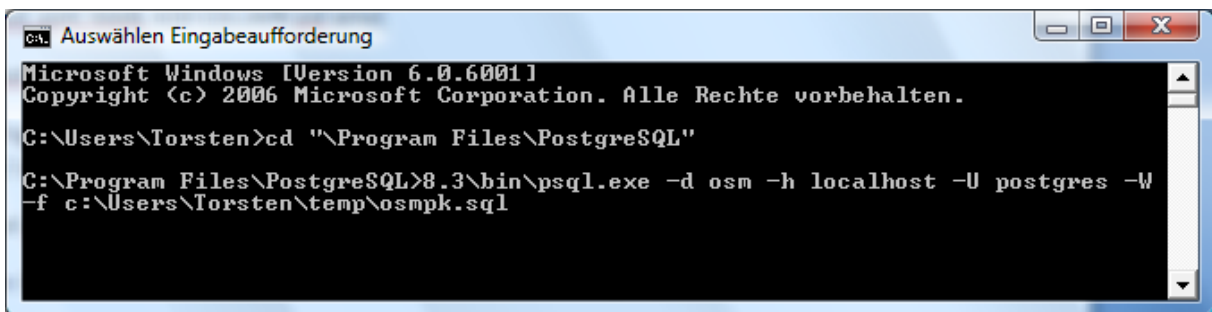
```
ALTER TABLE planet_osm_roads ADD COLUMN pid serial;
```

```
ALTER TABLE planet_osm_roads ADD CONSTRAINT pid_pkey_roads PRIMARY KEY (pid);
```

Öffnen Sie eine Eingabeaufforderung und wechseln Sie in das Installationsverzeichnis der PostgreSQL Installation (z.B. `C:\Program Files\PostgreSQL`). Führen Sie den folgenden Befehl aus.

```
8.3\bin\psql.exe -d osm -h localhost -U postgres -W -f %PFAD_ZUR_DATEI%\osmpk.sql
```

Hinweis: Tauschen Sie den Marker `%PFAD_ZUR_DATEI%` durch den Pfad zur `osmpk.sql` Datei aus.



```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Torsten>cd "\Program Files\PostgreSQL"

C:\Program Files\PostgreSQL>8.3\bin\psql.exe -d osm -h localhost -U postgres -W
-f c:\Users\Torsten\temp\osmpk.sql
```

Abbildung 9: Konsolen Screenshot Erzeugung Primary Keys auf den Datenbanktabellen

## OPTIMIERUNG DER OSM DATENBANK

Nach erfolgreichem Import sollten die Befehle *analyze* und *vacuum* auf der Datenbank ausgeführt werden. Diese Befehle können über das Administrationstool *pgAdmin III* an die Datenbank abgesetzt werden. Dazu muss eine Verbindung mit dem Datenbankserver hergestellt werden. Danach wird die OSM Datenbank selektiert. Über die Menüleiste *Werkzeuge/Abfragewerkzeug* wird eine SQL Konsole geöffnet, in der die Befehle an die Datenbank abgesetzt werden können.

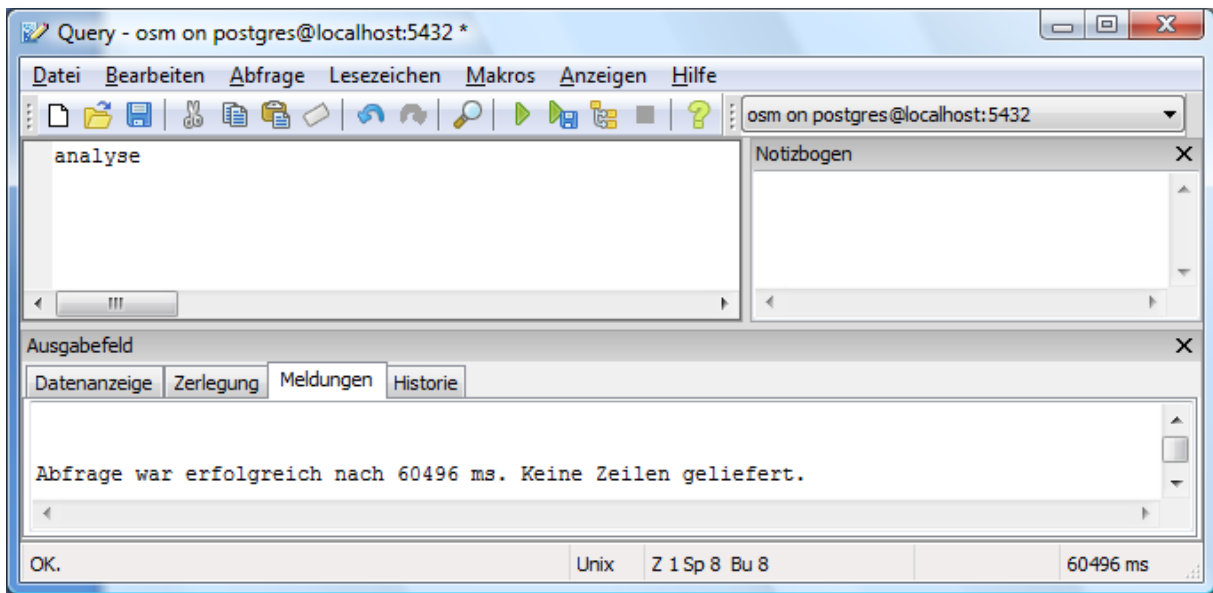


Abbildung 10: pgAdmin III Analyse Table

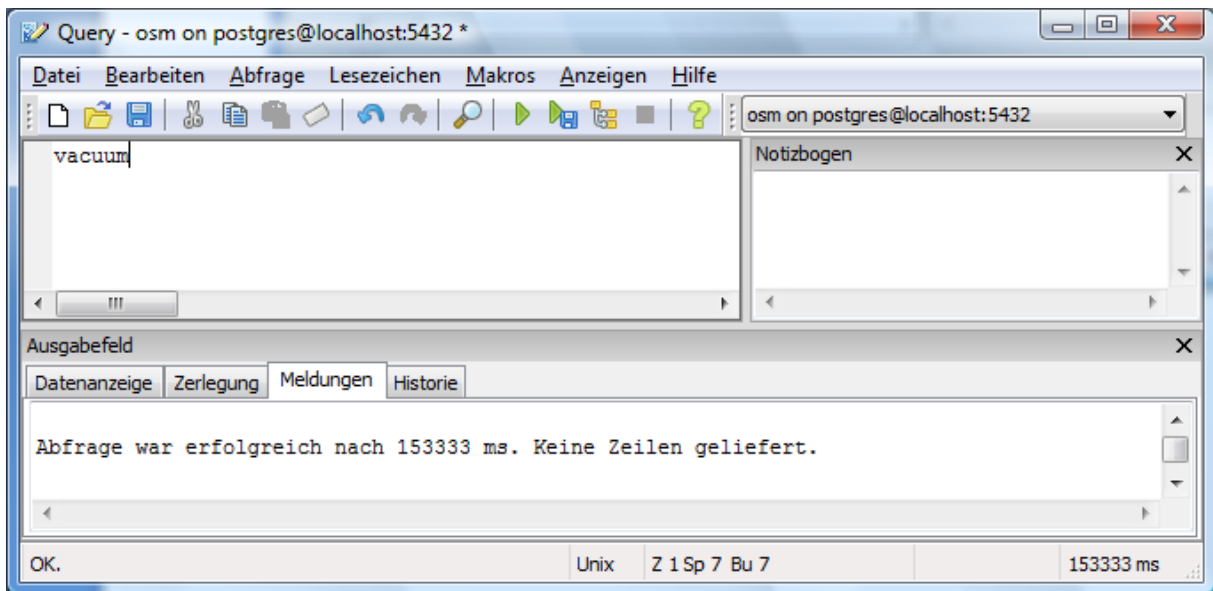


Abbildung 11: pgAdmin III Vacuum Table



## VERWALTUNG UND IMPORT - HÖHENLINIENDATEN

Die SRTM Höhendaten werden von der NASA im Internet auf Downloadservern angeboten. Das folgende Kapitel erläutert die Konvertierung der Höhendaten in Höhenlinien (Geometrische Objekte für die PostgreSQL Datenbank) und den Import der Höhenlinien in die lokale Datenbank.

### ERSTELLEN DER LOKALEN HÖHENLINIEN-DATENBANK

Das Erstellen der lokalen Höhenlinien-Datenbank erfolgt parallel zum vorhergehenden Kapitel. Als Name der Datenbank wird im Folgenden *srtm\_v2* angenommen.

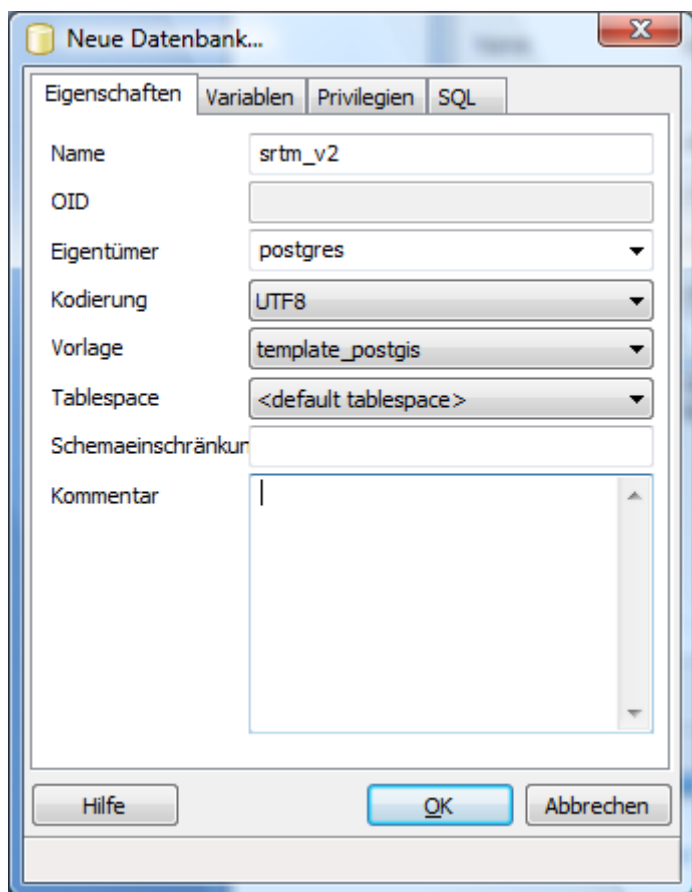


Abbildung 12: Erstellen einer neuen Höhenlinien-Datenbank

### NASA SRTM DATEN LIZENZ

Zitat: „...Die vom USGS<sup>16</sup> veröffentlichten Daten sind als Public Domain<sup>17</sup> zum Herunterladen kostenfrei verfügbar und dürfen uneingeschränkt verwendet werden: USGS-authored or produced data and information are in the public domain....“ (Quelle: <http://de.wikipedia.org/wiki/SRTM>)

<sup>16</sup> <http://de.wikipedia.org/wiki/USGS>

<sup>17</sup> [http://de.wikipedia.org/wiki/Public\\_Domain](http://de.wikipedia.org/wiki/Public_Domain)

## DOWNLOAD NASA SRTM DATEN

Ausgangsdaten sind komprimierte Höhendaten, die von der NASA auf ihren Download Servern angeboten werden. Diese Höhendaten wurden während des Flugs des Space Shuttle Endeavour im Februar 2000 durch ein speziell angepasstes Radarsystem an Board aufgezeichnet<sup>18,19</sup>. Die Version 2 des Datensatzes stellt eine überarbeitete Version des Rohdatensatzes der Space Shuttle Mission dar.

Die hier genutzten Höhendaten liegen in einer Auflösung von 3 Bogensekunden (30m Raster) vor. Die Daten können über die NASA Download Server<sup>20</sup> als Dateien, die in Länge und Breite ein Grad umfassen, heruntergeladen werden.

Für den Import in die lokale PostgreSQL Datenbank werden die vorhandenen Dateien innerhalb der BBOX [-5.0,40.0,18.0,55.0] analog zu den OSM Daten benötigt. Dazu müssen alle Dateien innerhalb der BBOX nach dem beschriebenen Muster<sup>21</sup> bestimmt und heruntergeladen werden.

## IMPORT DER HÖHENLINIEN

(Ausführliche Anleitung unter <http://wiki.openstreetmap.org/index.php/Contours> )

Die Höhendaten liegen aus dem SRTM Projekt in Form von Rasterdaten vor. Mit Hilfe der GDAL Bibliothek, werden Höhenlinien in 10m Höhenabstand aus diesen Rasterdaten berechnet. Diese Höhenlinien werden als geometrische Objekte (Multiline) in der PostgreSQL Datenbank gespeichert.

Zur Umrechnung der Rasterdaten in Höhenlinien werden die folgenden zusätzlichen Komponenten benötigt.

---

### GDAL - GEOSPATIAL DATA ABSTRACTION LIBRARY

Das GDAL Programmpaket lässt sich unter dem folgenden Link herunterladen:

<http://download.osgeo.org/gdal/win32/1.5/gdalwin32exe150.zip>

Das ZIP Archiv muss entpackt werden, die Verzeichnisstruktur kann dann z.B. nach `C:\Program Files\gdalwin32-1.5\` kopiert werden.

Nach dem Kopieren der Dateien sind die folgenden Systemvariablen anzulegen oder zu erweitern:

- `GDAL_DATA`: Verweist auf den `data` Unterordner der GDAL Installation.
- `PATH`: Erweitern um den Verweis auf den `bin` Unterordner der GDAL Installation

---

### SKRIPTE

Für den Import und die Berechnung der Höhenlinien werden zwei Shell Skripte benötigt. Diese UNIX Shell Skripte sind ohne weitere Anpassung auch unter Windows Vista in einer Cygwin<sup>22</sup> Bash Shell lauffähig.

---

<sup>18</sup> Quelle: <http://www2.jpl.nasa.gov/srtm/>

<sup>19</sup> Quelle: <http://de.wikipedia.org/wiki/SRTM>

<sup>20</sup> <ftp://e0srp01u.ecs.nasa.gov/srtm/version2/SRTM3/Eurasia/>

<sup>21</sup> [ftp://e0srp01u.ecs.nasa.gov/srtm/version2/Documentation/Continent\\_def.gif](ftp://e0srp01u.ecs.nasa.gov/srtm/version2/Documentation/Continent_def.gif)

### srtm\_generate\_hdr.sh

Das Skript extrahiert aus der jeweiligen HGT Datei unter anderem die GeoTIFF Datei, die das Raster mit den Höheninformationen enthält.

Das vollständige Skript kann unter der folgenden URL heruntergeladen und unverändert genutzt werden:  
[http://mpa.itc.it/rs/srtm/srtm\\_generate\\_hdr.sh](http://mpa.itc.it/rs/srtm/srtm_generate_hdr.sh)

### srtm\_import.sh

Dieses Skript ruft für jede HGT Datei das Skript *srtm\_generate\_hdr.sh* auf und berechnet dann mit Hilfe der GDAL Bibliothek die Höhenlinien. Die Höhenlinien werden in einer temporären SHAPE Datei zwischengespeichert, welche zum Schluss mit dem Befehl *shp2pgsql* (Bestandteil der PostgreSQL/PostGIS Installation) in die PostgreSQL Datenbank importiert wird.

```
#!/bin/bash
PREP_TABLE="1"
for X in *.hgt.zip; do
    yes | ./srtm_generate_hdr.sh $X
    rm -f "${X%%.zip}"
    # Import 10m contours
    rm -f "${X%%.hgt.zip}.shp" "${X%%.hgt.zip}.shx" "${X%%.hgt.zip}.dbf"
    gdal_contour -i 10 -snodata 32767 -a height "${X%%.hgt.zip}.tif" "${X%%.hgt.zip}.shp"
    [ "$PREP_TABLE" ] && shp2pgsql -p -s 4326 -l -g way "${X%%.hgt.zip}" contours | psql -q
    -U postgres -h dell srtm_v2
    shp2pgsql -a -s 4326 -g way "${X%%.hgt.zip}" contours | psql -q -U postgres -h dell srtm_v2
    rm -f "${X%%.hgt.zip}.shp" "${X%%.hgt.zip}.shx" "${X%%.hgt.zip}.dbf"
    rm -f "${X%%.hgt.zip}.bil"
    rm -f "${X%%.hgt.zip}.hdr"
    rm -f "${X%%.hgt.zip}.prj"
    rm -f "${X%%.hgt.zip}.tif"
    unset PREP_TABLE
done
```

Die notwendigen Tabellen in der Datenbank werden bei Bedarf automatisch erstellt. Anpassungen sind evtl. in der Parameterleiste des Befehls *psql* (im Skript hervorgehoben) notwendig.

## IMPORT

Die beiden Skripte sollten in das lokale Verzeichnis der NASA Höhendaten kopiert werden. Danach wird in einer BASH Shell das Skript *srtm\_import.sh* aufgerufen. Das Skript läuft über alle vorhandenen Dateien mit der Endung *\*.hgt.zip* und versucht diese zu importieren.

## OPTIMIERUNG DER SRTM DATENBANK

Nach erfolgreichem Import sollten auch auf dieser Datenbank die Befehle *analyze* und *vacuum* ausgeführt werden (siehe Kapitel *Import der OSM Daten mittels osm2pgsql*).

<sup>22</sup> <http://www.cygwin.com/>

## INSTALLATION MAPNIK

Der Mapnik Renderer besteht aus einer Menge von C/C++ Bibliotheken und einem Python API zur Ansteuerung des Renderprozesses. Im Folgenden wird die Installation der notwendigen Komponenten beschrieben.

### INSTALLATION PYTHON INTERPRETER

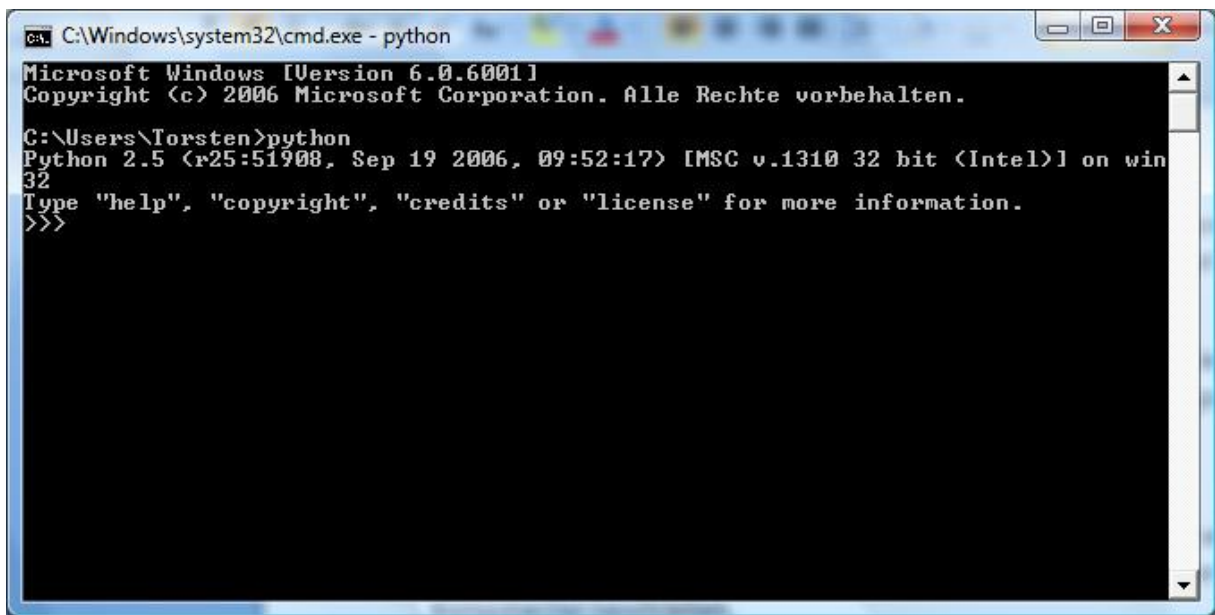
Python kann auf der offiziellen Python Homepage<sup>23</sup> als Windows Installer Package heruntergeladen werden. Die Installation muss mit Administratorenrechten ausgeführt werden und ist selbsterklärend.

Hinweis: Python kann im Gegensatz zum Mapnik Renderer an beliebiger Stelle installiert werden, auch wenn es dazu abweichende Aussagen im Netz gibt.

Nach dem erfolgreichen Abschluss der Installation müssen die folgenden Systemvariablen angelegt oder erweitert werden:

- *PATH:* Erweitern um Verweis auf den Python Installationsordner (z.B. *C:\Python25*)

Um den Erfolg der Installation zu testen, kann in einer Eingabeaufforderung der Python Interpreter testweise aufgerufen werden. Dazu eine Eingabeaufforderung (*cmd.exe*) öffnen und den Befehl *python* absetzen. (Evtl. ist vorher ein Neustart des Rechners notwendig)



```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Torsten>python
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

Abbildung 13: Eingabeaufforderung mit Python Interpreter

Hinweis: Mit der Tastenkombination *STRG + Z* kann der Interpreter beendet werden.

## INSTALLATION MAPNIK

Mapnik kann von der offiziellen Mapnik Homepage<sup>24</sup> für die Windows Plattform als ZIP Archiv herunter geladen werden. Ein Windows Installer Package wie es für die Installation von Python im vorherigen Kapitel verwendet wurde, wird nicht angeboten.

<sup>23</sup> <http://www.python.org/>



## INSTALLATION UND KONFIGURATION GNAVIGIA OSM-BACKGROUNDPROVIDER

Der OSM Backgroundprovider wird in Form einer DLL *OsmBackgroundProvider.dll* mit der GNavigia Installation ausgeliefert. Die Anbindung und Verwendung des COM Servers in GNavigia wird im folgenden Kapitel *Zugriff auf OSM Backgroundprovider in GNavigia* ausführlich beschrieben.

### FUNKTIONSBESCHREIBUNG

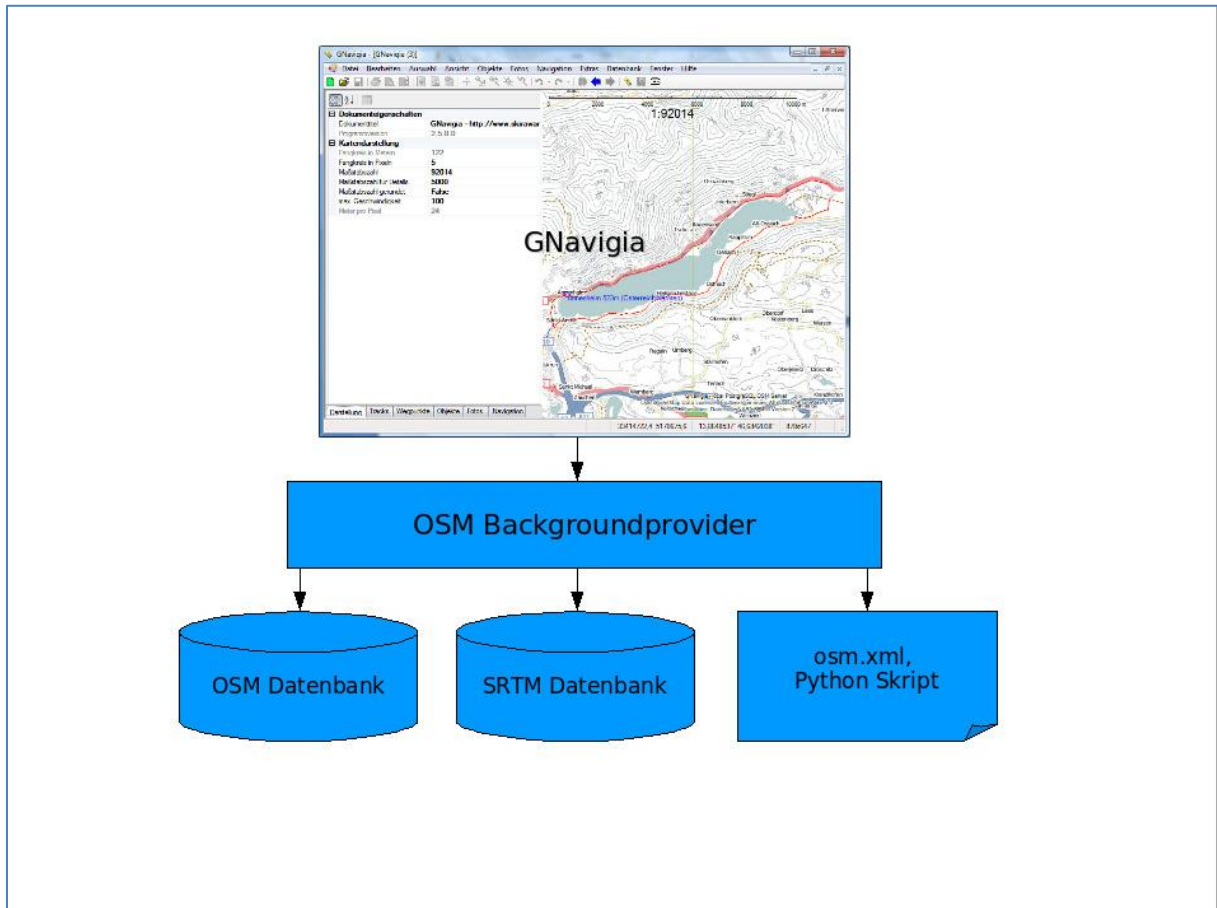


Abbildung 15: Schematischer Aufbau - GNavigia OSM Backgroundprovider

Nach jeder Aktion, die den Kartenausschnitt ändert, fragt GNavigia mit einer geringen Verzögerung den OSM Backgroundprovider nach einem neuen Kartenhintergrundbild.

Dazu übergibt der OSM Backgroundprovider folgende Parameter der Anfrage an das Python Skript:

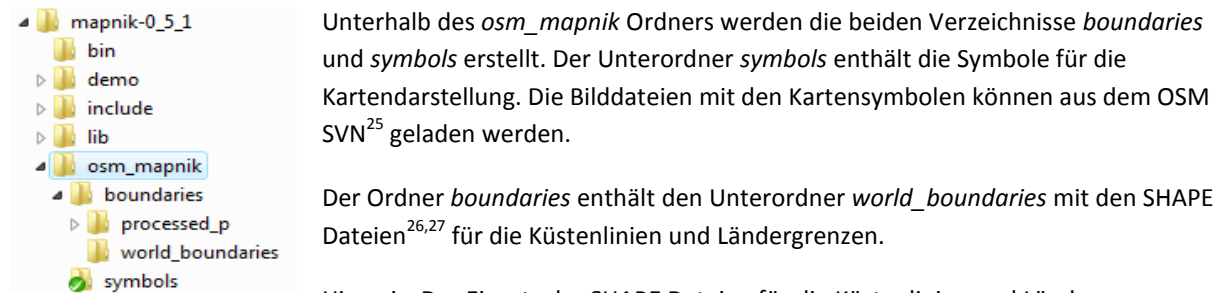
- Die Bounding Box des aktuellen Bildausschnittes in UTM Koordinaten,
- die geforderte Kartenbildgröße in Pixel,
- das geforderte Kartenprojektionssystem (z.B. UTM32, UTM33) und
- das Verzeichnis, in das die Datei mit dem Kartenhintergrundbild abgelegt werden soll. (Hierfür wird die Setzung aus der Umgebungsvariable `%TEMP%` des Benutzers genutzt.)

Das Python Skript wird durch den OSM Backgroundprovider als Konsolenprozess gestartet, der eine asynchrone Abarbeitung des Renderprozesses ermöglicht. Ist der Renderprozess beendet wird an den OSM Backgroundprovider der Dateiname des produzierten Kartenhintergrundbildes zurückgeliefert. Der Backgroundprovider lädt danach diese Datei und liefert das Kartenhintergrundbild an GNavigia zurück.

## INSTALLATION

### VERZEICHNISSTRUKTUR

Sämtliche Konfigurationsdaten, Shape Dateien und Symbole für den Mapnik Renderer werden in einem zentralen Ordner, *osm\_mapnik*, abgelegt. Dieser Ordner kann z.B. unterhalb der Mapnik Installation *C:\mapnik-0\_5\_1* erstellt werden.



Hinweis: Der Einsatz der SHAPE Dateien für die Küstenlinien und Ländergrenzen verschlechtert die Performance dramatisch. (Erstellungszeit für den Kartenausschnitt aus dem Eingangsbeispiel mit SHAPE Daten 12s ohne SHAPE Daten 2s!) Wenn nicht unbedingt benötigt, sollte auf den Einsatz der SHAPE Dateien verzichtet werden. In der folgenden Beschreibung wird trotzdem davon ausgegangen, dass die SHAPE Dateien eingebunden werden.

### PYTHON SKRIPT GENERATE\_GNAVIGIA\_IMAGE

Zur Ansteuerung des Mapnik Renderprozesses wird ein Python Skript benötigt. Dazu wird im Ordner *osm\_mapnik* die Datei *generate\_gnavigia\_image.py* angelegt und mit dem im Folgenden abgedruckten Python Skript gefüllt.

```
#!/usr/bin/python

from mapnik import *
import sys, os

if __name__ == "__main__":
    bboxSplitStringArr = sys.argv[1].split(',')
    ll = ( float(bboxSplitStringArr[0].strip()),
          float(bboxSplitStringArr[1].strip()),
          float(bboxSplitStringArr[2].strip()),
          float(bboxSplitStringArr[3].strip())

    # Image width
    imgx = int(sys.argv[2])

    # Image height
    imgy = int(sys.argv[3])
```

<sup>25</sup> <http://svn.openstreetmap.org/applications/rendering/mapnik/symbols/>

<sup>26</sup> [http://tile.openstreetmap.org/world\\_boundaries-spherical.tgz](http://tile.openstreetmap.org/world_boundaries-spherical.tgz)

<sup>27</sup> [http://hypercube.telascience.org/~kleptog/processed\\_p.zip](http://hypercube.telascience.org/~kleptog/processed_p.zip)

```

# Image URI
map_uri = sys.argv[4]

# Proj String
proj = sys.argv[5]

m = Map(imgx,imgy,proj)
load_map(m,"osm_mapnik/osm.xml")
bbox = Envelope(ll[0],ll[1],ll[2],ll[3])
m.zoom_to_box(bbox)
im = Image(imgx,imgy)
render(m, im)
view = im.view(0,0,imgx,imgy) # x,y,width,height
view.save(map_uri,'png')

```

---

## OSM XML DATEI

Die aktuelle *osm.xml* Datei kann ebenfalls aus dem OSM SVN<sup>28</sup> geladen werden und muss nach den folgenden Vorgaben angepasst werden:

1. Im XML Tag <MAP> definiert das Attribut *srs* das Projektionssystem des Kartenbildes. Diese Attribut wird entfernt, da das Projektionssystem in Abhängigkeit von der Kartenanfrage (UTM Streifen) im Python Skript gesetzt wird.
2. Im XML Tag <MAP> die Hintergrundfarbe (Attribute *bgcolor*) auf weiß setzen *bgcolor="#ffffff"*.
3. Sämtliche Pfade zu dem Verzeichnis mit den Symboldateien *c:/mapnik/symbols/* durch *osm\_mapnik/symbols/* ersetzen. (z.B. *c:/mapnik/symbols/halt.png* -> *osm\_mapnik/symbols/halt.png*)
4. Sämtliche Pfade zu den Unterverzeichnissen mit den SHAPE Dateien *c:/boundaries/* durch *osm\_mapnik/boundaries/* ersetzen.

### Beispiel:

```

<Layer name="world" status="on" srs="+proj=merc +datum=WGS84 +over">
  <StyleName>world</StyleName>
  <Datasource>
    <Parameter name="type">shape</Parameter>
    <Parameter name="file">osm_mapnik/boundaries/world_boundaries/world_bnd_m</Parameter>
  </Datasource>
</Layer>

```

5. Sämtliche XML Tags <Layer>, die Verbindungsparameter auf die PostgreSQL Datenbank enthalten, anpassen. (Der Parameter *extent* beschreibt die Bounding Box der importierten Daten z.B. *<Parameter name="extent">-5.0,40.0,18.0,55.0</Parameter>*)
6. An sämtlichen Layern, die auf die PostgreSQL Datenbank verweisen, das bestehende XML Attribut *srs="..."* durch *srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs"* ersetzen. (Die Daten wurden beim Import in WGS84 umgerechnet.)

### Beispiel:

```

<Layer name="leisure" status="on" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
  <StyleName>leisure</StyleName>
  <Datasource>
    <Parameter name="type">postgis</Parameter>

```

---

<sup>28</sup> <http://svn.openstreetmap.org/applications/rendering/mapnik/osm.xml>



```

<Parameter name="host">localhost</Parameter>
<Parameter name="port">5432</Parameter>
<Parameter name="user">postgres</Parameter>
<Parameter name="password">postgres</Parameter>
<Parameter name="dbname">osm</Parameter>
<Parameter name="table">(select * from planet_osm_polygon order by z_order,way_area desc)
as leisure</Parameter>
<Parameter name="estimate_extent">>false</Parameter>
<Parameter name="extent">-5.0,40.0,18.0,55.0</Parameter>
</Datasource>
</Layer>

```

**Hinweis:** Je nach Datenbankkonfiguration ist evtl. die Angabe des Passwortes (Parameter *password*) nicht notwendig.

7. Für die Darstellung der Höhenlinien müssen die folgenden zusätzlichen XML Style Tags in die OSM XML Datei eingefügt werden.

```

<!-- SRTM NASA Datensatz Contour -->
<Style name="contours10">
  <Rule>
    <MaxScaleDenominator>51185</MaxScaleDenominator>
    <MinScaleDenominator>1599</MinScaleDenominator>
    <LineStyle>
      <CssParameter name="stroke">#9cb197</CssParameter>
      <CssParameter name="stroke-width">0.5</CssParameter>
    </LineStyle>
  </Rule>
</Style>
<Style name="contours50">
  <Rule>
    <MaxScaleDenominator>204741</MaxScaleDenominator>
    <MinScaleDenominator>51185</MinScaleDenominator>
    <LineStyle>
      <CssParameter name="stroke">#9cb197</CssParameter>
      <CssParameter name="stroke-width">0.6</CssParameter>
    </LineStyle>
  </Rule>
  <Rule>
    <MaxScaleDenominator>51185</MaxScaleDenominator>
    <MinScaleDenominator>1599</MinScaleDenominator>
    <LineStyle>
      <CssParameter name="stroke">#747b90</CssParameter>
      <CssParameter name="stroke-width">0.6</CssParameter>
    </LineStyle>
  </Rule>
</Style>
<Style name="contours100">
  <Rule>
    <MaxScaleDenominator>409483</MaxScaleDenominator>
    <MinScaleDenominator>204741</MinScaleDenominator>
    <LineStyle>
      <CssParameter name="stroke">#9cb197</CssParameter>
      <CssParameter name="stroke-width">0.7</CssParameter>
    </LineStyle>
  </Rule>
</Style>

```

```

<Rule>
  <MaxScaleDenominator>204741</MaxScaleDenominator>
  <MinScaleDenominator>51185</MinScaleDenominator>
  <LineStyle>
    <CssParameter name="stroke">#747b90</CssParameter>
    <CssParameter name="stroke-width">0.7</CssParameter>
  </LineStyle>
</Rule>
<Rule>
  <MaxScaleDenominator>51185</MaxScaleDenominator>
  <MinScaleDenominator>1599</MinScaleDenominator>
  <LineStyle>
    <CssParameter name="stroke">#855d62</CssParameter>
    <CssParameter name="stroke-width">0.7</CssParameter>
  </LineStyle>
</Rule>
</Style>
<Style name="contours-text50">
  <Rule>
    <MaxScaleDenominator>51185</MaxScaleDenominator>
    <MinScaleDenominator>1599</MinScaleDenominator>
    <TextSymbolizer name="height" face_name="DejaVu Sans Book" size="8" fill="#747b90"
halo_radius="1" placement="line" />
  </Rule>
</Style>
<Style name="contours-text100">
  <Rule>
    <MaxScaleDenominator>102370</MaxScaleDenominator>
    <MinScaleDenominator>51185</MinScaleDenominator>
    <TextSymbolizer name="height" face_name="DejaVu Sans Book" size="8" fill="#747b90"
halo_radius="1" placement="line" />
  <Rule>
    <MaxScaleDenominator>51185</MaxScaleDenominator>
    <MinScaleDenominator>1599</MinScaleDenominator>
    <TextSymbolizer name="height" face_name="DejaVu Sans Book" size="8" fill="#855d62"
halo_radius="1" placement="line" />
  </Rule>
</Style>

```

8. Um die Datenquellen (SRTM PostgreSQL Datenbank) dem Mapnik Renderprozess bekannt zu geben, müssen die folgenden XML Layer Tags hinzugefügt werden.

```

<!-- SRTM NASA Datensatz Contour -->
<Layer name="srtm_10" status="on" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
  <StyleName>contours10</StyleName>
  <StyleName>contours-text10</StyleName>
  <Datasource>
    <Parameter name="type">postgis</Parameter>
    <Parameter name="host">localhost</Parameter>
    <Parameter name="port">5432</Parameter>
    <Parameter name="user">postgres</Parameter>
    <Parameter name="password">postgres</Parameter>
    <Parameter name="dbname">srtm_v2</Parameter>
    <Parameter name="table">(select way,height from contours WHERE height::integer % 10 = 0 AND
height::integer % 50 != 0 AND height::integer % 100 != 0) as "contours-10"</Parameter>
    <Parameter name="estimate_extent">>false</Parameter>
    <Parameter name="extent">-5.0,40.0,18.0,55.0</Parameter>

```

```

</Datasource>
</Layer>
<Layer name="srtm_50" status="on" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
  <StyleName>contours50</StyleName>
  <StyleName>contours-text50</StyleName>
  <Datasource>
    <Parameter name="type">postgis</Parameter>
    <Parameter name="host">localhost</Parameter>
    <Parameter name="port">5432</Parameter>
    <Parameter name="user">postgres</Parameter>
    <Parameter name="password">postgres</Parameter>
    <Parameter name="dbname">srtm_v2</Parameter>
    <Parameter name="table">(select way,height from contours WHERE height::integer % 50 = 0 AND
height::integer % 100 != 0) as "contours-50"</Parameter>
    <Parameter name="estimate_extent">>false</Parameter>
    <Parameter name="extent">-5.0,40.0,18.0,55.0</Parameter>
  </Datasource>
</Layer>
<Layer name="srtm_100" status="on" srs="+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs">
  <StyleName>contours100</StyleName>
  <StyleName>contours-text100</StyleName>
  <Datasource>
    <Parameter name="type">postgis</Parameter>
    <Parameter name="host">localhost</Parameter>
    <Parameter name="port">5432</Parameter>
    <Parameter name="user">postgres</Parameter>
    <Parameter name="password">postgres</Parameter>
    <Parameter name="dbname">srtm_v2</Parameter>
    <Parameter name="table">(select way,height from contours WHERE height::integer % 100 = 0) as
"contours-100"</Parameter>
    <Parameter name="estimate_extent">>false</Parameter>
    <Parameter name="extent">-5.0,40.0,18.0,55.0</Parameter>
  </Datasource>
</Layer>

```

**Hinweis:** Je nach Datenbankkonfiguration ist evtl. die Angabe des Passwortes (Parameter *password*) nicht notwendig.

**Hinweis:** Die Reihenfolge der XML Layer Tags beschreibt die Reihenfolge der Layer im Kartenhintergrundbild. Das erste XML Layer Tag beschreibt den untersten Layer im Kartenhintergrundbild. Um die Höhenlinien nicht über die Straßendarstellung oder die Gewässer zu zeichnen, sollte der Layer direkt nach dem ersten Layer *leisure* eingefügt werden. Der Layer *leisure* beschreibt Freizeitflächen z.B. wie Wälder oder Parks.

## KONFIGURATION

Zur Konfiguration des OSM Backgroundproviders sind die im folgenden Kapitel beschriebenen Schritte notwendig.

## UMGEBUNGSVARIABLEN

Der OSM Backgroundprovider benötigt zur Funktion die Umgebungsvariable *GNAVIGIA\_OSM\_PATH*. Diese Umgebungsvariable verweist auf den Ordner *osm\_mapnik* der neben einen Python Skript zur Ansteuerung von Mapnik auch die vorhandenen Kartenlayer, die Datenquellen und die Ausgestaltungsregeln des Kartenbildes der aktuellen Konfiguration enthält.

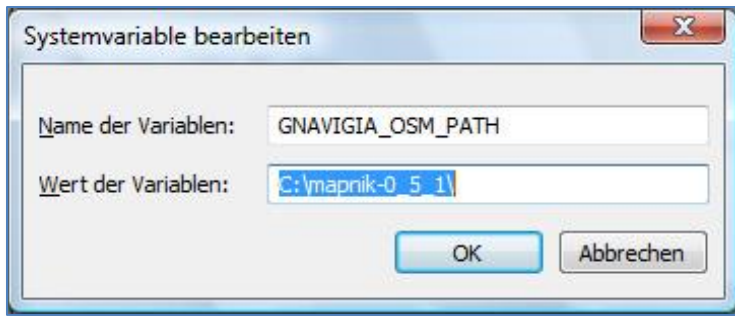


Abbildung 16: Umgebungsvariable GNAVIGIA\_OSM\_PATH

Diese Umgebungsvariable kann als Systemvariable für alle Benutzer des Rechners oder als Benutzervariable für den jeweiligen angemeldeten Benutzer definiert werden.

## ZUGRIFF AUF OSM BACKGROUNDPROVIDER IN GNAVIGIA

Um den Zugriff auf den COM Server *OSM Backgroundprovider* für GNavigia zu aktivieren, muss der Service für die Hintergrundbilder konfiguriert werden. Dazu wird der Dialog *Service für Hintergrundbilder* über den Menüeintrag *Ansicht / Server der Hintergrundbilder auswählen...* geöffnet.

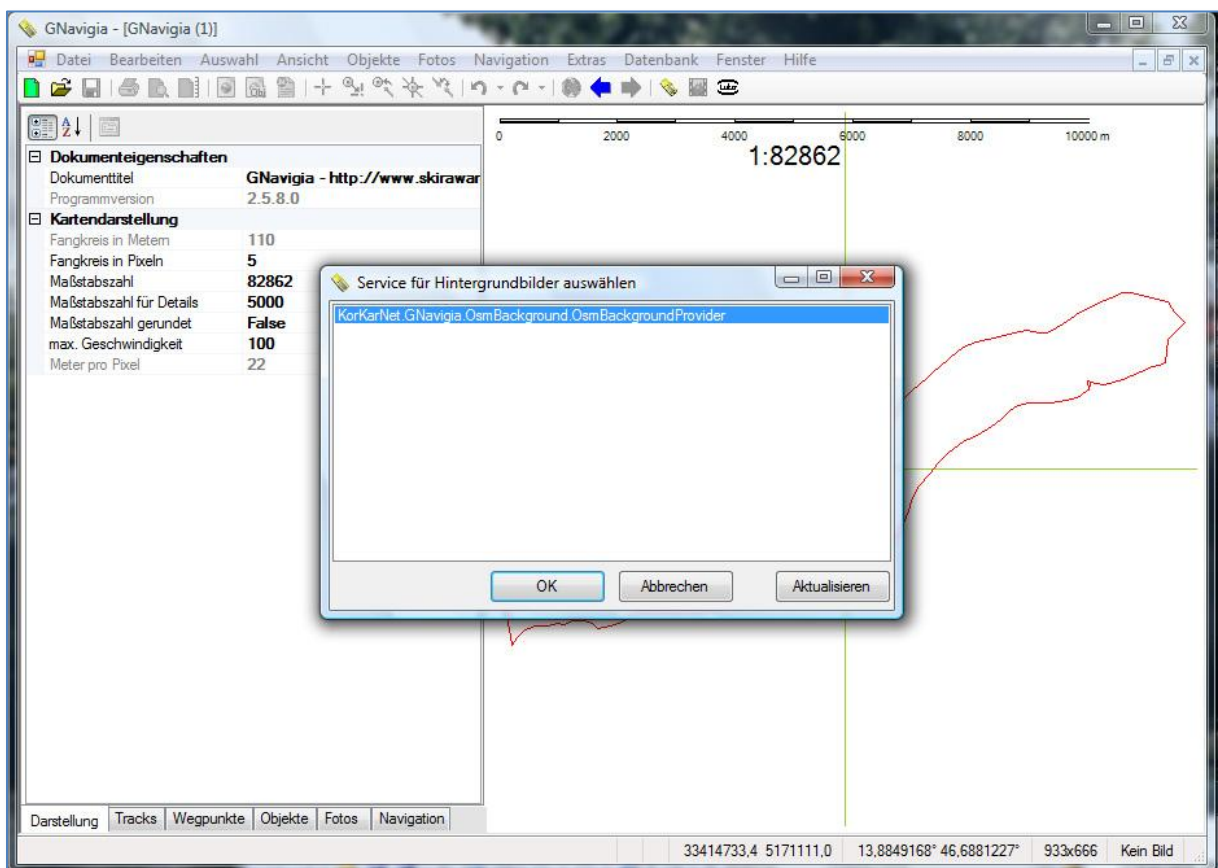


Abbildung 17: Dialog Service für Hintergrundbilder auswählen

Mit dem Öffnen des Dialogs sucht GNavigia sämtliche, auf das Interface für Hintergrundbilder registrierte COM Server und listet diese auf. Für die Darstellung der OSM Kartendaten wird der Eintrag *KorKarNet.GNavigia.OsmBackground.OsmBackgroundProvider* im Dialog markiert und der Dialog durch betätigen der Schaltfläche *OK* geschlossen.

Um die Darstellung der Hintergrundbilder nach Kartenoperationen zu aktivieren wird im Menü *Ansicht* der Eintrag *Hintergrundbild darstellen* aktiviert.

**Hinweis:** Die Aktivierung des Eintrags im Menü ist nur möglich, wenn Daten in GNavigia geladen wurden und in mindestens einem Fenster dargestellt werden. Ansonsten ist der Menüpunkt ausgegraut.

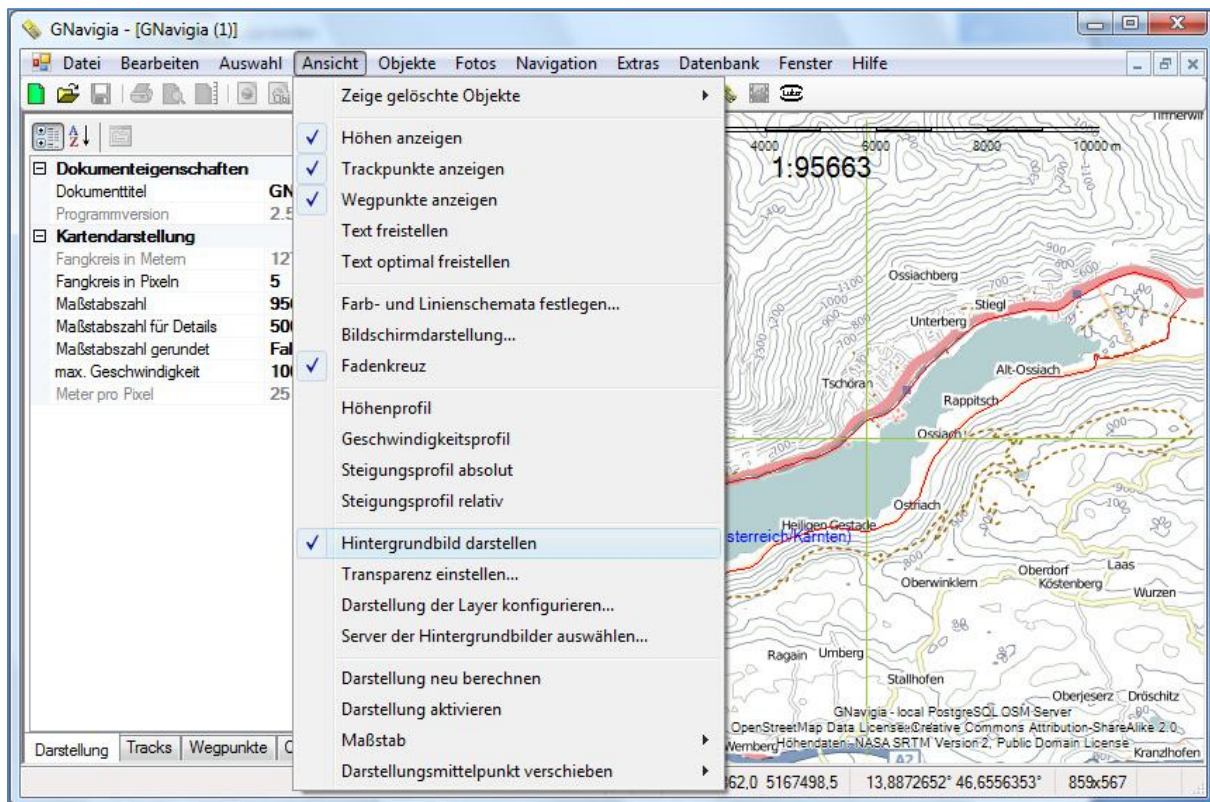


Abbildung 18: Aktivierung der Darstellung von Hintergrundbildern

Über den Menüpunkt *Transparenz einstellen...* im Menü *Ansicht*, kann die Hintergrundkarte in ihrer Darstellungsintensität abgeschwächt werden.

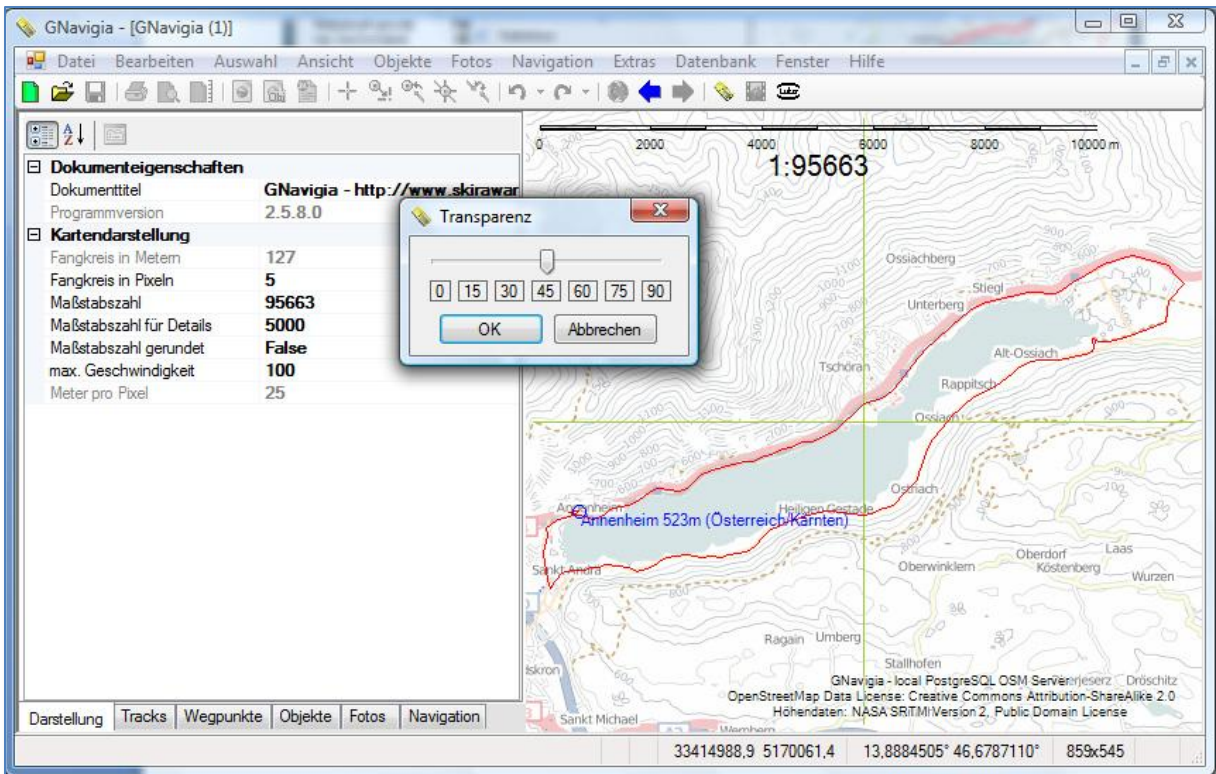


Abbildung 19: Darstellung Hintergrundbild mit Transparenz

## LOGGING

Der OSM Backgroundprovider benutzt zur Protokollierung von Fehlermeldungen die von GNavigia zur Verfügung gestellte Log4Net Umgebung. Die Konfiguration wird im zugehörigen Hauptprogramm vorgenommen. Folgerichtig ist dieser Schritt in der Dokumentation von GNavigia beschrieben.

## ABBILDUNGSVERZEICHNIS

Abbildung 1: Beispieldarstellung einer Radtour mit OSM/SRTM Hintergrunddaten .....	4
Abbildung 2: Mapnik Hintergrundkarte mit OSM Daten und Höhenlinien .....	5
Abbildung 3: Teilkartenbild OpenStreetMap Datensatz (Stand 12/2008) .....	6
Abbildung 4: Teilkartenbild Höhenlinien aus NASA SRTM Datensatz .....	7
Abbildung 5: Datenbank Administrationstool .....	9
Abbildung 6: Erstellen einer neuen OSM Datenbank .....	10
Abbildung 7: Konsolen Screenshot Installation Intarray Contrib Module .....	11
Abbildung 8: Konsolen Screenshot eines erfolgreichen Imports .....	14
Abbildung 11: Konsolen Screenshot Erzeugung Primary Keys auf den Datenbanktabellen .....	15
Abbildung 9: pgAdmin III Analyse Table .....	16
Abbildung 10: pgAdmin III Vacuum Table .....	16
Abbildung 12: Erstellen einer neuen Höhenlinien-Datenbank .....	17
Abbildung 13: Eingabeaufforderung mit Python Interpreter .....	20
Abbildung 14: Ergebnisbild Mapnik Funktionstest .....	21
Abbildung 15: Schematischer Aufbau - GNavigia OSM Backgroundprovider .....	22
Abbildung 16: Umgebungsvariable GNAVIGIA_OSM_PATH .....	28
Abbildung 17: Dialog Service für Hintergrundbilder auswählen .....	28
Abbildung 18: Aktivierung der Darstellung von Hintergrundbildern .....	29
Abbildung 19: Darstellung Hintergrundbild mit Transparenz .....	30