



# Quantum GIS

Handbuch zur  
Benutzung und Installation

Version 0.9.1 '*Ganymede*'

## Präambel

Dieses Werk ist das offizielle Handbuch zur Benutzung und Installation der Software Quantum GIS. Die in diesem Werk genannten Soft- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen. Quantum GIS ist unter der GNU General Public License veröffentlicht. Weitere Informationen finden Sie auf der Quantum GIS Homepage <http://www.qgis.org>.

Die in diesem Werk enthaltenen Angaben, Daten, Ergebnisse usw. wurden von den Autoren nach bestem Wissen erstellt und mit Sorgfalt überprüft. Dennoch sind inhaltliche Fehler nicht völlig auszuschließen. Daher erfolgen alle Angaben ohne jegliche Verpflichtung oder Garantie. Die Autoren und Herausgeber übernehmen aus diesem Grund auch keinerlei Verantwortung oder Haftung für Fehler und deren Folgen. Hinweise auf eventuelle Irrtümer werden gerne entgegengenommen.

Dieses Dokument wurde mit  $\text{\LaTeX}$  gesetzt. Es ist als  $\text{\LaTeX}$ -Quelltext erhältlich, und kann online als HTML und PDF Dokument unter <http://www.qgis.org> heruntergeladen werden.

Das englischsprachige Original und weitere in andere Sprachen übersetzte Versionen dieses Handbuchs können aus dem Dokumentationsbereich des Quantum GIS Projektes heruntergeladen werden. Weitere Informationen über dieses Dokument und Hilfestellungen für die Übersetzung in weitere Sprachen finden Sie im QGIS-Wiki unter:

<http://wiki.qgis.org/qgiswiki/DocumentationWritersCorner>.

### **Autoren des englischsprachigen Handbuchs:**

Gary E. Sherman, Tim Sutton, Radim Blazek, Stephan Holl, Otto Dassau, Tyler Mitchell, Brendan Morely, Lars Luthman, Godofredo Contreras, Magnus Homann, Martin Dobias, David Willis und Jürgen E. Fischer

Herzlichen Dank an Tisham Dhar für die Vorbereitung der MSYS-Umgebung, an Tom Elwertowski und William Kyngesburye für ihre Hilfe bei den MAC OSX Kapiteln und an Tara Athan für die Überarbeitung und Kontrolle.

### **Übersetzung ins Deutsche:**

Otto Dassau, Stephan Holl und Sascha Wernicke.

**Copyright © 2004 - 2008 Quantum GIS Project**

**Internet:** <http://www.qgis.org>

# Inhaltsverzeichnis

<b>Titelseite</b>	<b>i</b>
<b>Präambel</b>	<b>ii</b>
<b>Inhaltsverzeichnis</b>	<b>iii</b>
<b>Abbildungsverzeichnis</b>	<b>x</b>
<b>Tabellenverzeichnis</b>	<b>xii</b>
<b>1 Vorwort</b>	<b>1</b>
1.1 Funktionalitäten . . . . .	1
1.2 Was ist neu in 0.9 . . . . .	2
<b>2 Einführung in GIS</b>	<b>4</b>
2.1 Warum ist das alles so neu? . . . . .	5
2.1.1 Rasterdaten . . . . .	5
2.1.2 Vektordaten . . . . .	6
<b>3 Getting Started</b>	<b>7</b>
3.1 Installation . . . . .	7
3.2 Beispieldaten . . . . .	7
3.3 QGIS starten . . . . .	8
3.3.1 Optionen der Kommandozeile . . . . .	8
3.4 QGIS GUI . . . . .	10
3.4.1 Menüleiste . . . . .	11
3.4.2 Werkzeugleiste . . . . .	12
3.4.3 Kartenlegende . . . . .	12
3.4.4 Kartenfenster . . . . .	14
3.4.5 Kartenübersicht . . . . .	14
3.4.6 Statusleiste . . . . .	15
3.5 Daten anzeigen . . . . .	15
3.5.1 Maßstabsabhängige Datenanzeige . . . . .	15
3.5.2 Datenanzeige kontrollieren . . . . .	16
3.6 Messen . . . . .	16
3.6.1 Strecken messen . . . . .	17
3.6.2 Flächen messen . . . . .	17
3.7 Projekte managen . . . . .	17
3.8 GUI Optionen . . . . .	18
3.9 Räumliche Lesezeichen . . . . .	20
3.9.1 Ein Lesezeichen erstellen . . . . .	20
3.9.2 Arbeiten mit Lesezeichen . . . . .	20

---

3.9.3	Auf ein Lesezeichen zoomen . . . . .	20
3.9.4	Ein Lesezeichen löschen . . . . .	20
<b>4</b>	<b>Arbeiten mit Vektordaten</b>	<b>21</b>
4.1	ESRI Shapes . . . . .	21
4.1.1	Shape laden . . . . .	21
4.1.2	Darstellungsgeschwindigkeit optimieren . . . . .	23
4.1.3	MapInfo Datei laden . . . . .	24
4.1.4	ArcInfo Coverage laden . . . . .	24
4.2	PostGIS-Ebene laden . . . . .	24
4.2.1	Erstellen einer Verknüpfung . . . . .	25
4.2.2	PostGIS-Ebene laden . . . . .	25
4.2.3	Einige Details zu PostGIS-Ebenen . . . . .	26
4.2.4	PostGIS-Ebene importieren . . . . .	26
4.2.5	Geschwindigkeit optimieren . . . . .	28
4.3	Dialog für Ebeneneigenschaften . . . . .	29
4.3.1	Bezeichnungen . . . . .	29
4.3.2	Allgemein . . . . .	31
4.3.3	Metadaten . . . . .	31
4.3.4	Beschriftungen . . . . .	31
4.3.5	Aktionen auf Basis von Attributen . . . . .	32
4.4	Editieren . . . . .	35
4.4.1	Einstellen der Snapping-Toleranz . . . . .	36
4.4.2	Editieren einer existierenden Ebene . . . . .	36
4.4.3	Neue Vektorlayer erstellen . . . . .	42
4.5	Query Builder . . . . .	43
4.5.1	PostGIS-Ebenen abfragen . . . . .	44
4.5.2	OGR- und GRASS-Ebenen abfragen . . . . .	44
<b>5</b>	<b>Arbeiten mit Rasterdaten</b>	<b>45</b>
5.1	Was ist eine Rasterdatei? . . . . .	45
5.2	Durch QGIS unterstützte Rasterformate . . . . .	45
5.3	Rasterdaten in QGIS laden . . . . .	46
5.4	Eigenschaften von Rasterdaten . . . . .	46
5.4.1	Bezeichnungen . . . . .	47
5.4.2	Allgemein . . . . .	48
5.4.3	Metadaten . . . . .	49
5.4.4	Pyramiden . . . . .	49
5.4.5	Histogramm . . . . .	49
<b>6</b>	<b>Mit OGC-Daten arbeiten</b>	<b>50</b>
6.1	Was sind OGC-Daten? . . . . .	50
6.2	WMS-Klient . . . . .	50

---

6.2.1	Übersicht über die WMS-Unterstützung . . . . .	50
6.2.2	WMS-Server auswählen . . . . .	51
6.2.3	Eine WMS-Layer laden . . . . .	52
6.2.4	Das Abfragewerkzeug . . . . .	54
6.2.5	Eigenschaften . . . . .	54
6.2.6	Einschränkungen des WMS-Klienten . . . . .	56
6.3	WFS-Klient . . . . .	56
6.3.1	Laden eines WFS-Layers . . . . .	56
<b>7</b>	<b>Arbeiten mit Projektionen</b>	<b>59</b>
7.1	Überblick zur Projektionsunterstützung . . . . .	59
7.2	Getting Started . . . . .	59
7.2.1	Eine Projektion spezifizieren . . . . .	61
7.3	Eigene Projektion definieren . . . . .	61
<b>8</b>	<b>GRASS Integration</b>	<b>63</b>
8.1	QGIS mit GRASS-Unterstützung starten . . . . .	63
8.2	GRASS-Daten laden . . . . .	64
8.3	Erstellen einer Location . . . . .	64
8.4	Vektordatenmodell . . . . .	66
8.5	Digitalisier- und Bearbeitungswerkzeuge . . . . .	67
8.5.1	Werkzeugleiste . . . . .	67
8.5.2	Kategorie-Reiter . . . . .	67
8.5.3	Einstellungsreiter . . . . .	68
8.5.4	Symbolgiereiter . . . . .	69
8.5.5	Tabellenreiter . . . . .	70
8.6	Regionswerkzeug . . . . .	70
8.7	GRASS Werkzeugkasten . . . . .	70
8.7.1	Die Module im Werkzeugkasten . . . . .	71
8.7.2	GRASS Datei-Browser . . . . .	72
8.7.3	Anpassen der Module . . . . .	72
8.8	Neue GRASS-Vektoren erzeugen . . . . .	74
<b>9</b>	<b>Erstellen einer Mapserver Map-Datei</b>	<b>75</b>
9.1	Erstellen einer QGIS Projektdatei . . . . .	75
9.2	Export einer QGIS Projektdatei als Map-Datei . . . . .	75
9.3	Testen der Map-Datei . . . . .	78
<b>10</b>	<b>Print Composer</b>	<b>79</b>
10.1	Den Print Composer benutzen . . . . .	79
10.1.1	Eine Karte zum Print Composer hinzufügen . . . . .	79
10.1.2	Weitere Elemente zum Print Composer hinzufügen . . . . .	81
10.1.3	Weitere Funktionen . . . . .	81

---

10.1.4 Eine Ausgabe erzeugen . . . . .	81
<b>11 Plugins benutzen</b>	<b>84</b>
11.1 Eine Einführung in das Nutzen der Plugins . . . . .	84
11.1.1 Ein Plugin finden und installieren . . . . .	84
11.1.2 Plugins verwalten . . . . .	84
11.1.3 Datenprovider . . . . .	84
11.1.4 Kernplugins . . . . .	86
11.1.5 Externe Plugins . . . . .	86
11.1.6 Plugin-Vorlagen . . . . .	87
11.2 Benutzung der Dekorations-Plugins . . . . .	88
11.2.1 Das Copyright Plugin . . . . .	88
11.2.2 Das Nordpfeil Plugin . . . . .	89
11.2.3 Das Maßstabsbalken Plugin . . . . .	89
11.3 Das GPS-Plugin benutzen . . . . .	91
11.3.1 Was ist GPS? . . . . .	91
11.3.2 GPS-Daten aus einer Datei laden . . . . .	91
11.3.3 GPSBabel . . . . .	92
11.3.4 GPS-Daten importieren . . . . .	92
11.3.5 GPS-Daten von einem Empfänger herunterladen . . . . .	92
11.3.6 GPS-Daten auf einen Empfänger hochladen . . . . .	93
11.3.7 Neue GPS-Gerätetypen definieren . . . . .	94
11.4 Das CSV-Plugin benutzen . . . . .	96
11.4.1 Anforderungen . . . . .	96
11.4.2 Das Plugin benutzen . . . . .	97
11.5 Die Benutzung des Gitternetz Plugins . . . . .	99
11.6 Die Benutzung des Georeferenzierungs-Plugin . . . . .	100
11.7 Die Benutzung des Python Plugins . . . . .	105
11.7.1 Die Erstellung einer Plugin-Struktur . . . . .	105
11.7.2 Das Plugin für QGIS sichtbar machen . . . . .	106
11.7.3 Quellen . . . . .	106
11.7.4 Erstellen der grafischen Benutzeroberfläche (GUI) . . . . .	107
11.7.5 Die Erstellung des Plugins . . . . .	107
11.7.6 Fragen und Probleme . . . . .	112
11.7.7 Hinzufügen einer Rückmeldung . . . . .	112
11.7.8 Zusammenfassung . . . . .	113
<b>12 Weitere Anwendungen mit den QGIS-Bibliotheken erstellen</b>	<b>114</b>
12.1 Erstellung der graphischen Oberfläche . . . . .	114
12.2 Erzeugen des Hauptfensters (MainWindow) . . . . .	115
12.3 Das Finale! . . . . .	120
12.4 Anwendung ausführen . . . . .	121

---

<b>13 Hilfe und Support</b>	<b>123</b>
13.1 Mailinglisten . . . . .	123
13.2 IRC . . . . .	124
13.3 BugTracker . . . . .	124
13.4 Blog . . . . .	125
13.5 Wiki . . . . .	125
<b>A Unterstützte Datenformate</b>	<b>126</b>
A.1 Unterstützte OGR-Formate . . . . .	126
A.2 GDAL Rasterformate . . . . .	126
<b>B Installationsanleitung</b>	<b>129</b>
B.1 Grundsätzliche Bauhinweise . . . . .	129
B.2 Übersicht über die Abhängigkeiten zum Bauen . . . . .	129
<b>C Bauen unter Windows mit msys</b>	<b>130</b>
C.1 MSYS: . . . . .	130
C.2 Qt4.3 . . . . .	130
C.3 Flex and Bison . . . . .	131
C.4 Python: (optional) . . . . .	131
C.4.1 Herunterladen und Installieren von Python - mit dem Windows- Installationsprogramm . . . . .	131
C.4.2 SIP- und PyQt4-Quellen herunterladen . . . . .	131
C.4.3 SIP kompilieren . . . . .	132
C.4.4 PyQt kompilieren . . . . .	132
C.4.5 Finale Python-Anmerkungen . . . . .	132
C.5 Subversion: . . . . .	132
C.6 CMake: . . . . .	132
C.7 QGIS: . . . . .	132
C.8 Kompilierung: . . . . .	133
C.9 Konfiguration . . . . .	133
C.10 Kompilieren und installieren . . . . .	134
C.11 qgis.exe aus dem Installationsverzeichnis starten (CMAKE_INSTALL_PREFIX) . . . .	134
C.12 Das Installationspaket erzeugen: (optional) . . . . .	134
<b>D Bauen auf Mac OSX mit frameworks und cmake (QGIS &gt; 0.8)</b>	<b>135</b>
D.1 Installieren von XCODE . . . . .	135
D.2 Installieren des Qt4 .dmg . . . . .	135
D.3 Installation der Entwicklungsumgebung für die QGIS-Anhängigkeiten . . . . .	136
D.3.1 Zusätzliche Abhängigkeit: GSL . . . . .	136
D.3.2 Zusätzliche Abhängigkeit: Expat . . . . .	136
D.3.3 Zusätzliche Abhängigkeit: SIP . . . . .	137
D.3.4 Zusätzliche Abhängigkeit: PyQt . . . . .	137

---

D.3.5	Zusätzliche Abhängigkeite: Bison . . . . .	138
D.4	CMake für Mac OSX . . . . .	138
D.5	Subversion für Mac OSX . . . . .	139
D.6	QGIS aus dem SVN auschecken . . . . .	139
D.7	Konfiguration der Quellen . . . . .	140
D.8	GEOS Hinweise . . . . .	141
D.9	Bauen . . . . .	141
<b>E</b>	<b>Bauen unter GNU/Linux</b>	<b>141</b>
E.1	QGIS mit Qt4.x bauen . . . . .	141
E.2	Apt vorbereiten . . . . .	142
E.3	Installieren von Qt4 . . . . .	142
E.4	Installation weitere Abhängigkeiten . . . . .	143
E.5	GRASS-spezifische Schritte . . . . .	143
E.6	Einrichtung von ccache (optional) . . . . .	143
E.7	Einrichten der Entwicklungsumgebung . . . . .	144
E.8	QGIS-Quellen auschecken . . . . .	144
E.9	Start des Kompilervorgangs . . . . .	145
E.10	QGIS ausführen . . . . .	146
<b>F</b>	<b>Erzeugen eine MSYS-Umgebung zum Kompilieren von QGIS</b>	<b>146</b>
F.1	Initiale Einrichtung . . . . .	146
F.1.1	MSYS . . . . .	146
F.1.2	MinGW . . . . .	146
F.1.3	Flex und Bison . . . . .	146
F.2	Abhängigkeiten installieren . . . . .	147
F.2.1	Ausgangssituation fertigstellen . . . . .	147
F.2.2	GDAL, erster Schritt . . . . .	147
F.2.3	GRASS . . . . .	149
F.2.4	GDAL, zweiter Schritt . . . . .	150
F.2.5	GEOS . . . . .	151
F.2.6	SQLITE . . . . .	151
F.2.7	GSL . . . . .	152
F.2.8	EXPAT . . . . .	152
F.2.9	POSTGRES . . . . .	152
F.3	Aufräumen . . . . .	152
<b>G</b>	<b>Bauen mit MS Visual Studio</b>	<b>153</b>
G.1	Visual Studio einrichten . . . . .	153
G.1.1	Express Edition . . . . .	153
G.1.2	Alle Editionen . . . . .	153
G.2	Herunterladen/Installieren der Abhängigkeiten . . . . .	154
G.2.1	Flex und Bison . . . . .	154



---

G.2.2 PostgreSQL Unterstützung in Qt . . . . .	154
G.2.3 Qt . . . . .	155
G.2.4 Proj.4 . . . . .	155
G.2.5 GSL . . . . .	156
G.2.6 GEOS . . . . .	156
G.2.7 GDAL . . . . .	157
G.2.8 PostGIS . . . . .	157
G.2.9 Expat . . . . .	157
G.2.10 CMake . . . . .	157
G.3 QGIS mit CMAKE bauen . . . . .	158
<b>H Mit MSVC Express unter Windows bauen</b>	<b>158</b>
H.1 System preparation . . . . .	159
H.2 Install the libraries archive . . . . .	159
H.3 Install Visual Studio Express 2005 . . . . .	159
H.4 Install Microsoft Platform SDK2 . . . . .	160
H.5 Edit your vsvars . . . . .	163
H.6 Environment Variables . . . . .	164
H.7 Install CMake . . . . .	167
H.8 Install Subversion . . . . .	167
H.9 Initial SVN Check out . . . . .	168
H.10 Create Makefiles using cmake-setup.exe . . . . .	169
H.11 Running and packaging . . . . .	169
<b>I GNU Public License</b>	<b>170</b>
I.1 Quantum GIS Qt Ausnahme der GPL . . . . .	175
<b>Literaturverzeichnis</b>	<b>176</b>

## Abbildungsverzeichnis

1	Hauptfenster mit Alaska Beispieldaten (GNU/Linux mit KDE)	10
2	Messwerkzeuge in Aktion	17
3	Dialog zum öffnen eines OGR-Vektordatenlayer	22
4	QGIS mit geladenem Shapefile von Alaska	23
5	Dialog für Ebeneneigenschaften	30
6	Auswählen eines Objektes mit einer Aktion	35
7	Dialog zum Digitalisieren von Vektor-Attributen	38
8	Dialog zum Erstellen eines neuen Vektorlayers	42
9	Query Builder	43
10	Popupmenü für Rasterlayer	46
11	Dialog mit den Rastereigenschaften	47
12	Dialog zum Hinzufügen eines WMS-Server mit den verfügbaren Layer	53
13	Einen WFS-Layer hinzufügen	57
14	Dialog Projekteinstellungen	60
15	Dialog Definition einer eigenen Projektion	61
16	Erstellen einer GRASS Location in QGIS	65
17	GRASS Editordialog	68
18	GRASS Werkzeugkasten	73
19	Das neu integrierte Modul nach der Auswertung der XML-Datei	74
20	Das 'Exportiere als MapServer Map' Module in QGIS	77
21	Print Composer	80
22	Print Composer mit Kartenfenster, Legende, Maßstab und Beschriftung	82
23	Plugin Manager	85
24	Das Copyright Plugin	88
25	Das Nordpfeil Plugin	89
26	Das Maßstabsbalken Plugin	90
27	Der Dialog <i>GPS Werkzeuge</i>	91
28	Dialog zum Importieren anderer GPS-Formate	93
29	Das Programm 'Von GPS herunterladen'	94
30	Getrennter Text Dialog	97
31	Ausgewählte Textdatei	98
32	In der Textdatei gefundene Spalten	98
33	Die X- und Y-Spalten auswählen	98
34	Erstellung eines Gitternetz-Layers	99
35	Ein Bild für die Georeferenzierung auswählen	100
36	Optimales Anordnen des QGIS Kartenfensters mit dem Georeferenzierer	101
37	Koordinaten für die Georeferenzierung setzen	102
38	Georeferenzierte Karte mit überlagerter Vektorkarte der spearfish location	104
39	Geben Sie einen neuen PostGIS Tabellennamen ein	110
40	Geben sie die Feldnamen für die neue PostGIS Tabelle ein	111

41	Geben Sie den DSN für die Verbindung mit der PostGIS Datenbank ein . . . . .	111
42	Nachrichtenfenster mit den Plugin-Ergebnissen . . . . .	113
43	Starten der neuen Demoanwendung . . . . .	121
44	Hinzufügen eines Themas zur Demo-Anwendung . . . . .	122

## Tabellenverzeichnis

1	PostGIS Verbindungsparameter . . . . .	25
2	WMS Verbindungs-Parameter . . . . .	51
3	Beispielhafte öffentliche WMS-URLs . . . . .	52
4	GRASS Editierwerkzeuge . . . . .	69
5	QGIS Kernplugins . . . . .	86

---

# 1 Vorwort

Willkommen in der wunderbaren Welt der Geographischen Informationssysteme (GIS)! Quantum GIS ist ein Freies (Open Source) GIS. Die Idee zu dem Projekt wurde im Mai 2002 geboren und bereits im Juni desselben Jahres bei SourceForge etabliert. Wir haben hart daran gearbeitet, traditionell sehr teure GIS Software kostenfrei für jeden, der Zugang zu einem PC hat, bereitzustellen. QGIS kann unter den meisten Unices, Windows und MacOSX betrieben werden. QGIS wurde mit Hilfe des Qt toolkit (<http://www.trolltech.com>) und C++ entwickelt. Dadurch ist QGIS sehr benutzerfreundlich und besitzt eine einfach zu bedienende und intuitive grafische Benutzeroberfläche.

QGIS soll ein einfach zu benutzendes GIS sein und grundlegende GIS-Funktionalitäten bieten. Das anfängliche Ziel bestand darin, einen einfachen Geo-Datenviewer zu entwickeln. Dieses Ziel wurde bereits mehr als erreicht, so dass QGIS mittlerweile von vielen Anwendern für ihre tägliche Arbeit eingesetzt wird. QGIS unterstützt eine Vielzahl von Raster- und Vektorformaten. Mit Hilfe der Plugin-Architektur können weitere Funktionalitäten einfach ergänzt werden (vgl. Appendix A für eine vollständige Liste derzeit unterstützter Datenformate).

QGIS wird unter der GNU Public License (GPL) herausgegeben. Für die Entwicklung des Programms bedeutet dies das Recht, den Quellcode einzusehen und entsprechend der Lizenz verändern zu dürfen. Für die Anwendung der Software ist damit garantiert, dass QGIS kostenfrei aus dem Internet heruntergeladen, genutzt und weitergegeben werden kann. Eine vollständige Kopie der Lizenz ist dem Programm beigelegt und kann auch im Appendix I eingesehen werden.

**Bemerkung:** Die aktuellste englische Version dieser Dokumentation finden Sie hier:  
<http://qgis.org/docs/userguide.pdf>

## 1.1 Funktionalitäten

Quantum GIS bietet zahlreiche GIS Funktionalitäten. Die wichtigsten sind hier aufgelistet, unterteilt in Kernfunktionalitäten und Plugins.

### Kernfunktionalitäten

- Raster- und Vektorsupport über die OGR-Bibliothek
- Support für PostgreSQL/PostGIS Datenbanken
- GRASS Integration zum Visualisieren, Editieren und Analysieren von GRASS Daten
- Digitalisieren von GRASS und OGR/Shapefiles
- Map Composer zum Erstellen von druckfertigen Kartenlayouts
- OGC-Support

- Übersichtsfenster
- Räumliche Bookmarks
- Identifizieren und Selektieren von Objekten
- Editieren, Visualisieren und Suchen nach Attributdaten
- Labeln von Objekten
- 'On the fly' Projektion
- Speichern und Öffnen von Projekten
- Export als Mapserver Mapfile
- Ändern der Vektor- und Rasterlayereigenschaften
- Erweiterbare Plugin-Architektur

### **Plugins**

- WFS Layer hinzufügen
- ASCII Tabellen als Layer hinzufügen
- Copyright Label, Nordpfeil und Maßstabsleiste hinzufügen
- Georeferenzierung
- GPS Tools
- GRASS-Anbindung
- Erstellen eines Gradnetzes
- PostgreSQL Geoprocessing Funktionalitäten
- SPIT (Shapefile nach PostgreSQL/PostGIS importieren)
- Python-Konsole
- openModeller

## **1.2 Was ist neu in 0.9**

Wie immer gibt es eine Reihe neuer, interessanter Funktionen in Version 0.9.

- Python bindings - Dies war das Hauptaugenmerk für diese Veröffentlichung. Es ist nun möglich, Plugins in Python zu erstellen. Ausserdem können nun GIS-unterstützende Applikationen in Python geschrieben werden, die auf die QGIS-Bibliotheken zugreifen.
- Das automake build system wurde entfernt - QGIS braucht nun CMake für das Kompilieren.
- Viele neue GRASS Module in der GRASS Toolbox (dank an <http://faunalia.it/>)
- Updates für den Map Composer

- Crash fix für 2.5D Shapefiles
- Optimierung des Georeferencer Plugins.
- Mittlerweile werden 26 Sprachen unterstützt

Für QGIS Version 0.9.1 stand Stabilisierung und Funktionsoptimierung im Vordergrund.

- 66 Bugfixes und Verbesserungen von Funktionen
- Automatische Anordnung der Kartenfenster beim Georeferencer Plugin
- Einstellungsmöglichkeit der Sprache im Menü Optionen
- Informationen über den Fortschritt des Datentransfers für WMS und WFS Daten
- Integration weiterer GRASS Module in die GRASS Toolbox

---

## 2 Einführung in GIS

Ein Geographisches Informationssystem (GIS)(1)<sup>1</sup> besteht aus einer Auswahl verschiedener Software, die einem ermöglicht, geografische Daten zu erstellen, zu visualisieren, abzufragen und zu analysieren. Geografische Daten beziehen sich dabei auf Informationen über die geografische Lage von Objekten. Dies bedingt meist eine geografische Koordinate oder die Zuweisung eines Längen und Breitengrades. Neben der Bezeichnung Geografische Daten ist Räumliche Daten ein ebenfalls häufig benutzter Begriff, genau wie GIS-Daten, Kartendaten, Location-Daten, Koordinatendaten und räumliche Geometriendaten.

Applikationen, die mit räumlichen Daten arbeiten, leisten eine Vielzahl unterschiedlicher Funktionen. Kartenerstellung ist dabei die am einfachsten zu begreifende GIS-Applikation. Mit Hilfe von Programmen werden dabei aus räumlichen Daten Karten erstellt, die digital auf PCs oder analog auf Papier gedruckt werden. Die Daten können dabei statisch (einfaches Bild) oder dynamisch (durch den Benutzer veränderbar) bereitgestellt werden.

Viele Menschen nehmen irrtümlicherweise an, dass GIS-Applikationen nur zur Kartenerstellung benutzt werden. Räumliche Datenanalyse ist jedoch eine weitere wichtige Funktion von Geographischen Informationssystemen. Einige typische Analysen sind:

1. Messen von Distanzen zwischen geografischen Punkten
2. Berechnung von Flächengrößen (z.B.: Quadratmeter) innerhalb einer Region
3. Bestimmung der Überlappung unterschiedlicher räumlicher Objekte
4. Bestimmung der überlappten Flächen innerhalb einer Region
5. Die Anzahl von Ortschaften entlang einer definierten Distanz
6. und so weiter...

Dies mag einfach erscheinen, wird jedoch täglich für viele verschiedene Fragestellungen angewendet und benötigt. Die Ergebnisse derartiger Analysen können auf Karten visualisiert oder in Form von Tabellen als Entscheidungshilfe dienen.

Das aktuelle Phänomen von ortsbasierten Diensten 'location-based services' verspricht dabei die räumliche Integration unterschiedlichster Informationen. Oft werden diese aus einer Kombination von Karten und darauf basierenden Analysen erstellt. Stellen Sie sich vor, Sie besitzen ein mobiles Telefon, das ihre genaue geografische Position ermittelt. Wenn Sie die entsprechende Software nutzen, kann Ihnen das Telefon sagen, welche Restaurants sich in der Nähe befinden. Während dies eine neuartige GIS-Technologie ist, ist das Analysieren von Geodaten und das Bereitstellen daraus gewonnener Informationen als eine Basis dafür zu verstehen.

---

<sup>1</sup>Dieses Kapitel ist von Tyler Mitchell (<http://www.oreillynet.com/pub/wlg/7053>) und wurde unter der Creative Commons License veröffentlicht. Tyler ist der Autor des Buches *Web Mapping Illustrated*, veröffentlicht von O'Reilly, 2005.



## 2.1 Warum ist das alles so neu?

Nun ja, ist es nicht. Es gibt heute viele Geräte auf dem Markt, die mobile räumliche Dienste anbieten, darunter auch viele Freie (Open Source) GIS-Applikationen. Die Existenz solcher Geräte ist nicht neu. GPS-Geräte sind z.B. heute für uns noch recht neu, werden aber von Unternehmen bereits seit mehr als 10 Jahren eingesetzt. Datenanalysen oder die digitale Erfassung der Umwelt mit dem PC auf Basis von Luftbildern war auch bereits früher ein kommerzieller Markt, der sich aber hauptsächlich auf Industrien aus dem Umweltsektor konzentriert hat.

Neu an all dem ist, wie die neueste Hard- und Software angewendet wird und von wem. Traditionell waren Anwender von GIS-Applikationen hochspezialisierte Forscher oder Techniker, die in der Anwendung von CAD-ähnlichen Anwendungen geschult waren. Mittlerweile bietet die Leistungsfähigkeit moderner Heim-PCs zusammen mit Freier (Open Source) Software einer Armada von Hobbyanwendern, Internetentwicklern usw. die Möglichkeit, geografische Daten selber zu nutzen und zu erstellen. Die Lernkurve ist dabei immer flacher geworden, während die Anzahl der Anwendungsmöglichkeiten stetig gestiegen ist.

Wie werden Geodaten gespeichert? Kurz gesagt gibt es zwei Datenmodelle, die heutzutage benutzt werden, ergänzend zu traditionell in tabellarischer Form abgelegten Informationen, die weiterhin eine wichtige Rolle bei der räumlichen Analyse spielen.

### 2.1.1 Rasterdaten

Das eine Datenmodell wird als Rasterdaten, oder einfach als 'Raster' bezeichnet. Ein typisches Beispiel hierfür sind Satelliten- und Luftbilder, genauso wie Höhenmodelle. Abgesehen von ein paar Ausnahmen können fast alle Daten als Raster dargestellt und gespeichert werden.

Ein Raster ist ein gleichmäßiges Gitter, das aus Zellen besteht, oder im Falle eines Bildes aus Pixeln. Ein Raster hat eine definierte Anzahl von Zeilen und Spalten. Jede Zelle hat einen numerischen Wert und eine bestimmte geografische Größe (z.B.: 30x30 Meter).

Sich mehrfach überlagernde Raster werden benutzt, um Bilder mit mehr als einem Farbwert zu repräsentieren. Ein Beispiel hierfür sind die Rot-, Grün- und Blaukanäle eines Satellitenbildes. Jeder Kanal eines Satellitenbildes ist ein einzelnes, sich mit den anderen Kanälen räumlich überlappendes Raster, das jeweils eine bestimmte Wellenlänge des Lichts darstellt. Wie man sich gut vorstellen kann, benötigt ein großes Raster mehr Speicherplatz als ein kleines, und je kleiner die Zellen eines Rasters sind, desto mehr Details können zwar dargestellt werden, aber desto mehr Speicherplatz wird benötigt. Die Kunst besteht darin, die Balance zwischen der maximalen Detailschärfe für die Analyse und einem möglichst geringen Speichervolumen zu finden.

### 2.1.2 Vektordaten

Vektordaten sind das andere Datenmodell und werden ebenso in GIS-Applikationen genutzt. Falls Sie in der Schule bei den Themen Trigonometrie und Koordinatensysteme wach geblieben sind, sind Sie sicher vertraut mit den Qualitäten und Besonderheiten von Vektoren. Einfach gesagt beschreiben Vektoren einen Ort durch eine Anzahl von Koordinaten. Jede Koordinate bezieht sich auf einen bestimmten geografischen Punkt mit Hilfe von x- und y-Werten.

Besser vorstellen kann man sich dies anhand einer kartesischen Ebene. Diese Diagramme aus der Schule mit einer x- und einer y-Achse. Vielleicht haben Sie es benutzt, um die sinkenden Renten oder steigenden Hypotheken darzustellen. Das Konzept ist ähnlich zur räumlichen Datenanalyse und Kartenerstellung.

Es gibt verschiedene Wege, um geografische Koordinaten darzustellen, abhängig von den Absichten und Zielen. Das Thema Kartenprojektionen ist dabei jedoch sehr komplex und wird an dieser Stelle daher nicht weiter erklärt.

Es gibt drei Arten von Vektordaten, die progressiv aufeinander aufbauen und dadurch immer komplexere Formen annehmen.

1. Punkte - Eine einzelne Koordinate (x y) repräsentiert ein diskretes geografisches Objekt
2. Linien - Multiple Koordinaten (x1 y1, x2 y2, x3 y4, ... xn yn), die in einer bestimmten Anordnung miteinander verbunden sind. Etwa wie eine Linie von Punkt (x1 y1) nach Punkt (x2 y2) und so weiter. Die Strecken zwischen den einzelnen Punkten sind Liniensegmente. Sie haben eine Länge und eine Richtung entsprechend der Anordnung der Punkte. Technisch gesehen ist eine Linie ein Paar von Koordinaten, die miteinander verbunden sind, wobei eine Linienkette aus mehreren Linien zusammengesetzt ist.
3. Polygone - Wenn Linien aus mehr als zwei Punkten bestehen und der letzte Punkte der Linie identisch mit dem ersten ist, wird das als Polygon bezeichnet. Dreiecke, Kreise, Rechtecke usw. sind Polygone. Die Schlüsseleigenschaft eines Polygons besteht darin, dass eine definierte Fläche dargestellt wird.

---

## 3 Getting Started

Dieses Kapitel gibt einen Überblick, wie man QGIS startet und erste Beispieldaten von der QGIS Webseite lädt und erkundet.

### 3.1 Installation

Die Installation von QGIS aus dem Quellcode ist in Appendix C für MS Windows, in Appendix D für Mac OSX und Appendix E für GNU/Linux dokumentiert. Das Installationshandbuch wird als Teil des mit den QGIS-Quellen bereitgestellt und ist auch zugänglich unter <http://qgis.org>

Aktuelle Binärpakete und aktuelle Informationen für MS Windows, Mac OSX und verschiedene GNU/Linux Distributionen gibt es auch auf der QGIS Webseite unter <http://download.qgis.org>.

### 3.2 Beispieldaten

Wenn Sie keine GIS-Daten zur Hand haben, können Sie die Alaska Beispieldaten von der QGIS Webseite <http://qgis.org> herunterladen. Die Projektion der Daten ist Alaska Albers Equal Area mit der Maßeinheit Meter:

```
PROJCS['NAD_1927_Albers',  
    GEOGCS['GCS_North_American_1927',  
    DATUM ['D_North_American_1927',  
        SPHEROID['Clarke_1866', 6378206.4, 294.9786982]],  
        PRIMEM['Greenwich', 0.0],  
        UNIT['Degree', 0.0174532925199433]],  
    PROJECTION['Albers'],  
    PARAMETER['False_Easting', 0.0],  
    PARAMETER['False_Northing', 0.0],  
    PARAMETER['Central_Meridian', -154.0],  
    PARAMETER['Standard_Parallel_1', 55.0],  
    PARAMETER['Standard_Parallel_2', 65.0],  
    PARAMETER ['Latitude_Of_Origin', 50.0],  
    UNIT['Meter', 1.0]]
```

Um das GRASS Plugin zu nutzen, kann eine Beispieldatenbank (z.B. spearfish60) von der Webseite des GRASS GIS Projektes <http://grass.itc.it/download/data.php> heruntergeladen werden. Die Projektion der Daten ist UTM Zone 13, Northern Hemisphere:

```

PROJCS['UTM Zone 13, Northern Hemisphere',
  GEOGCS['clark66',
    DATUM['North_American_Datum_1927',
      SPHEROID['clark66',6378206.4,294.9786982]],
    PRIMEM['Greenwich',0],
    UNIT['degree',0.0174532925199433]],
  PROJECTION['Transverse_Mercator'],
  PARAMETER['latitude_of_origin',0],
  PARAMETER['central_meridian',-105],
  PARAMETER['scale_factor',0.9996],
  PARAMETER['false_easting',500000],
  PARAMETER['false_northing',0],
  UNIT['meter',1]]

```

Diese beiden Datensätze werden auch als Basis für zahlreiche Beispiele und Screenshots in diesem Handbuch benutzt.

### 3.3 QGIS starten

Wenn Sie QGIS bereits installiert haben, kann das Programm gestartet werden, indem Sie in der Kommandozeile eines Terminals den Befehl **qgis** eintippen oder unter MS Windows auf das QGIS Icon auf dem Desktop doppelklicken. Unter Mac OSX liegt das QGIS Icon im Programmordner.

#### 3.3.1 Optionen der Kommandozeile

QGIS unterstützt eine Reihe von Optionen, wenn es aus der Kommandozeile gestartet wird. Um eine Liste der vorhandenen Optionen anzuzeigen, geben Sie `qgis --help` im Terminal ein. Die Ausgabe ist:

```

user@pc:~ > qgis --help
Quantum GIS - 0.9.1 'Ganymede'
Quantum GIS (QGIS) is a viewer for spatial data sets, including
raster and vector data.
Usage: qgis [options] [FILES]
  options:
    [--snapshot filename]  emit snapshot of loaded datasets to given file
    [--lang language]      use language for interface text
    [--project projectfile] load the given QGIS project
    [--extent xmin,ymin,xmax,ymax] set initial map extent
    [--help]                this text

```

**FILES:**

Files specified on the command line can include rasters, vectors, and QGIS project files (.qgs):

1. Rasters - Supported formats include GeoTiff, DEM and others supported by GDAL
2. Vectors - Supported formats include ESRI Shapefiles and others supported by OGR and PostgreSQL layers using the PostGIS extension

---

**Tip 1 BEISPIEL ZUR BENUTZUNG DER KOMMANDOZEILE**

Beim Starten von QGIS können zu ladende Datensätze bereits mit angegeben werden. Wenn Sie sich z.B. in Ihrem Datenverzeichnis mit einer Raster- und einer Vektordatei befinden, können diese beim Starten von QGIS durch folgenden Befehl mitgeladen werden: `qgis ak_shade.tif alaska.shp majrivers.shp`

---

**Option --snapshot**

Diese Option ermöglicht es, einen Snapshot des aktuellen Kartenausschnitts in QGIS zu erstellen. Dies ist besonders dann nützlich, wenn Sie schnell eine Vielzahl von Snapshots generieren möchten.

Dabei wird ein PNG mit einer Auflösung von 800x600 Pixeln generiert. Ein Dateiname wird vom Anwender festgelegt.

**Option --lang**

Auf Basis der Systemsprache Ihres PCs wird auch die Sprache der QGIS-Oberfläche eingestellt. Wenn Sie diese ändern möchten, können Sie das mit der Option `--lang` erreichen.

**Option --project**

Es ist auch möglich, beim Starten von QGIS ein Projekt zu laden. Fügen Sie dazu die Option `--project` mit dem Namen ihres Projektes hinzu und QGIS lädt alle darin enthaltenen Daten direkt beim Start.

**Option --extent**

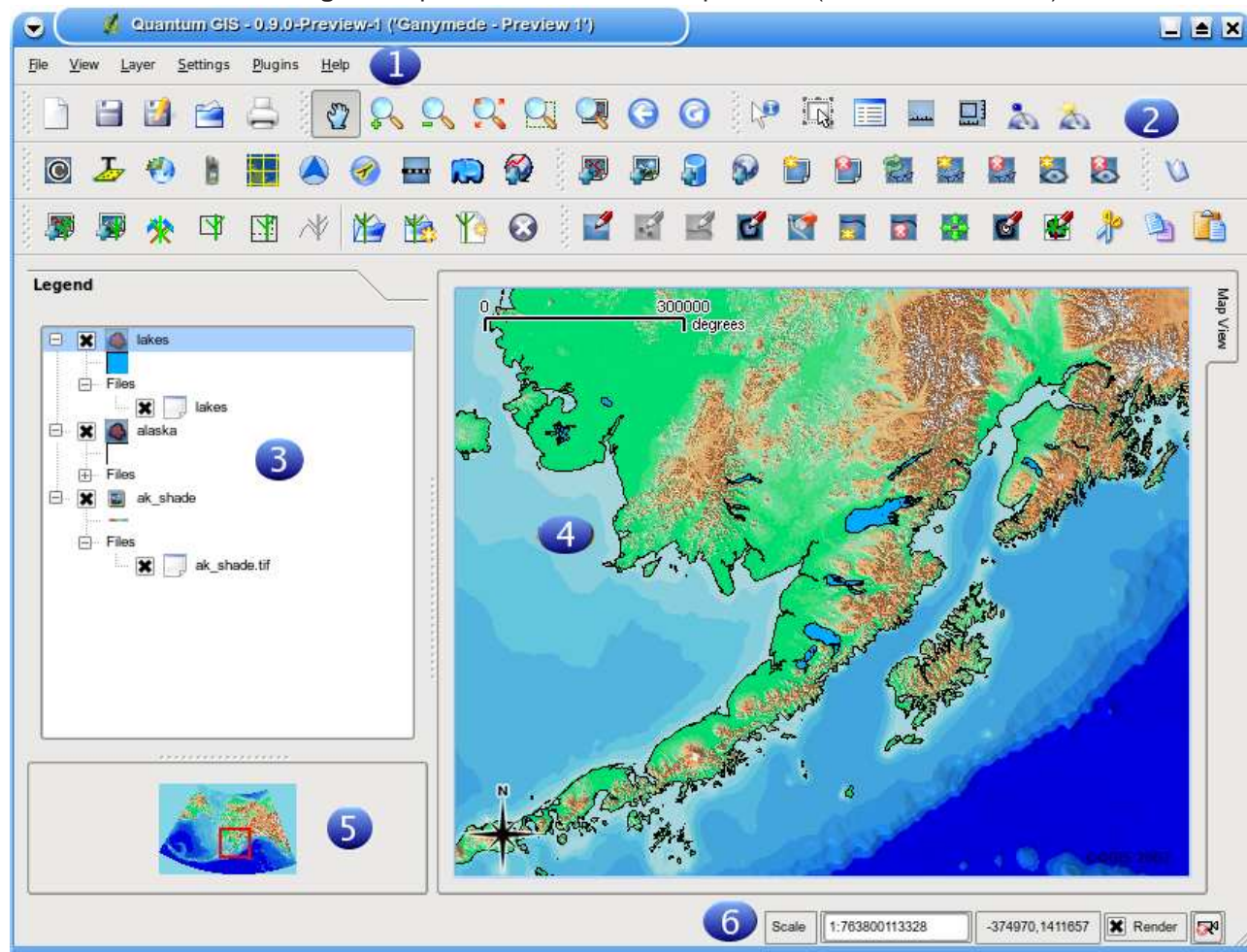
Um QGIS in einem bestimmten Ausschnitt Ihrer Daten zu starten, kann diese Option genutzt werden. Dazu wird durch die Eingabe von Eckkoordinaten eine 'Bounding Box' eingestellt. Die Koordinaten müssen durch Komma getrennt angegeben werden.

`--extent xmin,ymin,xmax,ymax`

### 3.4 QGIS GUI

Beim Starten von QGIS öffnet sich die grafische Benutzeroberfläche (GUI) wie in der folgenden Abbildung zu sehen. Die blauen Nummern von 1 bis 6 beziehen sich auf die sechs Hauptbereiche des Interfaces und werden unten genauer beschrieben.

**Abbildung 1:** Hauptfenster mit Alaska Beispieldaten (GNU/Linux mit KDE)



**Note:** Das Aussehen einzelner Bereiche (Menüleiste, etc.) kann abweichen in Abhängigkeit vom Betriebssystem und weiteren Einstellungen.

Die grafische Benutzeroberfläche von QGIS ist in sechs Bereiche unterteilt:

- |                   |                    |
|-------------------|--------------------|
| 1. Menüleiste     | 4. Kartenfenster   |
| 2. Werkzeugleiste | 5. Kartenübersicht |
| 3. Kartenlegende  | 6. Statusleiste    |

Diese sechs Komponenten des QGIS Interfaces werden im folgenden Abschnitt detailliert beschrieben.

### 3.4.1 Menüleiste

Die Menüleiste bietet Zugriff auf verschiedene Anwendungen mit Hilfe eines hierarchischen Menüs. Das Hauptmenü mit einer kurzen Beschreibung der enthaltenen Funktionen finden Sie hier:

- **Datei**
  - **Neues Projekt** - vgl. Kapitel 3.7
  - **Projekt öffnen** - vgl. Kapitel 3.7
  - **Öffne aktuelles Projekt** - vgl. Kapitel 3.7
  - **Speichere Projekt** - vgl. Kapitel 3.7
  - **Speichere Projekt als** - vgl. Kapitel 3.7
  - Speichere Bild als
  - **Exportiere als MapServer Datei** - vgl. Kapitel 9
  - **Drucken** - vgl. Kapitel 10
  - Beenden
- **Ansicht**
  - Volle Ausdehnung
  - Auf die Auswahl zoomen
  - Auf den Layer zoomen
  - Zur vorherigen Zoomeinstellung zurückkehren
  - Erneuern
  - Zeige Lesezeichen
  - Neues Lesezeichen
  - Die meisten Werkzeugleisten anzeigen
  - Die meisten Werkzeugleisten ausblenden
  - Werkzeugleistenanzeige...
- **Layer**
  - **Vektorlayer hinzufügen** - vgl. Kapitel 4
  - **Rasterlayer hinzufügen** - vgl. Kapitel 5
  - **PostGIS-Layer hinzufügen** - vgl. Kapitel 4.2
  - **WMS-Layer hinzufügen** - vgl. Kapitel 6.2
  - Layer löschen
  - **Neuer Vektorlayer** - vgl. Kapitel 4.4.3
  - In Übersicht anzeigen



- Alle zur Übersicht hinzufügen
  - Alle aus Übersicht entfernen
  - Alle Layer ausblenden
  - Alle Layer anzeigen
- **Einstellungen**
  - **Projekteinstellungen** - vgl. Kapitel 3.7
  - **Benutzerprojektionen** - vgl. Kapitel 7.3
  - **Optionen** - vgl. Kapitel 3.8
- **Plugins** - (Weitere Menüeinträge werden beim Laden ergänzt.)
  - **Plugin Manager** - vgl. Kapitel 11.1.2
- **Hilfe**
  - Hilfe Übersicht
  - QGIS Homepage
  - QGIS Version überprüfen
  - Über

### 3.4.2 Werkzeugleiste

Die Werkzeugleiste bietet mit Hilfe von Icons Zugriff auf alle Funktionen aus der Menüleiste sowie ein paar zusätzliche Tools, um mit der Karte zu interagieren. Jedes Icon der Werkzeugleiste bietet eine kurze Beschreibung ihrer Funktion. Indem Sie mit der Maus über das Icon fahren, erscheint es.

Jedes Untermenü der Werkzeugleiste kann nach eigenen Wünschen verschoben werden und jedes Untermenü kann auch an bzw. ausgeschaltet werden, indem Sie mit der Maus in den Bereich der Werkzeugleiste fahren und auf den rechten Mausknopf drücken.

---

**Tip 2 WERKZEUGLEISTE ZURÜCKHOLEN**

---

Wenn sie aus Versehen alle Untermenüs der Werkzeugleiste ausgeblendet haben, können Sie diese über den Knopf Die meisten Werkzeugleisten anzeigen aus dem Menü Ansicht weider zurückholen

---

### 3.4.3 Kartenlegende

Die Kartenlegende wird dazu benutzt, die Sichtbarkeit und die Z-Anordnung der Kartenlayer festzulegen. Z-Anordnung bedeutet, dass ein weiter oben in der Legende angeordneter Layer über einem weiter unten angeordneten Layer im Kartenfenster angezeigt wird. Mit der 'Checkbox' kann ausgewählt werden, ob ein Layer angezeigt werden soll, oder nicht.



Einzelne Layer können in der Legende zu Gruppen zusammengefasst werden. Gehen Sie dazu in den Bereich der Kartenlegende, klicken auf die rechte Maustaste und wählen Sie 'Gruppe hinzufügen'. Ein neuer Ordner erscheint und sie können Layer mit der Maus in diesen Ordner per 'drag and drop' verschieben. Um Layer wieder zurück in die Haupebene zu holen, gehen Sie mit der Maus in den Order ihrer Gruppe, wählen Sie einen Layer aus, klicken auf den rechten Mausknopf und wählen 'Als Top-Level Objekt machen' aus. Um dem Gruppenordner einen anderen Namen zu geben, wählen Sie mit der Maus einen Ordner aus, klicken auf die rechte Maustaste und wählen 'Umbenennen'.

Die angezeigten Elemente der rechten Maustaste-Funktion sind abhängig davon, ob es sich um einen Raster- oder Vektorlayer handelt. Für geladene GRASS Vektorlayer steht die Funktion 'Bearbeitungsstatus umschalten' nicht zur Verfügung. Vergleichen Sie dazu das Kapitel 8.5 für weitere Informationen zum Editieren von GRASS Vektorlayern.

- **Rechte-Maustaste Menü für Rasterlayer**

- Auf die Layerausdehnung zoomen
- Auf besten Maßstab zoomen (100%)
- In der Übersicht anzeigen
- Entfernen
- Eigenschaften
- Umbenennen
- Gruppe hinzufügen
- Alles ausklappen
- Alles zusammenfallen
- Zeige Dateigruppen

- **Rechte-Maustaste Menü für Vektorlayer**

- Auf die Layerausdehnung zoomen
- In der Übersicht anzeigen
- Entfernen
- Attributtabelle öffnen
- Bearbeitungsstatus umschalten (nicht für GRASS layer)
- Als Shapefile abspeichern
- Auswahl als Shapefile abspeichern
- Eigenschaften
- Umbenennen
- Gruppe hinzufügen
- Alles ausklappen
- Alles zusammenfallen
- Zeige Dateigruppen

- **Rechte Maustasten-Menü für Layergruppen**

- Entfernen
- Umbenennen
- Gruppe hinzufügen
- Alles ausklappen
- Alles zusammenfallen
- Zeige Dateigruppen

Wenn Sie Vektorlayer mit identischen Attributen anzeigen, kann die Darstellung der Symbole gruppiert werden. Dies bewirkt, dass bei Veränderung der Symbologie eines Layers entsprechend auch die Symbologie der anderen Layer in der Gruppe angepasst wird. Um die Darstellung von Symbolen gleicher Layer zu gruppieren, gehen Sie mit der Maus in die Kartenlegende, klicken auf die rechte Maustaste und wählen 'Zeige Dateigruppen'. Diese können von einer Gruppe nun mit der Maus in eine andere verschoben werden. Dadurch wird die Darstellung der Symbole gruppiert. Bitte beachten Sie, dass dies nur möglich ist, wenn zwei Vektorlayer vom gleichen Typ (Punkt, Linie oder Polygon) sind und gleiche Attribute beinhalten.

#### 3.4.4 Kartenfenster

Dies ist der Ergebnisbereich von QGIS - hier werden die Daten visualisiert. Die Karten, die hier angezeigt werden, hängen davon ab, welche Raster- und Vektorlayer Sie ausgewählt haben (vgl. folgende Abschnitte). Innerhalb des Kartenfensters können Sie 'zoomen', verschieben ('pan') und eine Vielzahl weiterer Funktionen aus der Werkzeugleiste anwenden. Das Kartenfenster und die Kartenlegende sind miteinander verknüpft. Layer im Kartenfenster spiegeln Veränderungen in der Kartenlegende wider.

---

#### **Tip 3** ZOOMEN IM LAYER MIT DEM MAUSRAD

Sie können das Mausrad benutzen, um im Kartenfenster in Layer hinein- bzw. aus Layern herauszuzoomen. Plazieren Sie dazu den Mauszeiger in das Kartenfenster und drehen Sie das Mausrad nach vorne (herauszoomen) oder nach hinten hineinzoomen. Der Mauszeiger bildet dabei das Zentrum. Sie können das Verhalten des Mausrades in der Menüleiste **Einstellungen** -> **Optionen** unter **Kartenwerkzeuge** einstellen

---

#### 3.4.5 Kartenübersicht

Im Bereich Kartenübersicht wird die Gesamtausdehnung eines Gebietes anhand eines ausgewählten Layers als Übersicht dargestellt. Dadurch ist es sehr einfach, einen schnellen Überblick über die genaue Position im Kartenfenster zu bekommen. Beschriftungen werden nicht in der Kartenübersicht angezeigt. Sie können einen einzelnen Layer in der Kartenübersicht anzeigen lassen, indem Sie einen Layer mit der Maus auswählen und mit der rechten Maustaste auf **Zur Übersicht hinzufügen** klicken. Dazu können auch die entsprechenden Icons der Werkzeugleiste genutzt werden.

Sie können mit der Maus auch das rote Rechteck, welches für den aktuell angezeigten Ausschnitt steht, in der Kartenübersicht verschieben. Dadurch wird automatisch der Ausschnitt im Kartenfenster angepasst.

### 3.4.6 Statusleiste

Die Statusleiste zeigt die aktuelle Position des Mauszeigers als Koordinate (z.B. in Meter oder Dezimalgrad). Außerdem werden die aktuellen Ausmaße des Kartenfensters angezeigt, wenn sie zoomen oder verschieben. Für einige Funktionen, z.B.: das Messen von Längen oder Flächen, wird das Ergebnis in der Statusleiste angezeigt. Mit der 'Checkbox' auf der rechten Seite der Statusleiste kann eingestellt werden, ob Layer angezeigt werden sollen oder nicht (vgl. Abschnitt 3.5). Rechts außen befindet sich ein Kamera-Icon. Wenn Sie es anklicken, öffnet sich das Menü für die Projektionseigenschaften des aktuellen Projektes.

## 3.5 Daten anzeigen

Im Normalfall findet eine Wiedergabe der ausgewählten Layer immer dann statt, wenn eine 'Erneuerung' notwendig ist. Die Abläufe, die eine 'Erneuerung' der Wiedergabe verursachen sind:

- Einen Layer hinzufügen
- Karten verschieben, hinein- oder herauszoomen
- Größe des QGIS-Fensters verändern
- Layer ein- oder ausblenden

QGIS ermöglicht es, den Wiedergabeprozess auf verschiedene Arten zu kontrollieren

### 3.5.1 Maßstabsabhängige Datenanzeige

Die maßstabsabhängige Wiedergabe erlaubt es, einen mini- und maximalen Maßstab anzugeben, in dessen Rahmen ein Layer angezeigt wird. Um die maßstababhängige Wiedergabe einzustellen, öffnen Sie die Eigenschaften eines Layers, indem Sie diese in der Kartenlegende doppelt anklicken, im Reiter **Allgemein** die entsprechenden Einstellungen vornehmen und die Checkbox **Benutze maßstabsabhängiges Zeichnen** aktivieren.

Sie können es einstellen, in dem Sie erst mit der Maus im Kartenfenster den gewünschten Maßstab wählen und dann die Werte aus der Statusleiste übernehmen.

### 3.5.2 Datenanzeige kontrollieren

Die Kartenwiedergabe kann auf folgende Art kontrolliert werden:

#### Wiedergabe unterdrücken

Um die Wiedergabe einer Karte zu unterdrücken, klicken Sie auf die Checkbox **Zeichnen** auf der rechten Seite der Statusleiste. Wenn die Checkbox nicht aktiv ist, führt QGIS keine Erneuerung der Wiedergabe des Kartenfensters durch, wie in Abschnitt 3.5 beschrieben. Beispiele, bei denen das Unterdrücken der Wiedergabe angewendet wird, sind:

- Viele Layer hinzufügen und ihre Symbologie vor der Visualisierung einstellen
- Eine oder mehrere große Layer hinzufügen und erst die maßstabsabhängige Wiedergabe einstellen
- Eine oder mehrere große Layer hinzufügen und erst in ein bestimmtes Gebiet hineinzoomen
- Eine Kombination aus den oben genannten Punkten

Wenn Sie die Checkbox **Zeichnen** aktivieren, findet automatisch eine Erneuerung der Wiedergabe des Kartenfensters statt.

#### Option für Layer hinzufügen

Sie können für das Hinzuladen neuer Layer eine Einstellung vornehmen, die bewirkt, dass diese beim Laden ersteinmal nicht angezeigt werden. Um diese Option zu nutzen, deaktivieren Sie in der Menüleiste unter **Einstellungen->Optionen->Darstellung** die Checkbox **Standardmäßig werden alle neuen Layer in der Karte angezeigt**.

#### Anpassen des Kartenfensters bei der Wiedergabe

Normalerweise werden in QGIS Inhalte von Layern erst visualisiert, wenn sämtliche Daten geladen sind. Um zu bewirken, dass die Inhalte von Layern direkt beim Lesen angezeigt werden, verändern Sie in der Menüleiste unter **Einstellungen->Optionen->Darstellung** den Wert **Aktualisieren während des Zeichnens**. Der Wert 0 (Standard) verhindert die Aktualisierung während des Zeichnens. Bei zu niedrigem Wert leidet die Leistung, da ständig ein Update der Wiedergabe stattfindet. Zu Beginn ist daher ein Wert von 500 sinnvoll.

**Achtung:** Diese Funktion ist unter der Qt 4.2 Version abgeschaltet, da sie nicht wie erwartet arbeitet.

## 3.6 Messen

Das Messen funktioniert nur korrekt mit projizierten Koordinatensystemen (z.B. UTM). Wenn die geladene Karte als Geografisches Koordinatensystem abgebildet wird (z.B. Latitude/Longitude), sind

die Messergebnisse falsch. Sie müssen dann ein passendes Koordinatensystem verwenden.

### 3.6.1 Strecken messen



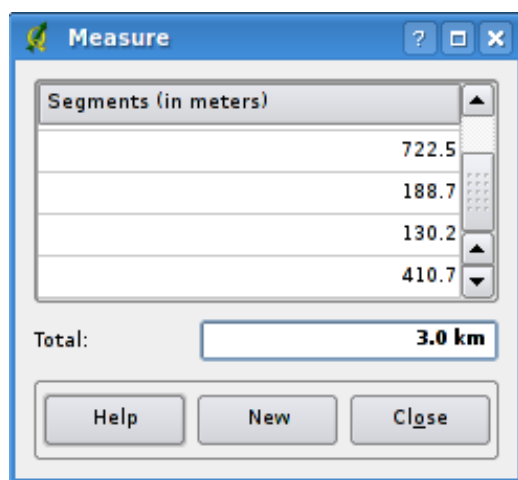
QGIS kann reale Distanzen zwischen Punkten auf Basis eines vorgegebenen Ellipsoiden messen. Dazu wählen Sie unter **Optionen** aus dem Menü **Einstellungen** den Reiter **Kartenwerkzeuge** und wählen einen Ellipsoiden aus. Das Tool ermöglicht es, Punkte im Kartenfenster auszuwählen. Jedes Längensegment wird in einem Popup angezeigt und zusätzlich auch die Gesamtlänge aller Längensegmente. Um das Messen von Strecken zu beenden, klicken Sie auf die rechte Maustaste.

### 3.6.2 Flächen messen

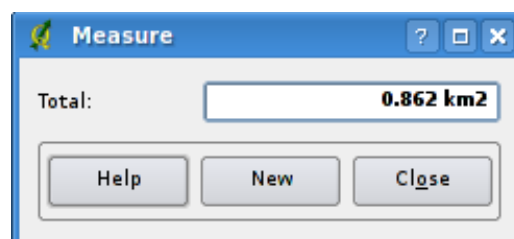


Auch Flächengrößen können gemessen werden. Das Popup zeigt dabei die akkumulierte Flächengröße (vgl. Bild 2).

**Abbildung 2:** Messwerkzeuge in Aktion



(a) Strecken messen



(b) Flächen messen

## 3.7 Projekte managen

Der Status einer gerade laufenden QGIS-Sitzung wird als Projekt bezeichnet. Sie können in QGIS immer nur an einem Projekt zur Zeit arbeiten. Einstellungen beziehen sich daher immer auf ein bestimmtes Projekt oder dienen als Standard-Einstellung für ein neues Projekt (vgl. Abschnitt 3.8).

QGIS kann den aktuellen Status eines Projektes in einer Projektdatei speichern. Wählen Sie dazu in

der Menüleiste Datei -> Speichere Projekt. Das Laden eines existierenden Projektes funktioniert entsprechend.

In einer Projektdatei sind folgenden Informationen gespeichert:

- Hinzugefügte Layer
- Einstellungen der Layer, inklusive Symbologie
- Projektion für das Kartenfenster
- Zuletzt gewählte Ausdehnung im Kartenfenster

Die Projektdatei wird im XML-Format gespeichert. Dadurch können Sie die Datei auch außerhalb von QGIS editieren, wenn Sie wissen, was Sie tun. Projektdateien aus älteren QGIS-Version funktionieren meist nicht.

## 3.8 GUI Optionen



Einige grundlegende Optionen für QGIS können unter Optionen eingestellt werden. Wählen Sie dazu unter **Einstellungen** den **Optionen**-Dialog (Alt-O). Einige der Optionen sind:

### Allgemein

- falls nötig zum Speichern des Projektes nachfragen

### Erscheinung

- Splashscreen beim Starten nicht anzeigen
- Ändere das Motiv der Icons
- Ändere Auswahl- und Hintergrundfarbe
- Setze Layernamen in Kapitälchen

### Darstellung

- Anzeige nach dem Laden aktualisieren.
- Wählen, ob neu geladene Layer angezeigt werden sollen oder nicht
- Linien auf Kosten der Zeichengeschwindigkeit weniger gezackt zeichnen
- Problem mit falsch gefüllten Polygonen beheben
- Karte kontinuierlich neuzeichnen, wenn der Teile zwischen Legende und Karte verschoben wird.

### Kartenwerkzeuge

- Suchradius für die Objektidentifizierung wählen
- Ellipsoid für Distanzberechnung definieren
- Farbe des Gummibandes der Distanzberechnung wählen
- Definiere Aktion des Mausekurses (Zoom, Zoom und Zentrieren, Nichts)
- Wähle einen Zoomfaktor für das Mausrad

### Projektion

- Definiere, was passieren soll, wenn ein Layer ohne Projektion geladen wird
  - Projektion erfragen
  - Projektweite Einstellung verwenden
  - Global definierte Projektion aus dem unteren Fenster benutzen

### Sprache

- Andere Spracheinstellung erzwingen
- Informationen zur aktuellen Einstellung anzeigen

### Hilfe Browser

- Definiere den Webbrowser, um die Hilfe anzuzeigen

Passen Sie die Optionen entsprechend Ihren Wünschen an. Einige Änderungen können einen Neustart von QGIS notwendig machen, bevor sie aktiv werden.

Unter GNU/Linux Distributionen ist alles gespeichert unter:

```
$HOME/.config/QuantumGIS/qgis.conf
```

Dies ist eine Textdatei, in der QGIS die aktuellen Optionen, PostGIS und WMS Verbindungen sowie andere Einstellungen speichert.

Unter MS Windows ist alles gespeichert unter:

```
\\HKEY_CURRENT_USER\\Software\\QuantumGIS\\qgis
```

Unter Mac OS ist alles gespeichert unter:

```
$HOME/Library/Preferences/org.qgis.qgis.plist
```

## 3.9 Räumliche Lesezeichen

Räumliche Lesezeichen ermöglichen es, geografische Orte zu markieren und als Lesezeichen abzuspeichern, um dorthin später zurückzufinden.

### 3.9.1 Ein Lesezeichen erstellen

Um ein Lesezeichen zu erstellen:

1. Zoomen Sie in einen Kartenausschnitt Ihrer Wahl.
2. Wählen Sie in der Menüleiste `Ansicht->Neues Lesezeichen` oder tippen Sie [Ctrl-B].
3. Geben Sie eine Beschreibung für das Lesezeichen ein (bis zu 255 Zeichen).
4. Klicken Sie auf `OK`, um das Lesezeichen zu speichern oder `Cancel` um kein Lesezeichen zu erstellen.

Sie können mehrere Lesezeichen mit demselben Namen erstellen.

### 3.9.2 Arbeiten mit Lesezeichen

Um ein bestehendes Lesezeichen zu editieren, wählen Sie in der Menüleiste `Ansicht->Zeige Lesezeichen`. Das Dialogfenster erlaubt es, auf ein Lesezeichen zu zoomen oder es zu löschen. Sie können die Koordinaten oder den Namen des Lesezeichens nicht verändern.

### 3.9.3 Auf ein Lesezeichen zoomen

Im Lesezeichen-Dialogfenster klicken Sie auf das gewünschte Lesezeichen und wählen dann den Knopf `Zoomen auf`. Sie können auch auf ein Lesezeichen zoomen, indem Sie es mit einem Doppelklick auswählen.

### 3.9.4 Ein Lesezeichen löschen

Um ein Lesezeichen aus dem Lesezeichen-Dialogfenster zu löschen, klicken Sie auf das gewünschte Lesezeichen, dann auf den Knopf `Löschen` und bestätigen Sie es mit `Ja` oder `Nein`.



---

## 4 Arbeiten mit Vektordaten

QGIS unterstützt eine Vielzahl unterschiedlicher Vektorformate, inklusive allen Formaten, die durch die OGR-Bibliothek unterstützt werden, wie etwa das ESRI Shape, MapInfo MIF (Austauschformat) und MapInfo TAB (Natives Format). QGIS unterstützt auch PostGIS-Ebenen in einer PostgreSQL Datenbank mit Hilfe des PostGIS-Plugins. Unterstützung für weitere Datenformate (z.B.: ASCII-Tabellen oder GPS-Daten) sind mittels weiterer Plugins realisiert.

In diesem Abschnitt wird beispielhaft beschrieben, wie man mit ESRI Shapes und PostGIS-Ebenen arbeitet. Viele Funktionen in QGIS sind unabhängig vom verwendeten Datenformat und verhalten sich daher identisch. Dies ist gewollt und bezieht sich u.a. auf Abfrage, Selektion, Beschriftung und Attributfunktionen.

Das Arbeiten mit GRASS Vektordaten bildet an manchen Stellen eine Ausnahme und wird daher in Kapitel 8 explizit beschrieben.

### 4.1 ESRI Shapes

ESRI Shape Unterstützung wird durch die OGR Simple Feature Library <http://www.gdal.org/ogr> bereitgestellt. Vergleichen Sie im Appendix A.1 die Liste mit den unterstützten Datenformaten.

Ein Shape besteht aus mindestens drei Dateien:

- .shp Datei (enthält die Geometrien)
- .dbf Datei (enthält die Attribute im dBase-Format)
- .shx Datei (enthält den räumlichen Index)

Idealerweise existiert noch eine weitere Datei mit .prj Endung. Diese enthält die Projektionsinformationen des Shapes.

Es können noch weitere Dateien zu einem Shape gehören. Weitere Details finden sich in der technischen Spezifikation unter <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

#### 4.1.1 Shape laden



Um ein Shape zu laden, starten Sie QGIS und klicken auf das Icon **Füge einen Vektorlayer hinzu** in der Werkzeugleiste. Mit demselben Tool können Sie auch alle anderen durch OGR unterstützten Formate laden.

Beim öffnen des Tools öffnet sich ein Standard-Dialog (Abbildung 3), mit dessen Hilfe Sie in Ihrem

Dateisystem navigieren und die entsprechenden zu ladenden Dateien auswählen können. Die Auswahlbox Files of type ermöglicht es, bereits ein Format vorauszuwählen.

Sie können auch die Kodierung der Datei auswählen, wenn benötigt.

**Abbildung 3:** Dialog zum öffnen eines OGR-Vektordatenlayer



Durch Auswahl eines Shapes und Anklicken von 'Open' wird die Datei in QGIS geladen. In Abbildung 4 sehen Sie das Ergebnis, nachdem die Beispieldatei alaska.shp geladen wurde.

---

#### **Tip 4** FARBEN VON VEKTORLAYERN

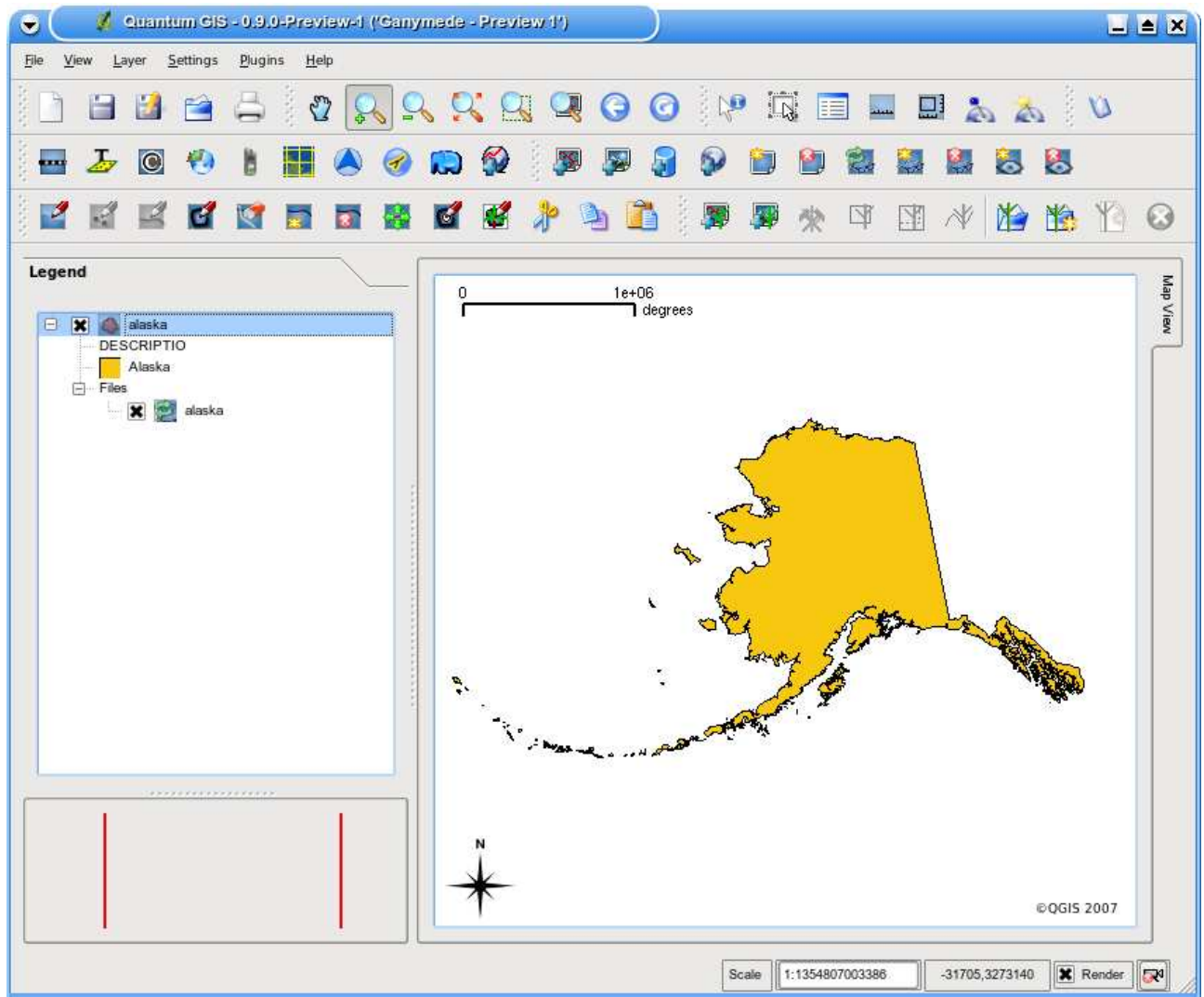
---

Wenn Sie eine neue Ebene in QGIS laden, werden Farben zufällig zugewiesen. Wenn Sie mehrere neue Ebenen laden, werden jeweils unterschiedliche Farben zugewiesen.

---

Nach dem Laden können Sie mit den Navigationstools aus der Werkzeugleiste beliebig zoomen oder die Karte verschieben. Um die Symbologie einer Ebene zu verändern, doppelklicken Sie auf den Ebenennamen in der Kartenlegende oder zeigen Sie mit der Maus auf den Ebenennamen und klicken Sie auf die rechte Maustaste und wählen *Eigenschaften*. Vergleichen Sie Abschnitt 4.3.1 für weitere Informationen zum Editieren der Eigenschaften von Vektorlayern.

Abbildung 4: QGIS mit geladenem Shapefile von Alaska



#### 4.1.2 Darstellungsgeschwindigkeit optimieren

Um die Darstellungsgeschwindigkeit zu optimieren, kann ein räumlicher Index erstellt werden. Ein räumlicher Index erhöht die Geschwindigkeit beim Zoomen und Verschieben. Räumliche Indizes haben in QGIS die Endung `.qix`.

Benutzen Sie folgende Schritte zum Erstellen eines räumlichen Index:

- Laden Sie ein Shape
- Öffnen Sie den **Eigenschaften**-Dialog der Ebene, indem Sie auf den Namen der Ebene doppelklicken oder mit der rechten Maustaste **Eigenschaften** auswählen.

- In dem Reiter Allgemein klicken Sie den Erstellen-Knopf im Bereich 'Räumlichen Index erstellen'

#### 4.1.3 MapInfo Datei laden

Um eine MapInfo Datei zu laden, klicken Sie auf das Icon Füge einen Vektorlayer hinzu in der Werkzeugleiste, ändern Sie die Auswahlbox Files of type nach MapInfo (\*.mif \*.tab \*.MIF \*.TAB) und wählen die gewünschte Datei.

#### 4.1.4 ArcInfo Coverage laden

Um ein ArcInfo Coverage zu laden, klicken Sie auf das Icon Füge einen Vektorlayer hinzu in der Werkzeugleiste, ändern Sie die Auswahlbox Files of type nach All files (\*.\*) und wählen die gewünschte Datei.

- .lab - Attributlayer laden (Flächenattribute oder Punktdaten)
- .cnt - Polygon-Centroid-Layer laden
- .arc - Linienlayer laden
- .pal - Polygonlayer laden

### 4.2 PostGIS-Ebene laden

PostGIS-Ebenen sind in einer PostgreSQL Datenbank gespeichert. Der Vorteil von PostGIS liegt in der Fähigkeitkeit, räumliche Indizes, Filterungen und Abfragen bereitzustellen. Vektorfunktionen wie Selektieren und Abfragen funktionieren besser als bei Ebenen, die durch die OGR-Bibliothek geladen wurden.

Um PostGIS benutzen zu können, brauchen Sie folgende Schritte:

- Verknüpfen Sie QGIS mit einer PostgreSQL Datenbank (falls dies noch nicht geschehen ist)
- Stellen Sie eine Verbindung zur Datenbank her
- Wählen Sie eine PostGIS-Ebene zum laden
- Optional nutzen Sie eine SQL-Abfrage, um zu definieren, welche Objekte der Ebene angezeigt werden sollen.
- Laden Sie die Ebene

### 4.2.1 Erstellen einer Verknüpfung



Beim ersten Einbinden einer PostGIS -Datenquelle muss eine Verknüpfung erstellt und gespeichert werden. Klicken Sie dazu auf den Knopf **PostGIS-Ebene hinzufügen** in der Werkzeugleiste. Der **PostGIS Tabellen hinzufügen**-Dialog erscheint daraufhin. Um eine neue Verbindung zu erstellen, klicken Sie auf den Knopf **Neu**, um den Dialog **Neue PostGIS-Verbindung** erzeugen zu öffnen. Die notwendigen Parameter sind in Tabelle 1 dargestellt.

**Tabelle 1:** PostGIS Verbindungsparameter

Name	Ein Name für die Verbindung. Kann derselbe wie für die Datenbank sein
Host	Name des Datenbank-Hosts. Dies muss ein auflösbarer Name für den HOST sein, genau wie beim Benutzen von telnet oder ping
Datenbank	Name der Datenbank
Port	Port Nummer der Datenbank auf dem Server. Standard ist 5432.
Benutzername	Benutzername, um sich bei der Datenbank anzumelden
Passwort	Passwort, um sich als Benutzer mit der Datenbank zu verbinden

Wenn alle Parameter eingetragen sind, kann die Verbindung getestet werden, indem Sie auf den Knopf **Teste Verbindung** drücken. Um das Passwort zusammen mit den Verbindungsdetails zu speichern, aktivieren Sie **Speichere Passwort**.

---

#### Tip 5 QGIS BENUTZEREINSTELLUNGEN UND SICHERHEIT

---

Ihre Einstellungen für QGIS werden abhängig von Ihrem Betriebssystem gespeichert. Unter Linux/Unix ist dies in Ihrem Homeverzeichnis unter `.qt/qgisrc`. Unter MS Windows ist dies in der Registry. Abhängig von Ihrer Arbeitsumgebung kann das Speichern von Passwörtern in den QGIS-Einstellungen ein Sicherheitsrisiko darstellen.

---

### 4.2.2 PostGIS-Ebene laden



Nachdem Sie eine oder mehrere Verbindungen definiert und gespeichert haben, können Sie einzelne Ebenen aus PostGIS laden. Natürlich brauchen Sie dazu Daten in PostgreSQL/PostGIS. Eine Beschreibung zum Import von Daten nach PostgreSQL/PostGIS lesen Sie Abschnitt 4.2.4.

Um eine Ebene aus PostGIS zu laden, führen Sie folgende Schritte durch:

- Wenn der **PostGIS Tabellen hinzufügen**-Dialog noch nicht geöffnet ist, klicken Sie auf den Knopf **PostGIS-Ebene hinzufügen** in der Werkzeugleiste
- Wählen Sie eine Verbindung aus dem Drop-Down Menü und klicken auf **Verbinden**
- Suchen Sie eine Ebene, die sie laden möchten

- Wählen Sie die Ebene durch doppelklicken aus. Sie können mehrere Ebenen auswählen, indem Sie die Shift-Taste beim Auswählen gedrückt halten. Weitere Informationen zum Umgang mit dem PostgreSQL Query-Builder finden Sie in Abschnitt 4.5.
- Klicken Sie auf den Hinzufügen-Knopf, um die Ebene zu laden

---

**Tip 6 POSTGIS-EBENEN**

Normalerweise ist eine PostGIS-Ebene über einen Eintrag in der `geometry_columns` Tabelle definiert. Seit Version 0.8.1 ist QGIS in der Lage, Ebenen zu laden, die keinen Eintrag in der `geometry_columns` Tabelle besitzen. Dies bezieht sich auf Tabellen und Views. Um einen 'spatial view' zu definieren, brauchen Sie ein kraftvolles System, um die Daten zu visualisieren. Beziehen Sie sich auf das PostgreSQL Handbuch, um weitere Informationen über die Erstellung von Views zu erhalten.

---

### 4.2.3 Einige Details zu PostGIS-Ebenen

Dieser Abschnitt enthält einige Details, wie QGIS auf PostgreSQL zugreift. Meistens soll QGIS eine Liste mit ein paar Datenbanktabellen bereitstellen, die bei Bedarf geladen werden können. Wenn Sie Probleme mit dem Laden von Ebenen aus PostgreSQL haben, können die nun folgenden Informationen vielleicht eine Hilfe sein, die Fehlermeldungen von QGIS besser zu verstehen und eine Lösung zu finden, die PostgreSQL Tabellen- oder Viewdefinition anzupassen und somit die Ebenen laden zu können.

QGIS braucht eine Spalte in der PostgreSQL Tabelle, die als eindeutiger Schlüssel für die Ebene genutzt werden kann. Für Tabellen bedeutet dies, dass die Tabelle normalerweise einen Primärschlüssel braucht oder eine Spalte mit eindeutiger Zuordnung. Für QGIS muss diese Spalte außerdem vom Typ `int4` sein (Integer mit 4 Bytes). Wenn eine Tabelle diese Voraussetzungen nicht erfüllt, wird die `oid`-Spalte genutzt. Die Performanz steigt, wenn für die Spalte ein Index erstellt wurde (für Primärschlüssel wird in PostgreSQL automatisch ein Index erstellt).

Wenn die PostGIS-Ebene ein View ist, gelten dieselben Anforderungen, aber Views besitzen keinen Primärschlüssel oder Spalten mit eindeutiger Zuordnung. In diesem Fall wird QGIS versuchen, eine passende Spalte zu finden, die aus der Tabellenspalte erstellt wurde. Wenn so eine Spalte nicht existiert, wird die Ebene nicht geladen. In diesem Fall ist eine Lösung, den View anzupassen, damit er eine entsprechende Spalte vom Typ `int4` enthält, entweder mit einem Primärschlüssel oder eindeutiger Zuordnung und bevorzugt natürlich mit Index.

### 4.2.4 PostGIS-Ebene importieren

#### **shp2pgsql**

Ebenen können auf unterschiedliche Art nach PostGIS importiert werden. PostGIS selbst stellt ein Tool `shp2pgsql` bereit, das genutzt werden kann, um Shapes nach PostGIS zu importieren. Um ein

Shape mit dem Namen 'lakes' in eine PostgreSQL/PostGIS Datenbank zu importieren, wird folgendes Kommando benutzt:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

Dieser Befehl erzeugt eine neue Tabelle mit dem Namen 'lakes\_new' in der PostgreSQL/PostGIS Datenbank 'gis\_data'. Die neue Ebene wird die ID 2964 als 'spatial reference identifier' (SRID) tragen. Weitere Informationen zu räumlichen Referenzsystemen finden Sie in Abschnitt 7.

---

#### **Tip 7** POSTGIS-EBENEN EXPORTIEREN

Entsprechend dem Importtool `shp2pgsql` gibt es auch ein Exporttool, um Daten von PostGIS ins Shapeformat zu exportieren: `pgsql2shp`. Dies ist in der PostGIS-Installation enthalten.

---

### **SPIT Plugin**



QGIS bietet ein Plugin mit dem Namen SPIT (Shapefile to PostGIS Import Tool). SPIT kann dazu benutzt werden, gleichzeitig mehrere Shapes zu laden und bietet dabei auch Support für Schemas. Um das Plugin zu benutzen, öffnen Sie den Pluginmanager, wählen SPIT aus und laden es. Das SPIT-Icon wird nun in der Plugin-Werkzeugleiste angezeigt.

Um ein Shape zu importieren, klicken Sie auf das SPIT-Icon in der Plugin-Werkzeugleiste, um den Dialog zu öffnen. Sie können ein oder mehrere Shapes in die Liste laden, indem Sie auf den Knopf **Hinzufügen** klicken. Der Ablauf der Imports sowie Warnungen oder Fehlermeldungen werden für jedes Shape angezeigt.

---

#### **Tip 8** SHAPES MIT FÜR POSTGRESQL RESERVIERTEN AUSDRÜCKEN IMPORTIEREN

Wenn Sie ein Shape in der Liste zu importierender Daten haben, welches Felder mit für PostgreSQL reservierten Ausdrücken enthält, erscheint ein Popup mit dem jeweiligen Status der Felder. Sie können die Namen der Felder dann vor dem Import editieren und anpassen. Der Versuch die Datei ohne Änderung zu importieren, wird fehlschlagen.

---

### **ogr2ogr**

Abgesehen von `shp2pgsql` und SPIT gibt es noch ein weiteres Tool, um Daten nach PostGIS zu importieren: `ogr2ogr`. Dieses Tool ist Teil der GDAL-Installation. Um ein Shape nach PostGIS zu importieren, kann folgendes Kommando verwendet werden:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres \
password=topsecret" alaska.shp
```

Damit wird das Shape `alaska.shp` vom Benutzer `postgres` mit dem Passwort `topsecret` in die PostGIS Datenbank `postgis` auf dem Server `myhost.de` importiert.



Wichtig ist, dass GDAL gegen PostgreSQL kompiliert sein muss, damit der Import nach PostGIS möglich ist. Dies kann mit folgendem Kommando getestet werden:

```
ogrinfo --formats | grep -i post
```

#### 4.2.5 Geschwindigkeit optimieren

Der Datentransfer von einer PostgreSQL/PostGIS Datenbank kann langsam sein, besonders über ein Netzwerk. Die Geschwindigkeit kann optimiert werden, indem für alle Ebenen in PostgreSQL ein räumlicher Index erstellt wird. PostGIS unterstützt das Erstellen eines GiST (Generalized Search Tree) Index, um den Zugriff auf die Ebenen zu beschleunigen.

Die Syntax zur Erstellung eines GiST<sup>2</sup> Index ist:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

Bedenken Sie, dass das Erstellen eines Index bei großen Datenmengen zeitaufwendig ist. Nachdem der Index erstellt ist, sollte ein VACUUM ANALYZE durchgeführt werden (vgl. PostGIS Dokumentation (4)).

Beispiel zur Erstellung eines GiST Index:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.0.0, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit

gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data=# USING GIST (the_geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

<sup>2</sup>GiST Index Informationen wurden aus der PostGIS Dokumentation übernommen unter: <http://postgis.refractory.net>



### 4.3 Dialog für Ebeneneigenschaften

Der Dialog für die Ebeneneigenschaften von Vektordaten gibt Informationen über den Vektorlayer, ihre Symbologie, Eigenschaften und Beschriftungsoptionen. Wenn ein Vektorlayer aus einer PostgreSQL/PostGIS Datenbank geladen wurde, können auf diese SQL-Abfragen angewendet werden - entweder manuell im Allgemein Reiter des Dialogs oder indem der Query Builder im Reiter Allgemein genutzt wird. Um den Dialog für die Ebeneneigenschaften zu öffnen, doppelklicken Sie auf den Namen der Ebene in der Kartenlegende oder wählen mit der rechten Maustaste 'Eigenschaften' aus.

#### 4.3.1 Bezeichnungen

QGIS bietet eine Vielzahl von Symbolfunktionen, um die Darstellung von Vektoren zu kontrollieren. Derzeit sind folgende Optionen vorhanden:

**Einfaches Symbol** - eine definierte Symbologie wird allen Objekten der Ebene zugewiesen.

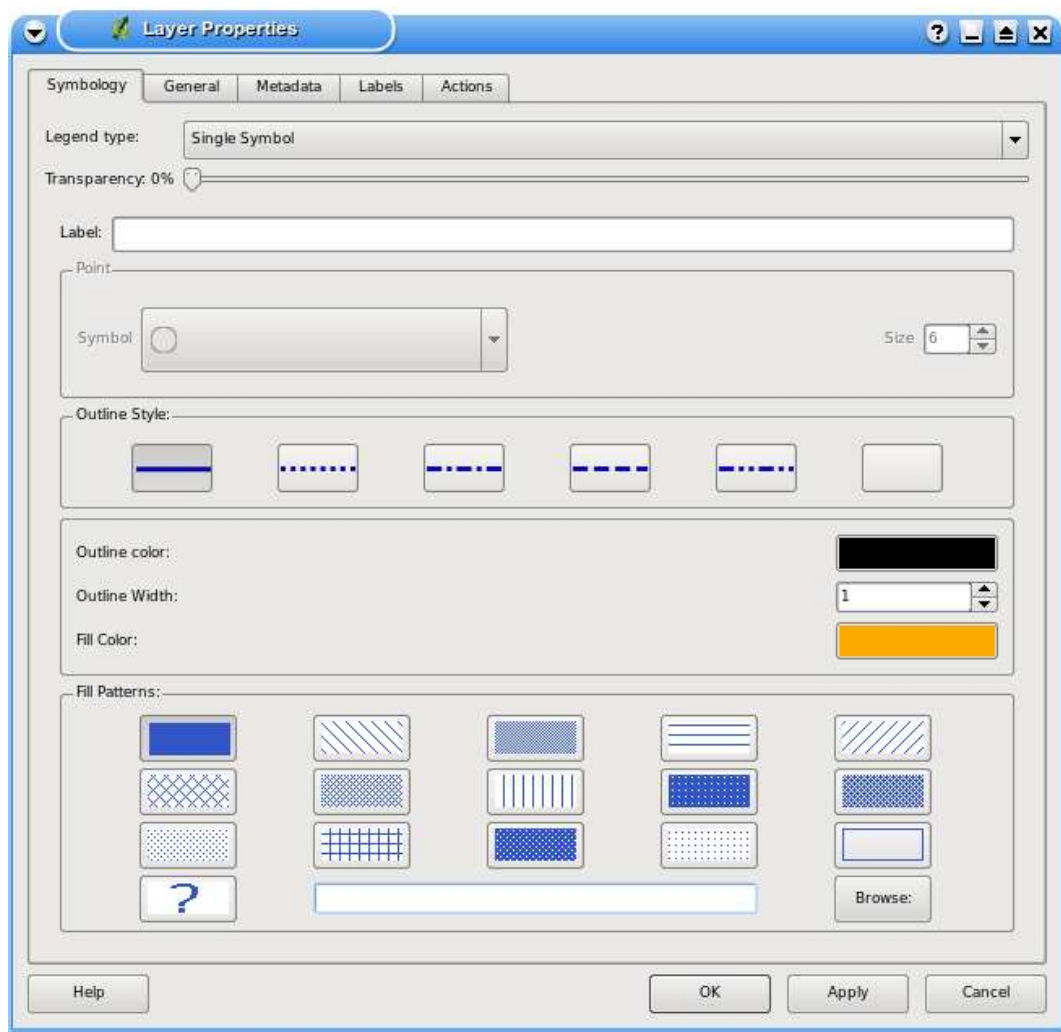
**Abgestuftes Symbol** - Objekten einer Ebene wird in Abhängigkeit von einer Attributspalte und einer definierten Anzahl von Klassen eine Symbologie zugewiesen.

**Fortlaufende Farbe** - Objekten einer Ebene wird anhand von definierten, numerischen Werten eine fortlaufende Farbe zugewiesen.

**Individueller Wert** - eine definierte Symbologie wird jeder Objektklasse einer Ebene anhand einer definierten Attributspalte zugewiesen.

Um die Symbologie einer Ebene zu ändern, doppelklicken Sie auf den Namen der Ebene in der Kartenlegende.

Abbildung 5: Dialog für Ebeneneigenschaften



**Neu in v0.9** ist eine Funktion, um Bilder zum als Füllfarbe für ein Vektorpolygon zu verwenden.

### Transparenz für Vektorlayer

QGIS ermöglicht es, einen Vektorlayer transparent darzustellen. Dies kann mit dem Schieber unterhalb des Legendentyps (vgl. Abbildung 5) erreicht werden und ist sehr nützlich, um mehrere Vektorlayer übereinander darzustellen.

### 4.3.2 Allgemein

Der Reiter Allgemein ist für Vektor- und Rasterlayer gleich. Es ermöglicht das Ändern des Anzeigens sowie die Einstellung einer maßstabsabhängigen Visualisierung. Zusätzlich ist es aber noch möglich, einen räumlichen Index zu erstellen, der die Visualisierung der Daten beschleunigt (nur für OGR-unterstützte Formate und PostGIS) sowie die Projektion der Visualisierung zu ändern.

Das Query Builder Icon ermöglicht es, eine Objektabfrage auf Basis der Attributdaten durchzuführen. Der Knopf ist im Moment aber nur auswählbar über das Icon Erweitert in der Attributtabelle.

### 4.3.3 Metadaten

Der Reiter Metadaten enthält Informationen zu dem Layer inklusive Angaben zu Datentyp, Datenquelle, Anzahl Objekte, Ausdehnung, Editiermöglichkeiten und Projektion. Dies ist ein schneller Weg, um allgemeine Informationen zu den Daten zu bekommen.

### 4.3.4 Beschriftungen

Der Reiter Beschriftungen ermöglicht es, Objekte zu beschriften und die Anordnung, den Stil sowie einen Puffer zu definieren.

Als Beispiel stellen wir die Beschriftung der Karte lakes.shp des QGIS Beispiel-Datensatzes vor:

1. Laden der Shapes alaska.shp und lakes.shp in QGIS
2. Zoomen Sie ein wenig in ein Gebiet mit ein paar Seen
3. Aktivieren die den lakes Layer
4. Öffnen Sie den Eigenschaften Dialog
5. Klicken Sie auf den "Beschriftungen" Reiter
6. Aktivieren Sie die Checkbox "Zeige Beschriftungen an"
7. Wählen Sie das Feld NAMES aus der Attributtabelle
8. Geben Sie eine Beschriftungsvorgabe für Seen ein, die keine Beschriftung im NAMES Feld enthalten.
9. Klicken Sie auf Anwenden

Nun werden Beschriftungen angezeigt, aber sie sind wahrscheinlich noch zu groß und schlecht platziert in Bezug auf die Objekte.

Klicken Sie auf den Knopf "Schriftstil" und ändern Sie Schrift und Farbe für die Beschriftung.

Um die Position der Beschriftungen zu ändern:

1. Klicken Sie auf den Knopf "Schrift Ausrichtung"
2. Ändern Sie die Ausrichtung durch auswählen der verschiedenen Knöpfe im Platzierungsfeld. In diesem Beispiel klicken auf den Knopf `Rechts`.
3. Klicken Sie wieder auf den Knopf `Anwenden`, um die Veränderungen zu sehen und schließen Sie dann den Dialog.

Nun sehen die Beschriftungen schon besser aus, aber Sie liegen noch ein wenig zu nah an den Objekten. Dies können Sie mit dem Reiter "Position" optimieren und eine X- und Y-Verschiebung einstellen. Mit einer X-Verschiebung mit dem Wert 5 sind die Beschriftungen wahrscheinlich schon besser lesbar. Ansonsten passen Sie den Wert entsprechend an.

Die letzte Anpassung der Beschriftung ist das Puffern. Dies bedeutet, dass hinter die einzelnen Beschriftungen ein Farbe gelegt wird, damit man sie besser lesen kann. Um die Beschriftungen der Karte `lakes.shp` zu Puffern tun Sie Folgendes:

1. Klicken Sie auf den Reiter "Puffer".
2. Aktivieren Sie den Knopf "Pufferbeschriftungen?".
3. Wählen Sie eine Größe für den Puffer.
4. Wählen Sie eine Farbe für den Puffer
5. Klicken Sie auf den Knopf `Anwenden` um das Ergebnis zu betrachten.

Wenn Sie mit dem Resultat noch nicht glücklich sind, dann verändern Sie die Einstellungen weiter, bis es Ihnen gefällt. Ein Puffer der Größe 2 gibt normalerweise recht ansehnliche Ergebnisse. Sie können die Puffergröße übrigens auch in Karteneinheiten definieren, wenn Ihnen das besser gefällt.

Als weitere Einstellungen im Reiter "Beschriftungen" ist es möglich, das Erscheinungsbild der Beschriftungen anhand von Angaben aus der Attributtabelle des Layers zu ändern. Der Reiter "Datendefinierter Stil" erlaubt es, alle notwendigen Parameter anhand der Attributspalten zu definieren.

#### 4.3.5 Aktionen auf Basis von Attributen

QGIS bietet die Möglichkeit, Aktionen auf Basis von Attributen einer Ebene durchzuführen. Dies kann für eine Vielzahl von Aktionen genutzt werden, z.B. um ein Programm mit Abfragen aus der Attributdatenbank zu füttern oder Ergebnisse einer Attributabfrage für ein statistisches Tool bereitzustellen.

Aktionen auf Basis von Attributen sind sinnvoll, wenn sie mit externen Applikationen auf Ergebnisse einer oder mehrerer Attributabfragen von Vektorlayern zugreifen möchten. Ein Beispiel finden Sie im nächsten Abschnitt.

### Aktionen definieren

Attribut-Aktionen werden im Dialog für die Ebeneneigenschaften im Reiter *Aktionen* definiert. Sie brauchen einen Namen für die Aktion und das externe Programm, das während der Aktion gestartet werden soll. Sie können ein oder mehrere Attributfelder einbinden. Alle Zeichen, die mit % und dem Namen eines Feldes beginnen, werden durch den Wert dieses Feldes ersetzt. Die Spezialfelder %% werden ersetzt durch den Wert des Feldes, der auf Basis von Abfrageergebnissen selektiert wird (vgl. Abschnitt *Aktionen anwenden*). Doppelte Anführungsstriche können dazu benutzt werden, um mehrere Worte als ein Ausdruck gruppiert zu übergeben. Doppelte Anführungsstriche werden ignoriert, wenn davor ein Backslash steht.

Zur besseren Erläuterung zwei Beispiele:

- `konqueror http://www.google.com/search?q=%nam`
- `konqueror http://www.google.com/search?q=%%`

Im ersten Beispiel wird an den Webbrowser Konqueror eine Google URL übergeben mit dem Wert eines Feldes `nam` eines Vektorlayers. Beachten Sie, dass sich das bei einer Aktion aufgerufene Programm im Pfad befinden muss. Ansonsten müssen Sie den gesamten Pfad angeben: (`/opt/kde3/bin/konqueror http://www.google.com/search?q=%nam`).

Das zweite Beispiel nutzt den Ausdruck `%%`, welcher unabhängig ist von einem speziellen Feld. Beim Ausführen des Befehls wird der Ausdruck `%%` durch den Wert des jeweils selektierten Feldes ersetzt.

### Aktionen anwenden

Aktionen können entweder über den Dialog *Abfrageergebnisse* oder die geöffnete *Attributtabelle* gestartet werden. Um eine Aktion aufzurufen, klicken Sie mit der rechten Maustaste auf einen Eintrag in der *Attributtabelle* und wählen die gewünschte Aktion aus der Liste aus.

Wenn eine Aktion mit dem Ausdruck `%%` gestartet werden soll, klicken Sie mit der rechten Maustaste auf einen Eintrag in der *Attributtabelle* oder im Dialog *Abfrageergebnisse* und wählen die gewünschte Aktion aus der Liste aus.

In einem weiteren Beispiel soll gezeigt werden, wie Attributwerte aus eines Vektorlayers abgefragt und in eine Textdatei mit Hilfe der Bash und des 'echo' Kommandos geschrieben werden (funktioniert also nur unter GNU/Linux und evtl. Mac OSX). Der Abfragelayer enthält die Felder `Art` (`taxon_name`), `Latitude` (`lat`) und `Longitude` (`long`). Um eine räumliche Abfrage eines Ortes zu erstellen und die Werte der drei Felder in eine Textdatei zu schreiben, wird folgende Aktion erstellt:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Nachdem ein paar Orte auf dem Bildschirm ausgewählt wurden (diese erscheinen gelb hinterlegt), starten wir die Aktion mit der rechten Maustaste über den Dialog *Abfrageergebnisse* und können

danach in der Textdatei die Ergebnisse ansehen:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

Als Übung erstellen wir eine Aktion, bei der Google nach dem Namen des jeweiligen Sees in dem Layer `lakes` sucht. Als erstes geben wir die URL ein, die wir brauchen, um auf Basis eines Schlüsselwortes zu suchen. Das ist einfach, in dem wir einfach auf der Google Webseite nach einem Begriff suchen und die URL dann kopieren. Das Format, das wir brauchen ist: `http://google.com/search?q=qgis`, wobei `qgis` das Schlüsselwort ist, nach dem wir suchen wollen. Nun kann es weiter gehen:

- Laden Sie den Layer `lakes.shp` in QGIS
- Öffnen Sie die Eigenschaften Dialog, indem Sie doppelt auf den Kartennamen `lakes` in der Legende klicken oder wählen Sie das Menü Eigenschaften mit der rechten Maustaste.
- Klicken Sie auf den Reiter "Aktionen"
- Geben Sie einen Namen für die Aktion ein, z.B: "Google Search"
- Für diese Aktion ist es notwendig, den Namen des externen Programms anzugeben, mit dem wir das Ergebnis anzeigen lassen wollen. In unserem Fall wählen wir Firefox als Webbrowser.
- Hinter dem Namen des Programms geben wir die URL ein, die wir für die Internetsuche benutzen wollen, aber ohne das Schlüsselwort: `http://google.com/search?q=`
- Der Text im Feld "Aktion" sollte nun folgendermaßen aussehen:  
`firefox http://google.com/search?q=`
- Klicken Sie nun auf die Drop-Down Box mit den Spaltennamen der Attributtabelle des Layers `lakes`. Der Knopf ist gleich rechts neben dem Knopf `Attribut einfügen`.
- Wählen Sie nun die Attributspalte `NAMES` und klicken auf `Attribut einfügen`
- Die Aktion sieht nun so aus:  
`firefox http://google.com/search?q=%NAMES`

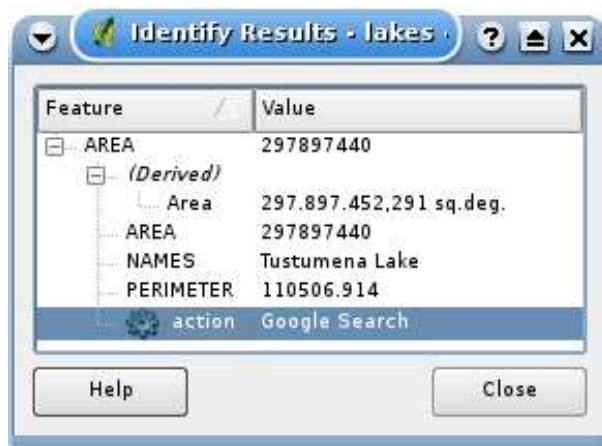
Damit ist die Aktion fertig für den Einsatz. Der gesamte Befehl der Aktion sollte folgendermaßen aussehen:

```
firefox http://google.com/search?q=%NAMES
```

Wir können die Aktion nun benutzen. Schließen Sie den Eigenschaften Dialog und zoomen Sie in eine Bereich ihrer Wahl mit ein paar Seen. Stellen Sie sicher, dass der Layer `lakes` in der Legende

aktiviert ist. Nun identifizieren Sie einen See. In der Ergebnisanzeige sollte nun die Aktion sichtbar sein:

**Abbildung 6:** Auswählen eines Objektes mit einer Aktion



Wenn wir nun auf das Wort “action” klicken, öffnet sich der Webbrowser Firefox und zeigt uns das Ergebnis der Internetrecherche z.B. nach dem See Tustumena an <http://www.google.com/search?q=Tustumena>. Es ist übrigens auch möglich, weitere Attributspalten zu ergänzen. Dazu fügen Sie einfach ein “+”-Zeichen an das Ende der Aktion, wählen eine weitere Attributspalte und klicken wieder auf den Knopf Attribut einfügen. In unserem Datensatz ist leider keine weitere sinnvolle Attributspalte vorhanden, nach der man im Internet suchen könnte.

Sie können auch mehrere Aktionen für einen Layer definieren. Sie alle werden dann bei der Abfrage von Objekten angezeigt und können ausgewählt und benutzt werden. Sie können auf eine Aktion starten, in dem Sie in der Attributtabelle in eine Zeile gehen und mit der rechten Maustaste eine Aktion für genau diesen einen Eintrag ausführen.

Sie sehen, man kann sich eine Vielzahl interessanter Aktionen ausdenken. Wenn Sie z.B. einen Punktlayer mit einzelne Punkten haben, an denen Photos geschossen wurden, dann könnten Sie eine Aktion erstellen, über die sie dann das entsprechende Photo anzeigen lassen können, wenn Sie auf den Punkt in der Karte klicken. Man könnte auch zu bestimmten Attributen webbasierte Information ablegen (z.B. in einer HTML-Datei) und diese dann über eine Aktion anzeigen lassen, etwa so wie in dem Google Beispiel.

## 4.4 Editieren

QGIS bietet einfache Editierfunktionen für räumliche Daten. Bevor Sie weiterlesen sollten Sie sich bewusst sein, dass derzeit nicht alle Funktionen ausgereift sind. Bevor Sie also anfangen, Vektorlayer zu editieren, sollten Sie die Daten sichern.

**Bemerkung** - die Vorgehensweise um GRASS-Ebenen zu editieren unterscheidet sich - vgl. Kapitel 8.5 für weitere Informationen dazu.

#### 4.4.1 Einstellen der Snapping-Toleranz

Vor dem Editieren von Stützpunkten ist es notwendig, die Snapping-Toleranz festzulegen. Dies ist die Distanz, die QGIS benutzt, um den nächstgelegenen Stützpunkt zu finden, den Sie editieren möchten, wenn Sie in die Karte klicken. Wenn Sie sich nicht innerhalb des Suchradius befinden, findet QGIS keine Stützpunkte zum editieren. Die Snapping-Toleranz wird in Karteneinheiten angegeben. Sie sollten ein wenig experimentieren, welcher Wert für Ihre Aufgabe am besten funktioniert. Wenn der Wert zu hoch ist, kann es passieren, dass QGIS beim Snappen nicht den richtigen Stützpunkt benutzt, besonders wenn die Anzahl vorhandener Stützpunkte in der Editierumgebung sehr hoch ist. Ist der Wert zu klein, wird nichts gefunden und eine Warnung erscheint.

Um die Snapping-Toleranz einzustellen wählen Sie **Projekteigenschaften** aus dem Menü **Einstellungen** und klicken auf den **Allgemein** Reiter. Beachten Sie, der Wert ist in Karteneinheiten. In unserem kleinen Digitalisierungsbeispiel sind die Einheiten in Dezimalgrad und der Wert sollte in etwa 0.05 bis 0.1 betragen.

#### 4.4.2 Editieren einer existierenden Ebene

Standardmäßig lädt QGIS Vektorlayer im 'read-only' Modus, um ungewolltes Editieren zu verhindern. Ansonsten können aber alle Ebenen editiert werden, wenn es der Datenanbieter erlaubt bzw. die Rechte entsprechend gesetzt sind.

Das Editieren von Vektorlayern ist am vielseitigsten bei PostgreSQL/PostGIS Daten.



---

#### **Tip 9** DATEN-INTEGRITÄT

Bitte beachten Sie, Daten vor und während des Editierens regelmäßig zu sichern, um bei Fehlern keine unnötigen Verluste zu erleiden.

QGIS befindet sich noch immer in einem Entwicklungsstadium unterhalb der Versionsnummer 1.0. Es kann daher noch zu unerwarteten Fehlern kommen.

---

Vor jedem Editieren eines Vektorlayers muss die Option **Editieren** erlauben aktiviert sein. Dazu klicken Sie mit der rechten Maustaste auf den Namen der Ebene in der Kartenlegende. Alternativ können Sie auch die Icons  **Starte die Digitalisierung für die aktuelle Ebene** oder  **Stoppe Digitalisierung für die aktuelle Ebene** aus der Werkzeugleiste benutzen.

Folgende Editierfunktionen stehen zur Verfügung:

- Objekt hinzufügen
- Verschiebe ausgewähltes Objekt



- Teile ausgewähltes Objekt
- Knotenpunkt zu einem Objekt hinzufügen
- Knotenpunkte eines Objektes verschieben
- Knotenpunkte eines Objektes löschen
- Ring hinzufügen
- Insel hinzufügen
- Selektiertes Objekt ausschneiden
- Selektiertes Objekt kopieren
- Selektiertes Objekt übergeben (paste)
- Selektiertes Objekt löschen

---

**Tip 10** DAS EDITIEREN VON GEOMETRIEN UNTERSCHIEDET SICH VOM EDITIEREN VON WERTEN IN DER ATTRIBUTTABELLE

---

In dieser QGIS Version agieren die Knöpfe zum Editieren von Geometrien in der Werkzeugleiste unabhängig von den Knöpfen *Digitalisieren starten* und *Bearbeitung stoppen* in der geöffneten Attributtabelle.

---

---

**Tip 11** REGELMÄSSIGES SICHERN DER DATEN

---

Bitte schließen Sie die Option *Editieren erlauben* oder klicken auf das Icon *Bearbeitung stoppen* regelmäßig. Dadurch werden alle bis dahin gemachten Änderungen gespeichert und es ist sichergestellt, dass die Änderungen auch akzeptiert werden.

---

---

**Tip 12** ZEITGLEICHES EDITIEREN

---

Diese QGIS Version kontrolliert nicht, ob zwei Personen gleichzeitig an einem Vektorlayer arbeiten. Die zuletzt schreibende Person gewinnt.

---

Wenn ein Vektorlayer im Editiermodus ist, erscheinen Kreuze für die Knotenpunkte der Geometrien im Kartenfenster.

### **Objekt hinzufügen**

Vor dem Editieren zoomen Sie in den Bereich, den Sie verändern möchten.

Danach können Sie mit den Icons *Punkt digitalisieren*, *Linie digitalisieren* oder *Polygon digitalisieren* in der Werkzeugleiste den Mauszeiger in den Digitalisiermodus bringen.

Für jedes Objekt wird erst die Geometrie digitalisiert und dann das Attribut hinzugefügt. Um eine Geometrie zu digitalisieren, klicken Sie mit der linken Maustaste an eine gewünschte Stelle im Kartenfenster, um den ersten Punkt zu erstellen.

Für Linien und Polygone klicken Sie für jeden weiteren Knotenpunkt wieder die linke Maustaste. Zum Beenden klicken Sie irgendwo im Kartenfenster auf die rechte Maustaste, um anzuzeigen, dass die Geometrie ihres Objektes fertig gestellt ist. Dadurch wird das Attributfenster geöffnet und ermöglicht

**Tip 13 ZOOMEN ZUM LETZTEN EDITIERSTAND**

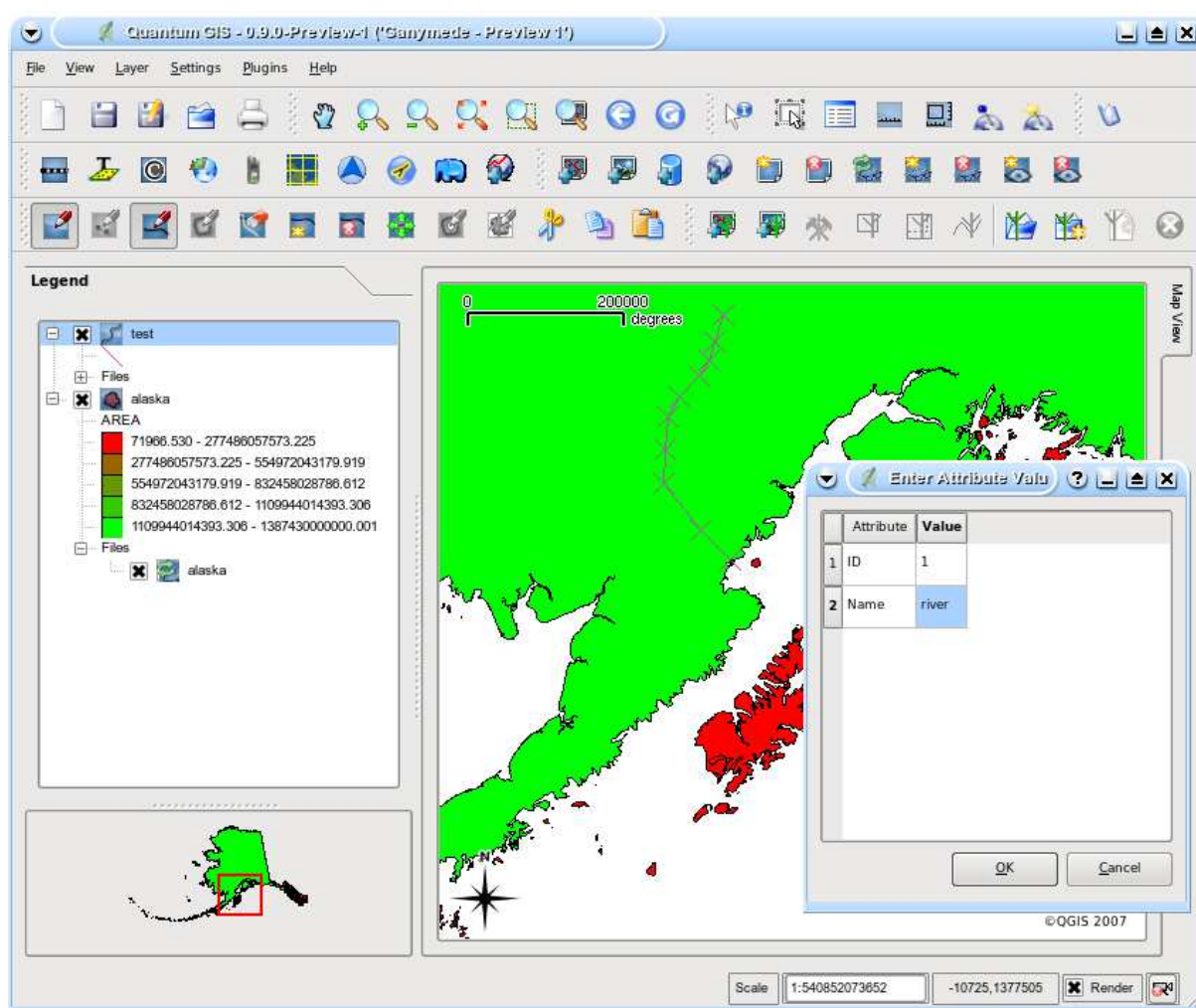
Vor dem Editieren eines Vektorlayers sollten Sie in den Bereich zoomen, den Sie verändern möchten. Dadurch wird die zeitaufwendige Darstellung sämtlicher Knotenpunkte des Vektorlayers im Kartenfenster verhindert.

**Tip 14 KNOTENPUNKT KREUZE**

Diese QGIS Version ermöglicht es nicht, das Kreuzsymbol für die Knotenpunkte zu ändern.

die Attribute für das neue Objekt einzutragen. In Abbildung 7 ist ein Beispiel für einen fiktiven Fluss in Alaska gezeigt.

**Abbildung 7:** Dialog zum Digitalisieren von Vektor-Attributen

**Knotenpunkte eines Objektes editieren****Neu in Version 0.9.1**

**Tip 15** DATENTYP DER ATTRIBUTE

Die aktuelle QGIS Version kontrolliert nicht, ob die eingetragenen Werte dem Datentyp der Attributspalte entspricht (z.B.: Zahl oder Text), Versichern Sie sich dessen, bevor Sie auf **OK** drücken, sonst werden später Fehler auftreten, wenn Sie versuchen, die Daten zu speichern und anderweitig zu nutzen.

Für PostgreSQL/PostGIS und Shape-basierte Vektorlayer können die Knotenpunkte von Objekten editiert werden.

Knotenpunkte können direkt editiert werden, ohne vorher auszuwählen, welches Objekt gemeint ist. In einigen Fällen teilen sich Objekte dieselben Knotenpunkte. Folgende Regeln kommen daher zum Einsatz:

- **Linien** - Die zum Mauszeiger nächste Linie wird als Zielobjekt benutzt. Beim verschieben oder löschen eines Knotenpunktes wird der zum Mauszeiger nächste Knotenpunkt auf einer Linie editiert.
- **Polygone** - Wenn der Mauszeiger sich innerhalb eines Polygons befindet, ist dies das Zielobjekt, ansonsten wird das zum Mauszeiger nächstliegende Objekt benutzt. Beim verschieben oder löschen eines Knotenpunktes wird der zum Mauszeiger nächste Knotenpunkt auf dem Polygon editiert.

Sie sollten den Suchradius beim Digitalisieren über die Menüleiste unter **Einstellungen->Projekt Eigenschaften->Allgemein->Suchradius** größer 0 einstellen. Ansonsten ist QGIS nicht in der Lage, ein Objekt auszuwählen.

**Knotenpunkt zu einem Objekt hinzufügen****Neu in Version 0.9.1**

Sie können einen Knotenpunkt zu einem Objekt hinzufügen, indem Sie das Icon **Knotenpunkt hinzufügen** aus der Werkzeugleiste auswählen. Beachten Sie, es macht wenig Sinn, weitere Knotenpunkte zu einem Punktobjekt zu ergänzen.

In dieser QGIS Version können Knotenpunkte nur zu bereits vorhandenen Liniensegmenten einer Linie hinzugefügt werden. Wenn Sie eine Linie über ihr Ende hinaus verlängern möchten, müssen Sie erst den letzten Knotenpunkt verschieben und dann weitere Knotenpunkte ergänzen.

**Knotenpunkte eines Objektes verschieben****Neu in Version 0.9.1**

Sie können Knotenpunkte mit dem Icon **Knotenpunkt verschieben** aus der Werkzeugleiste verschieben.

## Knotenpunkte eines Objektes löschen

### Neu in Version 0.9.1

Sie können einen Knotenpunkt aus einem Objekt löschen, indem Sie das Icon **Knotenpunkt löschen** aus der Werkzeugleiste auswählen. Beachten Sie, es macht keinen Sinn, einen Knotenpunkt aus einem Punktobjekt zu löschen. Löschen Sie in dem Fall das Objekt. Ebenso ist eine Linie, die aus einem Knotenpunkt besteht oder ein Polygon, das aus zwei Knotenpunkten besteht sinnlos und führt zu fehlerhaften Ergebnissen.

**Warnung:** Ein Knotenpunkt ist zum löschen ausgewählt, sobald Sie mit der Maus in die Nähe des Knotenpunktes klicken. Um dies rückgängig zu machen, müssen Sie den Editiermodus verlassen und bei der Nachfrage, ob die Veränderungen gespeichert werden sollen **Nein** auswählen (Dadurch werden natürlich alle bis dahin vorgenommenen Änderungen verworfen.)

## Ausschneiden, kopieren und übergeben von Objekten

Ausgewählte Objekte können ausgeschnitten, kopiert und an andere Ebenen im aktuellen QGIS-Projekt übergeben werden, wenn der Ziellayer sich auch im Editiermodus befinden.

Objekte können auch als Text an externe Applikationen übergeben werden: Dabei wird ein Objekt im .csv Format mit ihrer Geometrie im OGC Well-Known Text (WKT) Format erzeugt.

In dieser QGIS Version können Textobjekte außerhalb von QGIS nicht in eine geladene Ebene eingefügt werden.

Wann macht es Sinn, die Kopier- und Ausschneidefunktion zu benutzen? Ein Beispiel ist, wenn Sie parallel an mehreren Layern arbeiten und Objekte zwischen den Layern hin- und herkopieren möchten. Ein Szenario könnte sein, dass Sie einen neuen Layer erstellen möchten, in dem aber nur einige Objekte aus einem bereits existierenden Layer verwendet werden sollen, wie etwa 5 Seen aus der Karte lakes.shp, die insgesamt tausende Seen enthält.

Wir nutzen diese Idee gleich und machen daraus ein kleines Anwendungsbeispiel:

1. Laden Sie den Layer, von dem Sie einige Objekte kopieren wollen (Quelle)
2. Laden oder erstellen einen Layer, in den die kopierten Objekte eingefügt werden sollen (Ziel)
3. Schalten Sie für beide Layer den Bearbeitungsmodus ein
4. Stellen Sie die "Quelle" aktiv, indem Sie es in der Legende anklicken
5. Benutzen Sie das "Wähle Objekt aus" Tool und wählen ein paar Objekte in der Quelle aus
6. Klicken Sie auf das Icon **Objekte kopieren**
7. Stellen Sie das "Ziel" aktiv, indem Sie es in der Legende anklicken
8. Klicken Sie auf das Icon **Objekte einfügen**
9. Beenden Sie den Bearbeitungsmodus für beide Layer und speichern Sie das Ergebnis

Was passiert, wenn der Quell- und Ziellayer ein unterschiedliches Schema enthält (Spaltennamen und -typen unterscheiden sich)? QGIS verwendet die Einträge, die gleich sind und ignoriert den Rest. Wenn es Ihnen egal ist, ob die Attribute korrekt übernommen werden, dann ist es egal, wie Sie die Spaltennamen und -typen der Attributtabelle erstellen. Wenn auch die Attributdaten korrekt übernommen werden sollen, dann stellen Sie sicher, dass auch die Spaltennamen und -typen beider Layer zueinander passen.

---

**Tip 16 DECKUNGSGLEICHHEIT ÜBERGEBENER OBJEKTE**

---

Wenn der Quell- und Ziellayer die gleiche Projektion nutzen, hat das übergebene Objekt die identische Geometrie wie in dem Quelllayer. Wenn der Quell- und Ziellayer eine unterschiedliche Projektion nutzen, kann das übergebene Objekt eine andere Geometrie wie in dem Quelllayer besitzen. Der Grund liegt in auftretenden Rundungsabweichungen bei der Konvertierung zwischen Projektionen.

---

### Löschen ausgewählter Objekte

Wenn Sie ein ganzes Polygon löschen möchten, selektieren Sie es erst mit dem "Wähle Objekt aus" Tool. Sie können dabei auch mehrere Objekte gleichzeitig auswählen. Wenn Sie die Objekte ausgewählt haben, klicken Sie auf das Icon *Ausgewähltes löschen*. Es gibt keine Funktion, um das Löschen wieder rückgängig zu machen. Aber bedenken Sie, dass die Änderungen erst dann wirksam werden, nachdem Sie den Bearbeitungsstatus verändert und die Veränderungen gespeichert haben. Wenn Sie also einen Fehler gemacht haben, können Sie einfach das Speichern abbrechen.

Das *Selektierte Objekte ausschneiden* Tool kann auch benutzt werden, um Objekte zu löschen. Die Objekte werden gelöscht aber zusätzlich noch im "spatial clipboard" abgelegt. In diesem Fall könnte man dann den letzten Schritt, falls uns dabei ein Fehler unterlaufen ist, wieder rückgängig machen, indem wir auf das Icon *Objekte einfügen* klicken. Ausschneiden, kopieren und übergeben von Objekten funktioniert mit den gerade ausgewählten Objekten, können als nach Bedarf und kombiniert verwendet werden.

---

**Tip 17 UNTERSTÜTZUNG ZUM LÖSCHEN VON OBJEKTEN**

---

Wenn Sie Shapes editieren wird das Löschen von Objekten nur unterstützt, wenn QGIS zu einer GDAL Version 1.3.2 oder höher verlinkt ist. Die OS X, Windows und GNU/Linux Versionen von QGIS auf der Downloadseite von QGIS wurden mit GDAL Version 1.3.2 oder höher erstellt.

---

### Suchradius einstellen

QGIS ermöglicht es, digitalisierte Knotenpunkte an andere Knotenpunkte derselben Ebene anzubinden. Dazu müssen Sie den Suchradius als Karteneinheit in der Menüleiste unter *Einstellungen->Projekt Eigenschaften->Allgemein->Suchradius* entsprechend einstellen.

### Editierbare Ebenen speichern

Wenn sich ein Vektorlayer im Editiermodus befindet, werden alle Änderungen im Speicher von QGIS abgelegt. Sie werden also nicht direkt auf den Datenträger oder die Festplatte geschrieben. Wenn Sie

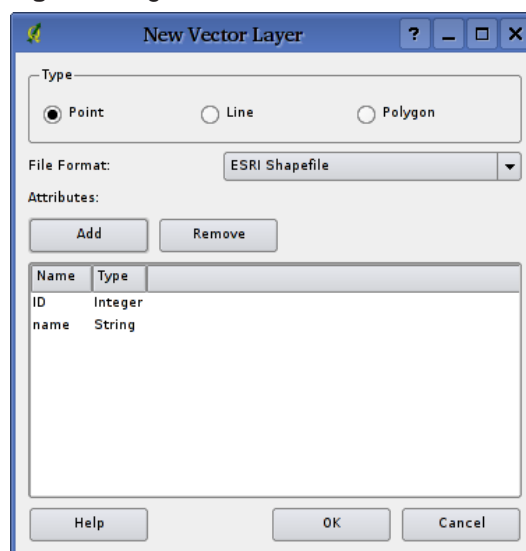
den Editiermodus oder QGIS selbst beenden, werden Sie gefragt, ob sie die aktuellen Änderungen speichern möchten.

Wenn die Änderungen nicht gespeichert werden können (z.B.: weil die Festplatte voll ist) bleibt der QGIS 'im-memory' Status erhalten. Sie können somit Veränderungen vornehmen und das Speichern erneut versuchen.

#### 4.4.3 Neue Vektorlayer erstellen

Um einen neuen Vektorlayer zu erstellen, wählen Sie in der Menüleiste **Neuer Vektorlayer** aus dem Menü **Ebene**. Dadurch startet der **Neuer Vektorlayer Dialog** (vgl. Abb. 8). Wählen Sie hier den Typ der neuen Ebene (Punkt, Linie oder Polygon).

**Abbildung 8:** Dialog zum Erstellen eines neuen Vektorlayers



QGIS kann derzeit noch keine 2,5D Objekte erstellen (z.B. Objekte mit X,Y,Z Koordinaten). Im Moment können nur Shapes erstellt werden. In zukünftigen Versionen werden auch andere durch OGR oder PostgreSQL unterstützte Datenformate erstellt werden können.

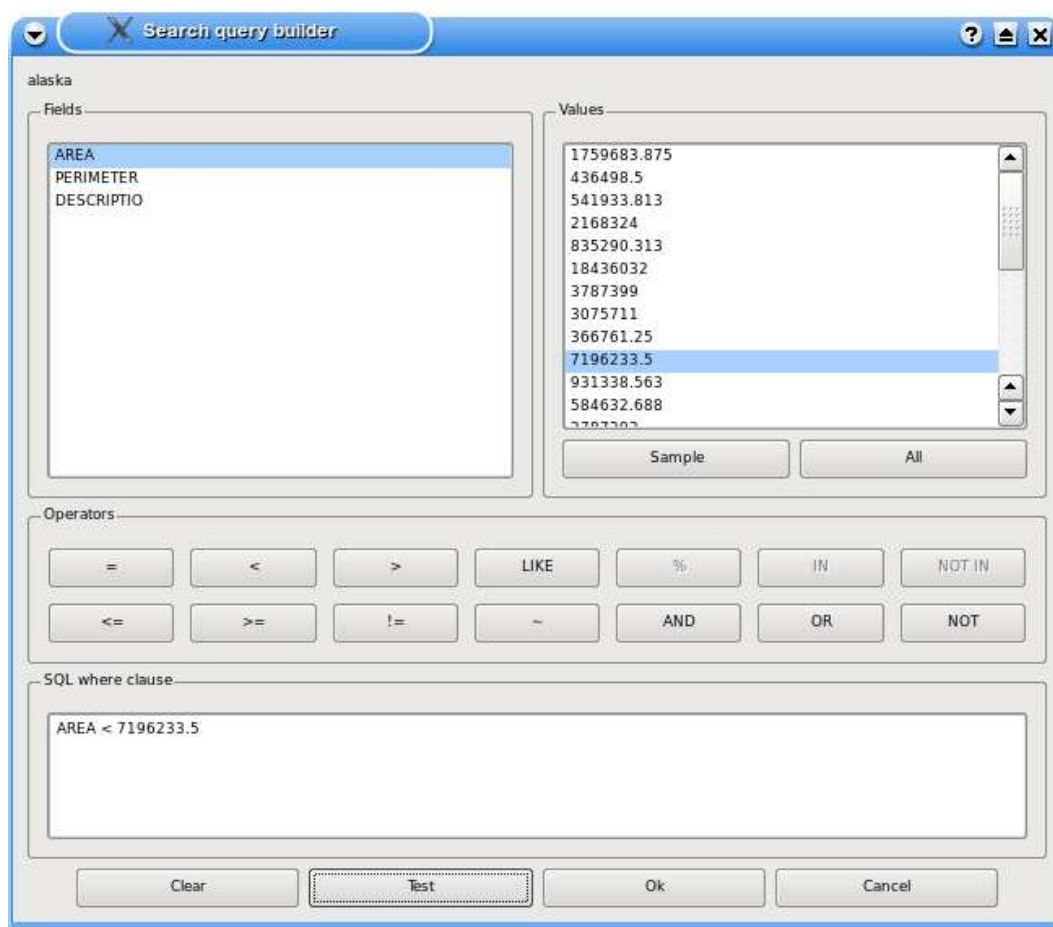
Das Erstellen eines GRASS Vektorlayers ist mit dem GRASS-Plugin möglich. Weitere Information dazu finden Sie im Abschnitt 8.8.

Um die Erstellung eines neuen Vektorlayers abzuschließen, ergänzen Sie die gewünschten Attributfelder, indem Sie auf den Knopf **Hinzufügen** klicken und den Namen und Datentyp angeben. Es werden nur die Datentypen `real`, `integer` und `string` unterstützt. Wenn dies getan ist, bestätigen Sie mit **OK** und geben einen Namen für den Vektorlayer ein. Die neue Ebene wird automatisch in die Kartenlegende eingefügt und Sie können diese entsprechend dem Abschnitt 4.4.2 editieren.

## 4.5 Query Builder

Der Query Builder ermöglicht es, eine Auswahl einer Attributtabelle zu erstellen und die dazu gehörigen Geometrien im Kartenfenster anzuzeigen. Er kann für alle OGR-unterstützten Formate, GRASS- und PostGIS-Ebenen verwendet werden. Wenn Sie beispielsweise eine Ebene 'Städte' haben, können Sie alle Städte mit einer Einwohnerzahl > 100000 auswählen und anzeigen lassen. Abbildung 9 zeigt eine Beispielanwendung des Query Builder mit einer PostGIS-Ebene.

Abbildung 9: Query Builder



Der Query Builder listet die vorhandenen Attributspalten eines Vektorlayers in der linken oberen Box (Felder). Sie können einen Ausschnitt der vorhandenen Attributwerte im rechten oberen Feld (Werte) anzeigen, indem Sie eine Attributspalte auswählen und auf den Knopf *Stichprobe* klicken. Dadurch werden die ersten 25 unterschiedlichen Werte der Attributspalte angezeigt. Um eine vollständige Liste zu bekommen, klicken Sie auf den Knopf *All*. Um eine Attributspalte in das untere Fenster (SQL where clause) zu kopieren, doppelklicken sie auf den Namen in der Box (Felder). Um eine Abfrage zu erstellen, können Sie die verschiedenen Operatoren benutzen.



Um eine Abfrage zu testen, klicken Sie auf den Knopf **Test**. Dies erzeugt ein Fenster, in dem die Anzahl der für diese Abfrage zutreffenden Objekte angezeigt wird. Nach erfolgreichem Testen klicken Sie auf den Knopf **Ok**. Das SQL-Kommando wird in der SQL-Spalte der Ebenenliste angezeigt werden.

---

**Tip 18** ÄNDERN DER LAYER-DEFINITION

---

Sie können die Definition der Ebene ändern nachdem sie geladen ist, indem sie die SQL-Abfrage anders starten. Dazu öffnen Sie den Dialog Ebeneneigenschaften durch Doppelklicken auf den Ebenennamen in der Kartenlegende. Im Reiter **Allgemein** befindet sich der Knopf **Query Builder**. Weitere Informationen können Sie im Abschnitt 4.3 finden.

---

#### 4.5.1 PostGIS-Ebenen abfragen

Um eine geladene PostGIS-Ebene abzufragen, gibt es zwei Optionen. Die erste besteht darin, auf den Knopf **Öffne Attributtabelle** in der Werkzeugleiste zu klicken und im Dialogfenster auf den Knopf **Erweitert**. Dadurch wird der Query Builder gestartet und ermöglicht Abfragen wie in Abschnitt 4.5 beschrieben.

Eine zweite Option besteht darin, das Dialogfenster Ebeneneigenschaften zu öffnen, indem Sie mit der rechten Maustaste auf den Ebenennamen in der Kartenlegende klicken und unter **Eigenschaften** im Reiter **Allgemein** den Knopf **Query Builder** auswählen.

#### 4.5.2 OGR- und GRASS-Ebenen abfragen

Um eine geladene OGR- oder GRASS-Ebene abzufragen, klicken Sie auf den Knopf **Öffne Attributtabelle** in der Werkzeugleiste und im Dialogfenster auf den Knopf **Erweitert**. Dadurch wird der Query Builder gestartet und ermöglicht Abfragen wie in Abschnitt 4.5 beschrieben.

Die zweite Option, wie im Abschnitt PostGIS-Ebenen 4.5.1 beschrieben, wird in der aktuellen Version nicht unterstützt.



---

## 5 Arbeiten mit Rasterdaten

QGIS unterstützt eine Vielzahl von Rasterformaten. Dieses Kapitel beschreibt die Arbeit mit Rasterdaten in QGIS.

### 5.1 Was ist eine Rasterdatei?

Rasterdaten in QGIS sind Gitter aus diskreten Zellen oder Pixeln, die Objekte über, auf oder unter der Erdoberfläche beschreiben. Jede Zelle eines Rasters hat die gleiche Größe und ist meistens rechteckig (in QGIS ist dies immer so). Typische Rasterdaten sind Fernerkungsdaten wie Luft- oder Satellitenbilddaten oder modellierte Daten wie Höhenmodelle.

Im Gegensatz zu Vektordaten haben Rasterdaten keine Verknüpfung zu einer Attributtabelle mit Werten für jede Zelle.

In Geographischen Informationssystemen (GIS) ist ein Raster eine georeferenzierte Datei mit Informationen, die räumlich korrekt mit anderen Vektorlayern oder Rasterdaten überlagert werden kann. QGIS kann Rasterdaten also korrekt visualisieren aber nicht analysieren.

### 5.2 Durch QGIS unterstützte Rasterformate

QGIS unterstützt eine Vielzahl von Rasterformaten. Aktuell getestete Formate sind:

- Arc/Info Binary Grid
- Arc/Info ASCII Grid
- GRASS Raster
- GeoTIFF
- Spatial Data Transfer Standard Grids (mit einigen Einschränkungen)
- USGS ASCII DEM
- Erdas Imagine

Da die Rasterimplementation in QGIS auf der GDAL-Bibliothek basiert, ist es wahrscheinlich, dass auch andere durch GDAL unterstützte Rasterformate unterstützt aber noch nicht getestet wurden. Vergleichen Sie dazu A.2 für weitere Details.

### 5.3 Rasterdaten in QGIS laden



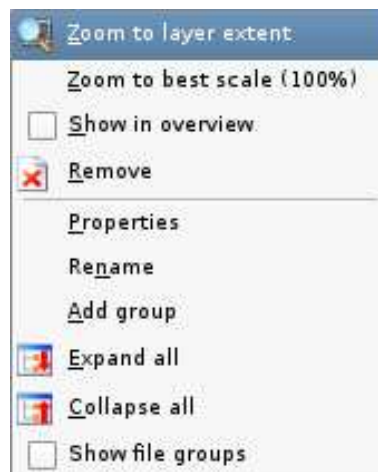
Rasterdaten können durch klicken auf das Füge einen Rasterlayer hinzu Icon oder über die Menüleiste unter Layer -> Füge einen Rasterlayer hinzu geladen werden. Mehrere Raster können gleichzeitig geladen werden, indem die Control-Taste bei der Auswahl der Karten gedrückt gehalten wird.

Informieren Sie sich in Abschnitt 8.2 über das Darstellen von GRASS Rasterdaten.

### 5.4 Eigenschaften von Rasterdaten

Um die Optionen von Rasterdaten anzusehen oder anzupassen, klicken Sie mit der rechten Maustaste auf den Layernamen in der Kartenlegende. Dadurch erscheint ein Popupmenue für Rasterlayer und bietet eine Vielzahl von Optionen:

**Abbildung 10:** Popupmenü für Rasterlayer

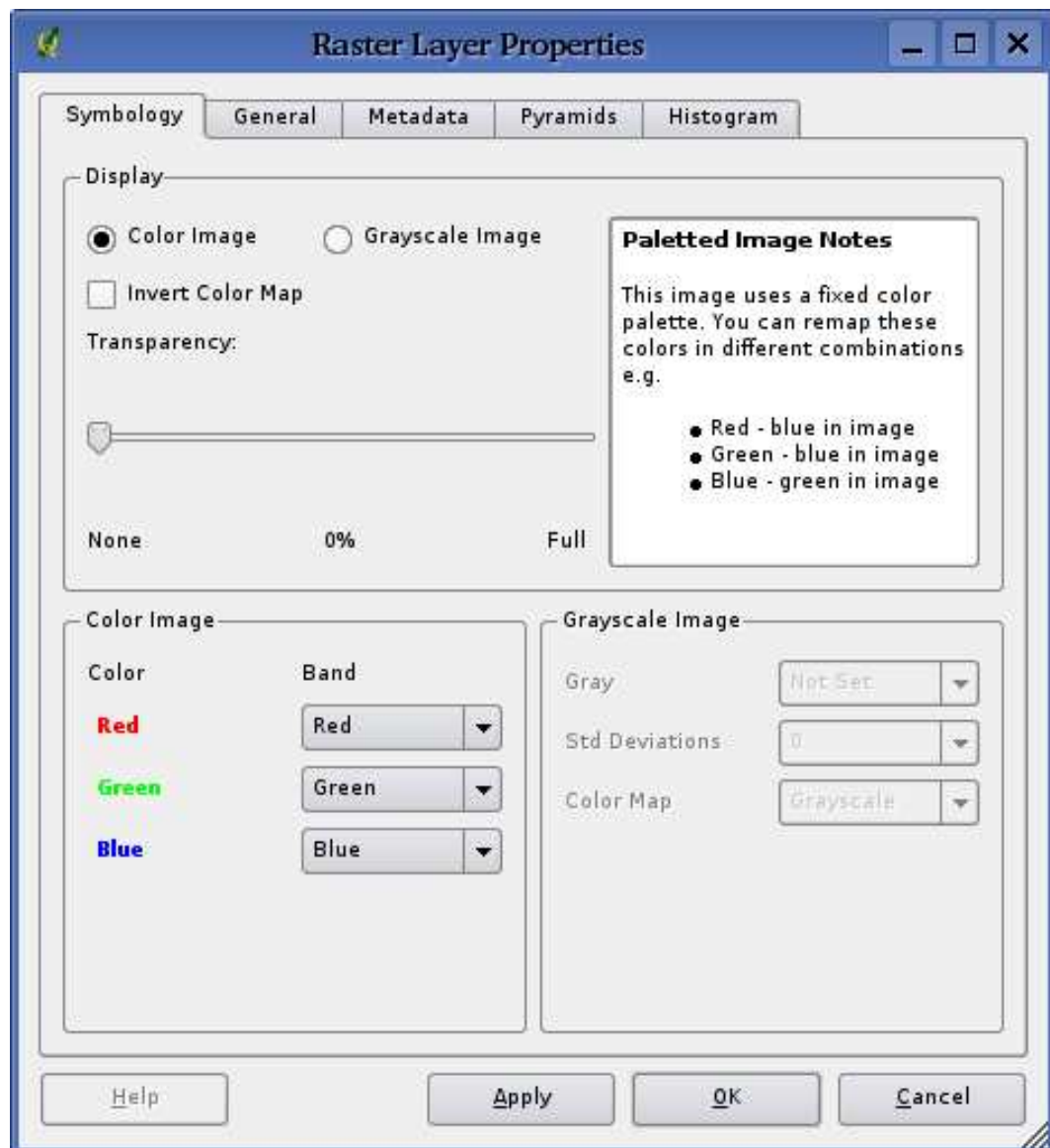


- Auf die Layerausdehnung zoomen
- Auf den besten Maßstab zoomen
- In der Übersicht anzeigen
- Entfernen
- Eigenschaften
- Umbenennen
- Gruppe hinzufügen
- Alles ausklappen
- Alles zusammenfallen

- Zeige Dateigruppen

Wähle Eigenschaften aus dem Popupmenü, um die Rasterlayer-Eigenschaften anzuzeigen. Abbildung 11 zeigt das Dialogfenster mit den Rastereigenschaften. Es gibt fünf Reiter: Bezeichnungen, Generell, Metadaten, Pyramiden und Histogramm.

Abbildung 11: Dialog mit den Rastereigenschaften



#### 5.4.1 Bezeichnungen

QGIS unterstützt drei Arten von Rasterlayer:

- Einkanal Graustufen Raster
- Farbbild RGB Raster
- Mehrkanal RGB Raster

Aus diesen drei Layertypen können acht Arten von Rastern dargestellt werden:

- Einkanal Graustufen
- Einkanal Pseudofarbe
- Graustufen Palette (nur die Rot-, Grün- oder Blaukomponenten wird angezeigt)
- Pseudofarben Palette (nur die Rot-, Grün- oder Blaukomponenten wird angezeigt mit Hilfe eines Pseudofarben-Algorithmus)
- RGB Palette
- Mehrkanal Graustufen (nur ein Kanal wird angezeigt)
- Mehrkanal Pseudofarben (nur ein Kanal wird angezeigt in Pseudofarben angezeigt)
- Mehrkanal RGB (ein Kombination der 3 Kanäle wird angezeigt)

QGIS kann die Farbwerte eines Layers invertieren, so dass helle Farben dunkel und dunkle Farben hell werden. Benutzen Sie dazu die `Farben invertieren` Checkbox, um diese Eigenschaft zu aktivieren oder zu deaktivieren.

QGIS kann Rasterlayer in unterschiedlicher Transparenz darstellen. Benutzen Sie dazu den `Transparenz Schieber`, um festzulegen, wieviel unter dem aktuellen Rasterlayer sichtbar sein soll.

QGIS kann so eingestellt werden, dass nur Rasterzellen gezeigt werden, die Werte innerhalb einer definierten Standardabweichung zum Mittelwert des Layers enthalten. Dies ist nützlich, wenn man Daten anzeigen lassen möchte, bei denen einzelne Zellen extrem vom Mittelwert abweichende Werte haben, was einen negativen Einfluss auf die Darstellung in QGIS hat. Diese Option ist nur für Pseudofarben Bilder anwendbar.

---

**Tip 19** EINEN EINZELNEN KANAL EINES MEHRKANAL-RASTERLAYERS ANZEIGEN

---

Wenn Sie einen einzelnen Kanal (z.B.: den Rot-Kanal) eines Mehrkanal-Rasterlayers sehen möchten, könnten Sie denken, dass Sie dazu den Wert für den Blau- und Grünkanal auf 'Nicht gesetzt' stellen sollten. Dies ist aber nicht der richtige Weg. Setzen Sie hierzu den Bildtyp auf Graustufen und wählen dann Rot als Kanal, den Sie als Graustufenbild anzeigen möchten.

---

### 5.4.2 Allgemein

Im Reiter `Allgemein` werden allgemeine Informationen zum ausgewählten Raster angezeigt, wie dem Namen und Quellpfad des Rasters (kann geändert werden). Außerdem wird ein Thumbnail der Karte gezeigt, die Legende und die Farbpalette.

Zusätzlich kann eine skalenabhängige Sichtbarkeit eingestellt werden. Dazu muss die Checkbox *Skalenabhängige Sichtbarkeit* aktiviert und ein entsprechender Maßstab eingetragen werden.

Im Feld *räumliches Referenzsystem* kann die Projektion abgelesen und mit dem Knopf *Wechseln* verändert werden.

### 5.4.3 Metadaten

Der Reiter *Metadaten* zeigt eine Anzahl von Informationen über das Raster, inklusive Statistiken über jeden Kanal innerhalb des Layers. Statistiken werden nach dem Prinzip 'was brauche ich' erstellt. Es kann also gut sein, dass für einen Rasterlayer keine Statistik zur Verfügung steht oder gesammelt wurde.

---

**Tip 20** RASTER-STATISTIKEN SAMMELN

---

Um statistische Daten zu einem Layer zu sammeln, wählen Sie *Pseudofarbe* und klicken auf *Anwenden*. Dieser Vorgang kann zeitaufwendig sein. Bitte seien Sie geduldig, während QGIS Daten sammelt!

---

### 5.4.4 Pyramiden

Hochaufgelöste Rasterdaten können das Navigieren in QGIS verlangsamen. Indem geringer aufgelöste Kopien (Pyramiden) erstellt werden, kann die Darstellung optimiert werden. QGIS wählt dann entsprechend des Zoom-Levels die passende Auflösung.

Sie brauchen dazu Schreibrecht in dem Ordner, in dem sich die Originaldaten befinden. Zwischen verschiedenen Resampling-Methoden kann gewählt werden:

- Durchschnitt
- Nächster Nachbar

Beachten Sie, dass das Erstellen von Pyramiden die Originaldaten verändern und dieser Schritt nicht mehr rückgängig gemacht werden kann. Wenn Sie eine 'nicht pyramidierte' Version der Daten wünschen, erstellen Sie vorher eine Kopie.

### 5.4.5 Histogramm

Der Histogramm-Reiter erlaubt es, ein Histogramm der Werteverteilung eines Rasters zu erstellen, und damit die Pixelverteilung darzustellen. Es gibt zwei Arten der Visualisierung: Balkendiagramm und Liniengrafik. Dazu müssen Sie erst auf den Knopf *Erneuern* klicken. Dann wählen Sie die Kanäle aus, die angezeigt werden sollen.

Die Berechneten Werte sind nun übrigens auch in dem Reiter *Metadaten* ergänzt worden.

---

## 6 Mit OGC-Daten arbeiten

QGIS unterstützt WMS und WFS als Datenquellen. Die Unterstützung für WFS ist zum jetzigen Zeitpunkt noch experimentell. Die WMS-Unterstützung ist nativ, die WFS-Unterstützung als Plugin integriert.

### 6.1 Was sind OGC-Daten?

Das Open Geospatial Consortium (OGC) ist eine internationale Organisation mit mehr als 300 Mitgliedern aus kommerziellen und behördlichen Bereichen, aus der Forschung sowie aus Vereinen (Non-Profit). Die Mitglieder entwickeln und implementieren Standards für den Austausch räumlicher Daten (Datendienste), GIS-Datenprocessing und standardisierte Bereitstellung von Geodaten.

Zur Beschreibung von geographischen Objekten in einem einfachen Datenmodell wurden eine steigende Zahl von Spezifikationen entwickelt, die spezielle Bedürfnisse der Interoperabilität bedienen, räumliche Informationen und GIS einbezogen. Weitere Informationen können unter <http://www.opengeospatial.org/> abgerufen werden.

Wichtige OGC-Spezifikationen sind:

- **WMS** - Web Map Service
- **WFS** - Web Feature Service
- **WCS** - Web Coverage Service
- **CAT** - Web Catalog Service
- **SFS** - Simple Features for SQL
- **GML** - Geography Markup Language

OGC-Dienste werden vermehrt zum Austausch von geographischen Daten zwischen unterschiedlichen GIS-Systemen und -implementierungen verwendet. QGIS unterstützt mittlerweile drei der oben genannten Spezifikationen; einerseits SFS (durch den PostgreSQL/PostGIS Datenprovider, vgl. 4.2) sowie als WFS und WMS als Klient.

### 6.2 WMS-Klient

#### 6.2.1 Übersicht über die WMS-Unterstützung



Derzeit kann QGIS als WMS-Klient eingesetzt werden. Es unterstützt die Versionen 1.1, 1.1.1 und 1.3 der WMS-Server. Gut getestet wurden die öffentlich verfügbaren Server wie beispielsweise DEMIS oder NASAs JPL OnEarth-Server.

WMS-Server liefern Daten aufgrund einer Anfrage eines Klienten (hier QGIS) als Rasterbild aus. Dabei spielen Ausdehnung, Anzahl der angefragten Layer, Symbolisierungen und Transparenz eine Rolle. Der WMS-Server holt die benötigten Daten dann aus seiner Datenquelle hervor, rendert diese in eine Rasterkarte und sendet das fertige Bild zurück zum Klienten. Das für QGIS typische Rasterformat ist in aller Regel JPEG oder PNG.

WMS ist ein komplett auf Übertragung ausgelegter Dienst (REST = Representational State Transfer). Daraus resultiert die Tatsache, dass die von QGIS generierte URL für das Bild auch in einem Browser eingesetzt werden kann. Das Resultat dieser Anfrage sieht in der Regel genauso aus wie in QGIS. Das ist besonders hilfreich, wenn es beim Einsatz von WMS Probleme geben sollte. Da es sehr viele unterschiedle WMS-Server-Anbieter am Markt gibt (und alle die WMS-Spezifikation etwas unterschiedlich interpretieren), ist eine Überprüfung im Browser sehr hilfreich.

WMS-Layer können sehr einfach hinzugefügt werden, solange man die URL des Servers kennt, eine Verbindung über HTTP zu diesem Server besteht und der angefragte Server auch HTTP versteht.

### 6.2.2 WMS-Server auswählen

Sie starten die Funktion, indem Sie auf das Icon *WMS-Layer hinzufügen* in der Werkzeugleiste klicken oder im Menü *Layer* die Funktion *WMS-Layer hinzufügen* auswählen.

Der Dialog zum Hinzufügen von WMS-Layern erscheint dann. Beim ersten Laden eines WMS-Layers sind noch keine Server definiert. Sie können aber einige vordefinierte Server hinzufügen, indem Sie auf den Knopf *Standardserver ergänzen* klicken. Dadurch werden mindestens 3 Server hinzugefügt (z.B. der NASA WMS-Server). Um einen neuen Server selbst zu definieren klicken Sie auf *Neu* und geben dann die Parameter, die zur Verbindung benötigt werden, ein (vgl. Tabelle 2):

**Tabelle 2:** WMS Verbindungs-Parameter

Name	Ein Name für diese Verbindung. Dieser Name wird in der Auswahlbox der vorhandenen WMS-Server erscheinen.
URL	Die URL des Servers, der die Daten vorhält. Dieser Name muss ein auflösbarer Host-Name sein.
Proxy Host	Netzwerk-Adresse oder Rechnername des Proxy-Servers. Falls Sie keinen Proxy benötigen, lassen Sie das Feld leer.
Proxy Port	Portnummer des Proxy-Servers.
Proxy User	Benutzername zum Einloggen am Proxy-Server.
Proxy Passwort	Passwort zum Einloggen am Proxy-Server.

Es müssen mindestens der *Name* sowie die *URL* eingetragen werden. Die Proxy-Einträge können leer gelassen werden, wenn sie keinen Proxy benötigen.

Ein einmal erstelltes WMS-Profil wird nun für zukünftige QGIS-Sessions abgespeichert.

**Tip 21 WMS-SERVER-URLS**

Stellen Sie sicher, dass Sie die Basis-URL des Servers eingeben. Achten Sie darauf, dass keine Fragmente wie `request=GetCapabilities` oder `version=1.0.0` in der URL enthalten sind.

Tabelle 3 zeigt beispielhaft einige WMS-URLs, damit Sie einen Startpunkt haben. Diese Links wurden im Dezember 2006 überprüft, sie können sich jedoch jederzeit ändern.

**Tabelle 3:** Beispielhafte öffentliche WMS-URLs

Name	URL
Atlas von Kanada	<a href="http://atlas.gc.ca/cgi-bin/atlaswms_en?">http://atlas.gc.ca/cgi-bin/atlaswms_en?</a>
DEMIS	<a href="http://www2.demis.nl/wms/wms.asp?wms=WorldMap&amp;">http://www2.demis.nl/wms/wms.asp?wms=WorldMap&amp;</a>
Geoscience Australia	<a href="http://www.ga.gov.au/bin/getmap.pl?dataset=national">http://www.ga.gov.au/bin/getmap.pl?dataset=national</a>
NASA JPL OnEarth	<a href="http://wms.jpl.nasa.gov/wms.cgi?">http://wms.jpl.nasa.gov/wms.cgi?</a>
QGIS Users	<a href="http://qgis.org/cgi-bin/mapserv?map=/var/www/maps/main.map&amp;">http://qgis.org/cgi-bin/mapserv?map=/var/www/maps/main.map&amp;</a>

Eine ausführliche Liste von öffentlich verfügbaren WMS-Server können Sie unter <http://wms-sites.com> finden.

### 6.2.3 Eine WMS-Layer laden

Nachdem Sie erfolgreich alle benötigten Parameter eingegeben haben können Sie den *Verbinden*-Knopf drücken und die Capabilities des Server holen. Daraus liest QGIS das unterstützte Bildformat, die verfügbaren Layer, die Stile sowie die verfügbaren Projektionen. Da dies eine Netzwerkoperation ist, hängt die Geschwindigkeit der Antwort des Servers von Ihrer Netzwerkgeschwindigkeit ab. Während die Daten vom Server heruntergeladen werden, wird der Fortschritt in der linken unteren Ecke des WMS-Plugin Dialogs angezeigt.

Ihr Bildschirm sollte in etwa so wie der in Abbildung 12 aussehen. Dort ist die Antwort des NASA JPL OnEarth WMS-Servers exemplarisch dargestellt.

#### Bildkodierung

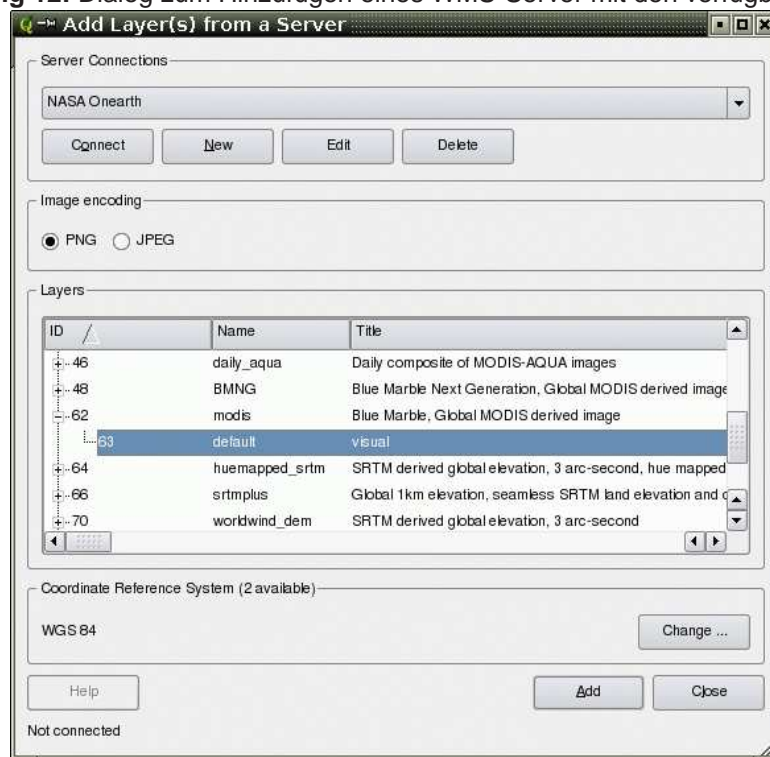
Der Abschnitt über die *Bildkodierung* gibt die Bildformate an, die der Server und QGIS unterstützen. Wählen Sie ein Format aus, das ihren Genauigkeits-Bedürfnissen entspricht.

#### Layer

Der Abschnitt der *Layer* listet alle verfügbaren Layer des WMS-Servers auf. Sie werden bemerkt haben, dass einige Layer aufklappbar sind, was bedeutet, dass der WMS-Server für diese Layer unterschiedliche Stile bereitstellt.

Sie könnten mehrere Layer auf einmal auswählen, jedoch nur einen Stil pro Layer. Wenn mehrere



**Abbildung 12:** Dialog zum Hinzufügen eines WMS-Server mit den verfügbaren Layer**Tip 22 BILDKODIERUNG**

In der Regel bieten WMS-Server JPEG oder PNG als Bildkodierung an. JPEG hat eine bildverschlechternde Kompression, während PNG zumeist die Qualität der ursprünglichen Rasterdaten widerspiegelt.

Benutzen Sie JPEG wenn Ihre WMS-Daten photographischer Natur (Luftbilder/Orthophotos) sind und/oder Sie eine geringe qualitative Beeinträchtigung in der Bildqualität nicht stört. Verglichen zum PNG reduziert diese kleine Unannehmlichkeit den Datentransfer bis zum Faktor 5.

Benutzen Sie PNG wenn sie eine exakte Wiedergabe der Originaldaten benötigen und Sie den erhöhten Datentransfer in Kauf nehmen können.

Layer ausgewählt wurden, kombiniert der WMS-Server diese und liefert sie als ein Bild an QGIS.

**Tip 23 WMS-LAYERSORTIERUNG**

In der vorliegenden Version von QGIS werden die WMS-Layer in der Reihenfolge dargestellt, wie sie in der Legende eingereiht sind, also von oben nach unten. Wenn Sie Layer in umgekehrter Reihenfolge überlagern wollen, dann können Sie *WMS-Layer hinzufügen* ein zweites Map ausführen, den gleichen Server erneut wählen. Wählen Sie eine weitere Gruppe von Layern, die über die erste Gruppe gelegt werden soll.

**Transparenz**

In der vorliegenden Version von QGIS wurde die Transparenz standardmäßig eingeschaltet, sofern der Server diese unterstützt. Daher existiert keine separate Option im Dialog.

In der Theorie erlaubt dies das Überlagern von WMS-Layern über andere Layer (Raster, Vektor oder andere WMS-Layer), sodass man die darunterliegenden Layer noch durchsehen kann.

---

**Tip 24 WMS-LAYER-TRANSPARENZ**

---

Transparenz hängt von der verwendeten Bildkodierung ab: PNG und GIF unterstützen Transparenz, JPEG hingegen nicht.

---

### **Koordinatensystem**

Koordinatensystem ist die Bezeichnung des OGC für Projektionen.

Jeder WMS-Layer kann in mehreren Koordinatensystemen dargestellt werden; ausschlaggebend sind die angebotenen System des WMS-Servers. Sie haben evtl. bemerkt, dass sich das x in der Spaltenüberschrift basierend auf dem ausgewählten Layer, verändert.

Um ein Koordinatensystem auszuwählen, klicken Sie auf den Knopf *Verändere....* Es öffnet sich ein Fenster ähnlich dem in Abb. 14 im Abschnitt 7.2. Der einzige Unterschied ist der, dass dort nur die Koordinatensysteme angezeigt werden, die auch vom WMS-Server angeboten werden.

---

**Tip 25 WMS-PROJEKTIONEN**

---

Um die besten Resultate zu erreichen, fügen Sie Ihrem Projekt als erstes die WMS-Layer hinzu. Dadurch stellt QGIS die Projekt-Projektion auf die des WMS-Servers ein. Direkte Projektion (On-the-fly-Projektion) kann nun benutzt werden, um andere Vektorlayer auf die WMS-Layer zu projizieren. (vgl. auch Abschnitt 7.2.1). Falls Sie einen WMS-Layer später in einem unterschiedlichen Koordinatensystem hinzufügen, können unvorhersehbare Ereignisse auftreten.

---

### **6.2.4 Das Abfragewerkzeug**

Nachdem Sie einen Layer von einem WMS-Server integriert haben, können Sie diese Layer mit dem *Abfragewerkzeug* abfragen, sofern der Server diese Funktion unterstützt. Ein Klick auf einen Pixel stellt dann eine Abfrage an den WMS-Server für diesen Pixel.

Das Ergebnis wird in Textform geliefert. Die Formatierung hängt von dem jeweilig verwendeten WMS-Server ab.

### **6.2.5 Eigenschaften**

Nach erfolgreichem Hinzufügen eines WMS-Layers können Sie die Eigenschaften des WMS-Server mit einem Rechtsklick auf den Layer in der Legendendarstellung das Kontextmenü aufrufen. Der Knopf *Eigenschaften* öffnet ein Dialogfenster.

## Metadaten-Reiter

Der Metadatenreiter im Kontextmenü zeigt eine Vielzahl von Informationen über den WMS-Server. Diese Infos sind dem Capabilities-Dokument des Servers entnommen.

Viele Definitionen können dem offiziellen Standarddokumenten (5), (6) entnommen werden; hier nur eine kleine nützliche Auswahl:

- **Servereigenschaften**

- **WMS Version** - Die WMS-Version, die vom Server unterstützt wird.
- **Bildformate** - Eine Liste der MIME-Typen mit denen der Server antworten kann. QGIS unterstützt jedes Format, welches die darunterliegende Bibliothek QT unterstützt, mindestens aber `image/png` und `image/jpeg`.
- **Abfrageformate** - Eine Liste der MIME-Typen mit denen der Server auf Pixel-Abfragen antworten kann. Derzeit wird von QGIS nur der Typ `text-plain` unterstützt.

- **Layereigenschaften**

- **ausgewählt** - Gibt an, ob dieser Layer während des Hinzufügens des Server ausgewählt war.
- **sichtbar** - Gibt an, ob der Layer in der Legende angezeigt wird oder nicht. (noch nicht verwendet in der aktuellen Version von QGIS).
- **Kann abfragen** - Gibt an, ob der Layer auf Abfragen Ergebnisse zurückgibt.
- **Kann Transparenz** - Gibt an, ob der Layer transparent gezeichnet werden kann. Diese Version verwendet ein hardcodiertes *Ja*, sofern die Bildkodierung Transparenz bietet.
- **Kann reingezoomt werden** - Gibt an, ob diese Layer gezoomt werden kann. Diese Version von QGIS verwendet standardmäßig *Ja*. Daher kann es sein, dass einige Layer komisch aussehen, die diese Funktion nicht unterstützen.
- **Kaskadierend** - WMS-Server können als Proxy zwischen anderen WMS-Servern agieren, um Rasterdaten für einen Layer anzufordern. Dieser Eintrag gibt an, wieviele WMS-Server angefragt werden müssen, um die Daten zu bekommen.
- **Fixierte Höhe** - Gibt an, ob der Layer eine feste Pixeldimension hat. Diese Version von QGIS nimmt an, dass alle WMS-Layer diesen Wert nicht gesetzt haben. Daher kann es sein, dass einige Layer komisch aussehen, die diese Funktion nicht unterstützen.
- **WGS 84 Boundingbox** - Gibt die Boundingbox eines Layers in WGS84-Koordinaten an. Einige WMS-Server setzen diese Werte nicht korrekt (z.B. stehen darin manchmal UTM-Koordinaten), sodass bei solchen Layern in QGIS der Eindruck entsteht, sehr weit herausgezoomt zu sein. Der Webmaster des WMS-Servers sollte dann auf dieses Problem aufmerksam gemacht werden. Das WMS XML-Element ist `LatLonBoundingBox`, `EX_GeographicBoundingBox` oder die `CRS:84 BoundingBox`.
- **verfügbare Koordinatensysteme** - Die Projektionen, in denen dieser Layer dargestellt werden kann. Diese sind dem Capabilities-Dokument des Servers entnommen.
- **verfügbare Stile** - Die Bildstile, in denen dieser Layer dargestellt werden kann.

### 6.2.6 Einschränkungen des WMS-Klienten

Nicht alle spezifizierten und möglichen WMS-Klientfunktionalitäten wurden in diese Version von QGIS implementiert. Einige nennenswerte Ausnahmen seien kurz erläutert:

#### Ändern der WMS-Layereigenschaften

Wenn der Layer über den Knopf *WMS-Layer hinzufügen* hinzugefügt wurde, besteht im Nachhinein keine Möglichkeit, diese noch einmal zu ändern.

Als Workaround sollte der Layer komplett gelöscht und mit den gewünschten Einstellungen erneut vom Server geladen werden.

#### WMS-Server, die Authentifizierung benötigen

Derzeit sind nur öffentlich zugängliche WMS-Server nutzbar. Es besteht noch keine Möglichkeit, eine Nutzer-Passwort-Kombination einzugeben.

## 6.3 WFS-Klient

In QGIS verhält sich ein WFS-Layer weitestgehend wie ein anderer Vektorlayer. Sie können Objekte abfragen, auswählen und sich die Attributtabelle anschauen. Das WFS-Plugin unterstützt zur Zeit noch kein Editieren eines WFS-Layers.

Das Hinzufügen eines WFS-Layers ist fast identisch mit dem Vorgehen beim Laden eines WMS-Layers. Der aktuelle Unterschied ist, dass keine Beispielservers vordefiniert sind, daher soll dies in dem folgenden Beispiel vorgeführt werden.

### 6.3.1 Laden eines WFS-Layers

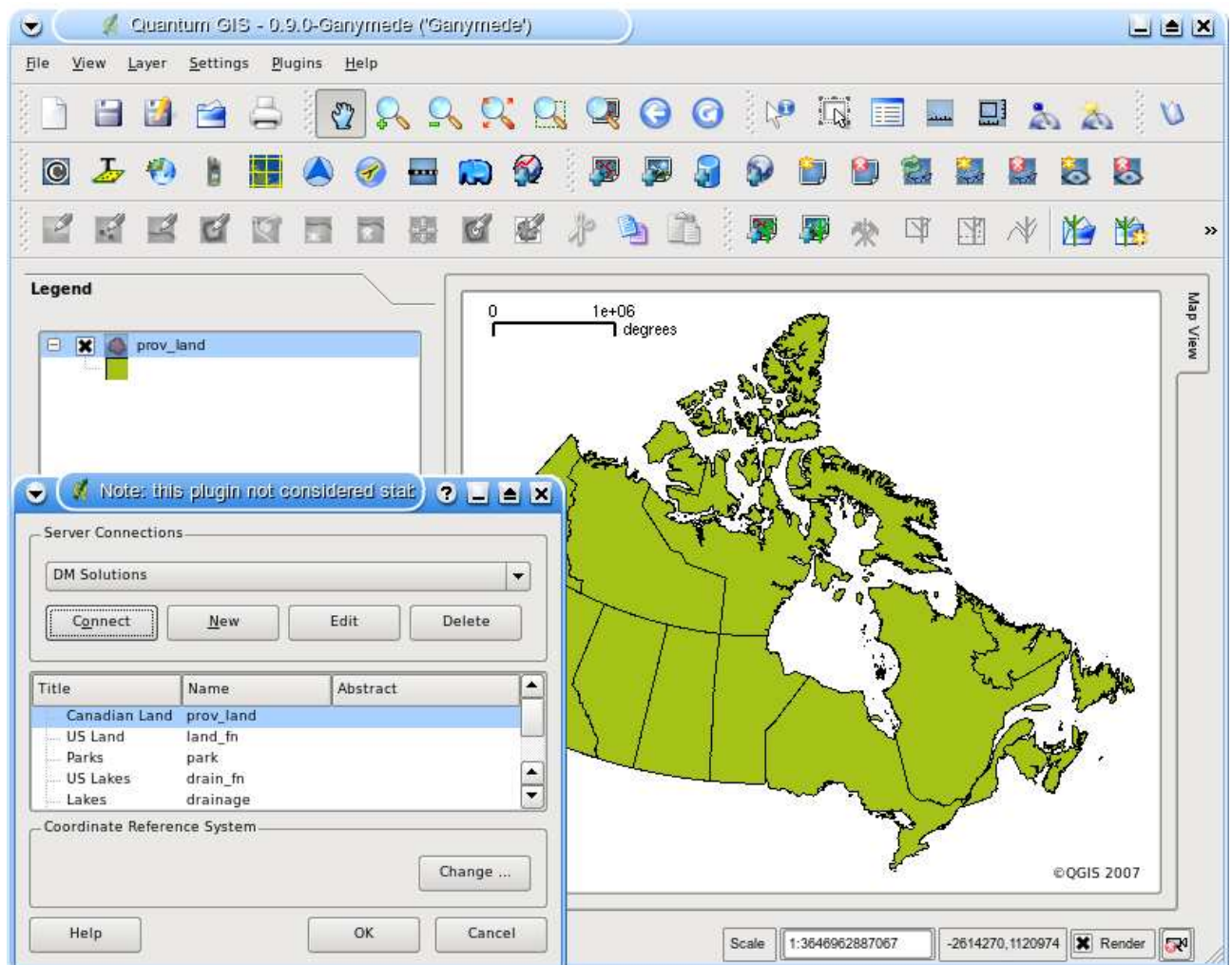
In diesem Beispiel verwenden wir den WFS-Server der Firma DMSolution und laden einen Layer. Die URL ist:

```
http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?VERSION=1.0.0&SERVICE=wfs&REQUEST=GetCapabilities
```

1. Laden Sie das WFS-Plugin, falls dies noch nicht geschehen ist. Öffnen Sie ansonsten den Plugin-Manager und laden Sie es
2. Klicken Sie auf *WFS-Layer hinzufügen* in der Plugin-Werkzeugleiste
3. Klicken Sie auf *Neu*

4. Tragen Sie "DM Solutions" als Name ein
5. Tragen Sie die URL ein (siehe oben)
6. Klicken Sie auf *OK*
7. Wählen Sie "DM Solutions" aus dem Menü der vorhandenen Server
8. Klicken Sie auf *Verbinden*
9. Warten Sie auf die Liste vorhandener Layer des Servers
10. Klicken Sie auf den Layer "Canadian Land"
11. Klicken Sie auf *Hinzufügen* um den Layer anzuzeigen
12. Warten Sie, dass die Daten heruntergeladen und angezeigt werden

Abbildung 13: Einen WFS-Layer hinzufügen



Sie werden sehen, dass der Fortschritt des Herunterladens in der unteren linken Ecke des QGIS-Hauptfensters angezeigt wird. Wenn der layer geladen und angezeigt wird, können Sie die Objekte auswählen und sich die dazugehörige Attributtabellen anzeigen lassen.

Bedenken Sie, dass dieses Plugin noch experimentell ist und Fehler auftreten können genauso wie Abstürze. Die aber hoffentlich in den kommenden Versionen repariert werden.

---

**Tip 26** WMS- UND WFS-SERVER FINDEN

---

Sie können weitere WMS- und WFS-Server mit Hilfe von Google oder ihrer bevorzugten Suchmaschine finden. Es gibt eine Vielzahl von Listen im Internet, die Links zu öffentlichen Servern bereitstellen.

---

---

## 7 Arbeiten mit Projektionen

QGIS unterstützt 'on-the-fly' (OTF) Projektion von Vektorlayern. Diese Funktion erlaubt es, Vektorlayer gemeinsam darzustellen, obwohl sie unterschiedliche Projektionen besitzen.

### 7.1 Überblick zur Projektionsunterstützung

QGIS unterstützt etwa 2700 bekannte Projektionen. Die Projektionen sind in einer SQLite-Datenbank abgelegt, die mit QGIS installiert wird. Normalerweise muss diese Datenbank nicht editiert werden, und es kann Probleme verursachen, wenn Sie es dennoch versuchen. Selbst definierte Projektionen sind in einer Benutzerdatenbank abgelegt. Informationen zum Anlegen einer Benutzerdatenbank finden Sie im Abschnitt 7.3.

Die Projektionen in QGIS basieren auf EPSG Codes und sind weitestgehend der `spatial_references` Tabelle von PostGIS Version 1.x entnommen. Wichtig ist, dass die IDs, welche von QGIS benutzt werden, nicht den IDs der `spatial_references` Tabelle oder den EPSG Codes entsprechen. Die EPSG und PostGIS IDs sind in der SQLite-Datenbank abgelegt und werden benutzt, um Projektionen in QGIS zu spezifizieren.

Um OTF Projektion zu verwenden müssen die Daten Informationen über ihr Koordinatensystem enthalten. Bei PostGIS-Layern benutzt QGIS die spatial reference ID, die bei der Erstellung des Layers festgelegt wurde. Bei Daten, die von der OGR-Bibliothek unterstützt werden, bezieht sich QGIS auf das Vorhandensein von Projektionsinformationen bei den Daten. Bei Shapes bedeutet dies, dass eine Datei mit der Endung `.prj` vorhanden sein muss, in der die Projektion im Well Known Text (WKT) Format angegeben ist. Für ein Shape mit dem Namen `lakes.shp` gäbe es also eine Datei `lakes.prj` mit den entsprechenden Projektionsinformationen.

### 7.2 Getting Started

Zu Beginn ist 'on-the-fly' (OTF) Projektion nicht aktiviert. Um es zu starten, muss in der Menüleiste unter *Einstellungen -> Projekteinstellungen* das Dialogfenster *Projekteigenschaften* geöffnet, im Reiter *Projektion* die Projektion für die zu ladende Datei angeschaltet werden und dann die Checkbox *Projektion zur Laufzeit einschalten* aktiviert werden.

Das Dialogfenster *Projekteigenschaften* kann auch angeschaltet werden, indem man auf das Kamera-Icon in der Statusleiste unten rechts klickt.

---

#### Tip 27 DIALOG PROJEKTEIGENSCHAFTEN

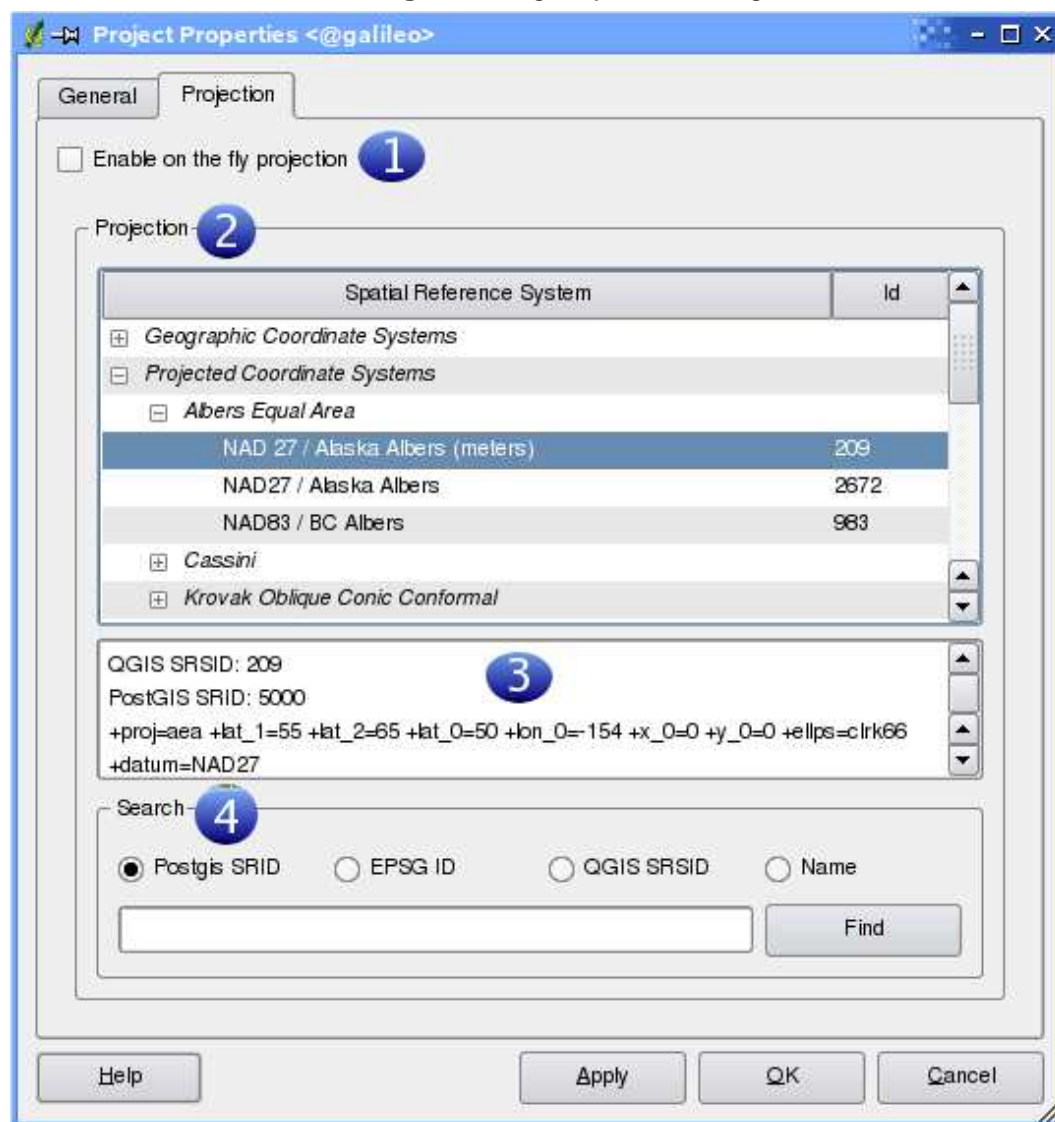
Wenn Sie den Dialog *Projekteigenschaften* über die Menüleiste öffnen, müssen Sie erst auf den Reiter *Projektion* drücken. Über das Kamera-Icon in der Statusleiste unten rechts wird der Reiter *Projektion* direkt in den Vordergrund gerückt.

---



Der Dialog Projekteinstellungen (vgl. Abbildung 14) enthält vier wichtige Optionen, die im Folgenden beschrieben werden.

Abbildung 14: Dialog Projekteinstellungen



1. **Projektion zur Laufzeit einschalten** - über diese Checkbox wird die OTF Projektion aktiviert oder deaktiviert. Wenn sie deaktiviert ist findet keine Projektion statt und jeder Layer wird entsprechend der Koordinaten in der Datei dargestellt. Wenn es aktiviert ist, wird jeder Layer beim Laden in die definierte Projektion 'on-the-fly' umgewandelt dargestellt.
2. **Räumliches Referenzsystem** - dies ist eine Liste mit allen Projektionen, die von QGIS unterstützt werden. Um ein Koordinatensystem zu benutzen, wählt man es aus der Liste aus. Die bis dahin aktive Projektion ist vorgegeben.



3. **Proj4 Text** - dies ist ein Ausdruck, der von der PROJ-Bibliothek genutzt wird. Es dient nur zur Information und kann nicht verändert werden.
4. **Suchen** - wenn Sie die PostGIS, EPSG oder QGIS spatial\_references ID für eine Projektion kennen, können Sie diese benutzen, um ihre Projektion zu finden. Geben Sie die ID ein und klicken Sie auf *Finden*.

### 7.2.1 Eine Projektion spezifizieren

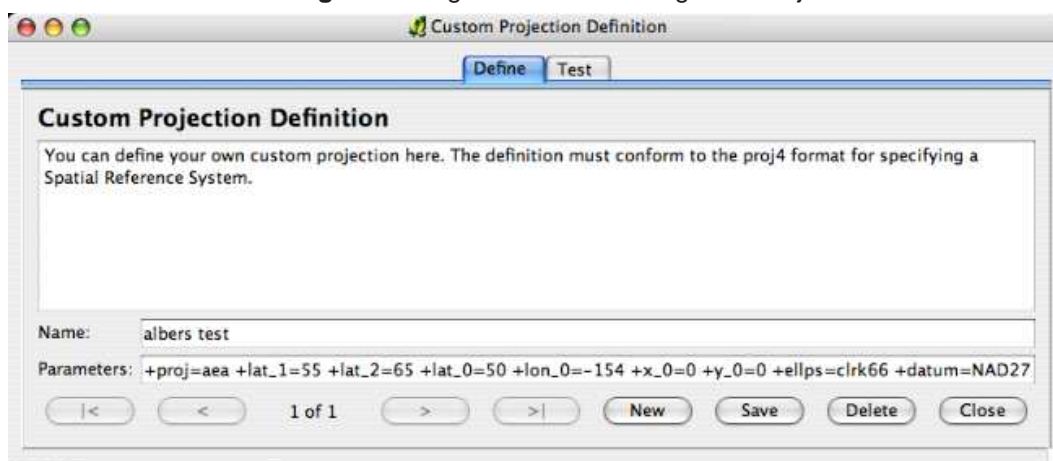
QGIS setzt die Projektion beim Start automatisch auf das Koordinatensystem des zuerst geladenen Layers. Um eine Projektion zu definieren lädt man zuerst einen Layer mit den gewünschten Projektionsinformationen. Dann öffnen Sie in der Menüleiste unter *Einstellungen -> Projekteinstellungen* das Dialogfenster *Projekteigenschaften* und aktivieren im Reiter *Projektion* die Checkbox *Projektion zur Laufzeit einschalten*. Danach schließen Sie den Dialog und laden weitere Layer.

Wenn Sie bereits einige Layer geladen haben und nun OTF Projektion aktivieren möchten, öffnen Sie das Dialogfenster *Projekteigenschaften* und wählen im Reiter *Projektion* die entsprechende Projektion aus der Liste. Wie zuvor beschrieben können Sie dazu auch die Suchfunktion nutzen.

## 7.3 Eigene Projektion definieren

Wenn QGIS nicht die Projektion bereithält, die Sie brauchen, können Sie eine eigene Projektion erstellen. Dazu wählen Sie in der Menüleiste unter *Einstellungen -> Eigene Projektion*. Die eigene Projektion wird in einer Benutzerdatenbank erstellt. Diese enthält ihre räumlichen Lesezeichen und die selbst erstellten Projektionen.

Abbildung 15: Dialog Definition einer eigenen Projektion



In QGIS Version 0.9.1 bedarf es einem grundlegenden Verständnis im Umgang mit der PROJ4-Bibliothek. Zu Beginn sollten Sie einen Blick in das Benutzerhandbuch von PROJ werfen. *Cartographic Projection Procedures for the UNIX Environment - A User's Manual* by Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (zu finden unter <ftp://ftp.remotesensing.org/proj/OF90-284.pdf>). Dieses Handbuch beschreibt die Anwendung des Programms. Die dort beschriebenen kartographischen Parameter sind identisch mit den in QGIS benutzten.

Der Dialog *Definition einer eigenen Projektion* braucht nur zwei Parameter, um eine eigene Projektion zu definieren:

1. Einen beschreibenden Namen
2. Die kartographischen Parameter

Um eine neue Projektion zu erstellen, geben Sie einen beschreibenden Namen ein und die kartographischen Parameter. Abbildung 15 zeigt den Dialog mit einer Beispielprojektion.

Sie können die neue Projektion testen, um zu sehen, ob bei einer Konvertierung von bekannten WGS84 Lat-Lon Koordinaten in ihre Projektion ein sinnvolles Ergebnis herauskommt. Klicken Sie dazu auf den Reiter *Testen*, kopieren ihre Parameter in das Feld *Parameter*, geben ein paar WGS84 Lat-Lon Koordinaten ein und klicken auf den Knopf *Berechnen*.

---

## 8 GRASS Integration

Das GRASS (3) Plugin ermöglicht es, auf GRASS Datenbanken von QGIS aus zuzugreifen. Dies umfasst das Visualisieren, Editieren und Datenerstellen sowie die Analyse von GRASS Daten mit den in GRASS vorhandenen Modulen.

In diesem Kapitel bekommen Sie eine kurze Einführung in das Plugin und dessen Verwendung. Folgende Funktionen stehen über das GRASS Plugin zur Verfügung:

-  GRASS Vektorlayer hinzufügen
-  GRASS Rasterlayer hinzufügen
-  GRASS Werkzeugkiste öffnen
-  Aktuelle GRASS Region bearbeiten
-  GRASS Vektorlayer bearbeiten
-  Mapset öffnen
-  Neues Mapset
-  Neuen GRASS Vektorlayer anlegen
-  Schliesse Mapset

### 8.1 QGIS mit GRASS-Unterstützung starten

Um GRASS-Funktionen in QGIS zu benutzen muss das GRASS-Plugin im Plugin-Manager geladen werden (vgl. Kapitel 11.1.2), genau wie alle anderen Plugins auch. Nach dem Laden erscheint eine neue Werkzeugleiste.<sup>3</sup>

Nun können Sie direkt Daten aus einer existierenden GRASS-Datenbank laden, indem Sie die entsprechenden Icons 'GRASS Vektorlayer hinzufügen' und 'GRASS Rasterlayer hinzufügen' auswählen (siehe Kapitel 8.2), oder Sie können eine neue GRASS Location erstellen (siehe Kapitel 8.3).

---

<sup>3</sup>Das GRASS-Plugin ist einzigartig, indem es seine eigene Werkzeugleiste eigenständig erstellt

## 8.2 GRASS-Daten laden

Wenn das GRASS-Plugin geladen ist, können Sie Vektor- und Rasterdaten mit den entsprechenden Knöpfen in der Werkzeugleiste laden. Als Beispiel benutzen wir den `spearfish` Beispiel-Datensatz in UTM Projektion (siehe Kapitel 3.2).

1. Laden Sie die Datei `spearfish_grass60data-0.3.zip` herunter
2. Erstellen Sie einen neuen Ordner `grassdata` und entpacken Sie die Datei `spearfish_grass60data-0.3.zip` in diesen Ordner.
3. Starten Sie QGIS
4. In der GRASS-Werkzeugleiste klicken Sie auf das Icon `Mapset öffnen`, um den Wählen Sie ein GRASS Mapset Dialog zu öffnen.
5. Für `Gisdbase` geben Sie den Pfad zu dem gerade erstellten Ordner `grassdata` ein.
6. Sie können nun die Location `spearfish60` und das Mapset `PERMANENT` oder `user1` auswählen.
7. Klicken Sie `OK`. Einige Werkzeuge des GRASS-Plugins, die vorher grau hinterlegt waren, sind nun auch aktiv und können benutzt werden.
8. Klicken Sie auf `GRASS Rasterlayer hinzufügen`, wählen die Karte `geology` und klicken auf `OK`. Die Karte `geology` wird nun angezeigt
9. Klicken Sie auf das Icon `GRASS Vektorlayer hinzufügen`, wählen die Karte `roads` und klicken wieder auf `OK`. Nun wird die Karte `roads` über der Karte `geology` angezeigt.

Wie man sehen kann, ist es sehr einfach sich Raster- und Vektordaten aus einer existierenden GRASS-Datenbank in QGIS anzeigen zu lassen. Lesen Sie die folgenden Kapitel, um zu lernen, wie man GRASS Daten editiert und eine neue Location erstellt.

---

### Tip 28 LADEN VON GRASS-DATEN

Sollten Sie wider Erwarten Probleme beim Laden von GRASS-Daten haben (z.B. beendet sich QGIS unvorhergesehen), dann überprüfen Sie bitte, ob das GRASS-Plugin korrekt geladen ist (siehe Abschnitt 8.1).

---

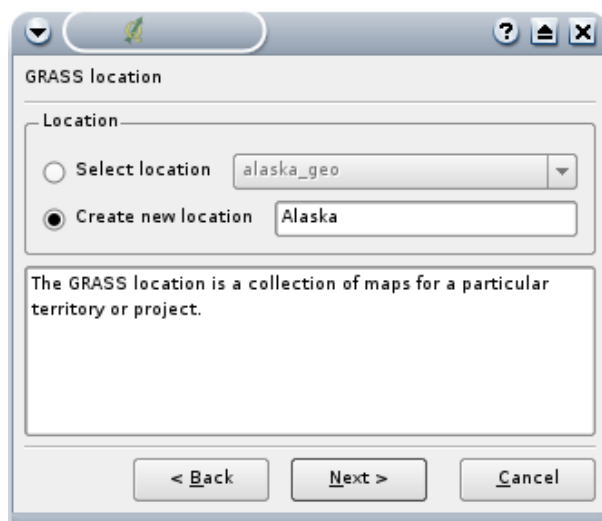
## 8.3 Erstellen einer Location

GRASS speichert Daten in einer sog. "Location", die ein durch den Benutzer festgelegtes Gebiet mit einer definierten Projektion darstellt. Damit Daten mit dem GRASS-Plugin bearbeitet werden können, müssen diese in eine `location` importiert werden.<sup>4</sup>

An dieser Stelle sehen Sie ein Beispiel, wie eine GRASS Location mit der Projektion `Albers Equal Area` (Einheit Meter) für den QGIS-Beispieldatensatz (siehe Kapitel 3.2) erstellt werden kann.

---

<sup>4</sup>Dies ist nicht ganz korrekt. Sie können mit dem Modul `v.external` auch Daten anzeigen, die nicht nach GRASS importiert wurden.

**Abbildung 16:** Erstellen einer GRASS Location in QGIS

1. Starten Sie QGIS
2. Laden Sie das GRASS Plugin, falls dies noch nicht geschehen ist
3. Laden Sie das Shape `alaska.shp` (vgl. Kapitel 4.1.1).
4. Klicken Sie auf das Icon Neues Mapset in der GRASS-Werkzeugleiste, um den Mapset Dialog zu starten
5. Alle GRASS locations sind in einem Ordner, der GRASS Datenbank abgelegt (meist `grassdata`). Wählen Sie einen existierenden Ordner mit bereits vorhandenen Locations oder erstellen Sie einen neuen für die zu erstellende Location
6. Klicken Sie auf Next
7. Mit dem Dialog kann eine bereits vorhandene Location durch eine weitere Mapset ergänzt werden, oder eine neue Location mit neuen Mapsets erstellt werden. Klicken Sie auf den Knopf "Erstelle neue Location"
8. Geben Sie einen Namen für die neue Location ein - wir nehmen Alaska
9. Klicken Sie auf Next
10. Definieren Sie die Projektion in dem Sie auf den Knopf "Projektion" klicken, um eine Liste auswählbarer Projektion angezeigt zu bekommen.
11. Wir benutzen die Albers Equal Area Alaska (meters) Projektion. Da wir wissen, dass es sich hierbei um den PostGIS SRID 5000 handelt, geben wir die Zahl in das Suchfenster ein. (Wenn Sie dies für einen anderen Layer wiederholen wollen, aber nicht den SRID kennen, klicken Sie einfach auf das Kamera Icon unten rechts in der Statusleiste (vgl. Kapitel 7.2).)
12. Klicken Sie auf Suchen, um die Projektion auszuwählen
13. Klicken Sie auf Next

14. Um die Standardausdehnung der Location zu definieren, müssen wir die Eckkoordinaten der Umrandung angeben. Da wir einen Layer geladen haben, klicken wir einfach auf den Knopf *Setze aktuelle QGIS Ausdehnung*, um die Ausdehnung der Karte als Standardausdehnung für die Location zu verwenden.
15. Klicken Sie auf *Next*
16. Wir müssen nun noch eine Mapset für die Location definieren. Der Name ist egal, sie können z.B. ihren Benutzernamen wählen.
17. Prüfen Sie alles, damit alles wie gewünscht ist
18. Klicken Sie auf *Finish*
19. Die Mapset und Location sind nun erstellt worden und sind als aktuelle Arbeitsumgebung definiert
20. Beachten Sie, dass einige Werkzeuge des GRASS Plugins grau hinterlegt waren und nun auch zur Verfügung stehen.

Auch wenn es recht kompliziert aussieht, ist dies doch ein sehr einfacher Weg, eine neue GRASS Location zu erstellen. Unsere Location "Alaska" ist nun bereit für den Import von Daten. Um die Standardausdehnung zu sehen, zoomen Sie ein Stück heraus und klicken dann auf das Icon *Aktuelle GRASS Region* darstellen. Die Umrandung der Region kann an- und ausgeschaltet werden.

## 8.4 Vektordatenmodell

Für das Arbeiten mit GRASS-Vektordaten ist es wichtig, das Datenmodell von GRASS-Vektoren zu kennen. GRASS nutzt ein topologisches Datenmodell. Das bedeutet, dass Flächen nicht als geschlossene Polygone vorhanden sind, sondern als ein oder mehrere Umrandungen (Boundaries). Eine Umrandung (Boundary) zwischen zwei aneinander grenzenden Flächen ist nur einmal digitalisiert worden; beide Flächen teilen sich diese Umrandung. Umrandungen dürfen keine Lücken haben. Eine Fläche besteht also aus einer Umrandung und einem Zentroid, der diese Fläche als ein sog. Labelpunkt mit einer Attributtabelle verknüpft.

Neben den Umrandungen und Zentroiden kann eine Vektorkarte selbstverständlich auch Punkte und Linien enthalten. Alle diese Geometrieelemente können innerhalb ein und dem selben Datensatz enthalten sein. Sie werden in unterschiedlichen 'Layern' innerhalb von QGIS dargestellt.

Es ist möglich, mehrere 'Layer' in einem Vektordatensatz zu speichern. Beispielsweise können Felder, Wälder und Seen in einem Vektordatensatz gespeichert werden. Angrenzende Seen, Felder und Wälder teilen sich dann die gleiche Umrandung, jedoch haben sie separate Attributtabellen. Darüber hinaus können Sie auch Attribute für die Umrandungen vergeben, falls eine Umrandung gleichzeitig einen Weg darstellt. In diesem Fall könnte auch die Umrandung eine separate Attributtabelle haben.

Der 'Layer' eines jeden Objektes wird in GRASS als 'layer' bezeichnet. 'Layer' bezeichnet eine Nummer, wenn mehr als ein Layer im Datensatz enthalten ist; beim obigen Beispiel wären es 3 Layer

(Felder, Wälder und Seen). Derzeit werden nur Nummern als Layername unterstützt, in kommenden GRASS-Versionen werden auch Namen möglich sein.

Attribute können in externen Datenbanktabellen abgelegt werden, beispielsweise DBF, PostgreSQL, MySQL, SQLITE3, etc.

Die Attribute in den Tabellen werden über ein sog. 'Kategoriefeld' an die Geometrien des Datensatzes gehängt. Die 'Kategorie' (oder key, ID, etc) ist eine Ganzzahl, über die eine Verknüpfung zwischen den Geometrien und den Spalten in der Datenbanktabelle hergestellt wird.

---

**Tip 29** DAS GRASS-VEKTORDATENMODELL LERNEN

---

Der beste Weg, etwas über das GRASS Vektordatenmodell zu erfahren, ist das Studieren eines der vielen verfügbaren GRASS-Tutorien, wo dieses Thema vertieft behandelt wird. Unter <http://grass.itc.it/gdp/manuals.php> sind eine Fülle von weiteren Informationsquellen in unterschiedlichen Sprachen vorhanden.

---

## 8.5 Digitalisier- und Bearbeitungswerkzeuge

Die Digitalisierwerkzeuge für GRASS-Vektorlayer werden über den Knopf *Bearbeiten von GRASS-Vektorlayern* in der GRASS-Werkzeugleiste gestartet. Stellen Sie sicher, dass der zu bearbeitende Layer geladen und ausgewählt ist, bevor Sie auf den Editier-Knopf drücken. Falls Sie einen neuen Vektordatensatz erzeugen wollen, benutzen Sie den Menüeintrag unter *Plugins->GRASS->Neuen Vektordatensatz erzeugen*.

Abbildung 17 zeigt den Editierdialog, der erscheint, wenn Sie den entsprechenden Knopf drücken.

Die Funktionalität und Einstellungsmöglichkeiten werden im folgenden Abschnitt beschrieben.

### 8.5.1 Werkzeugleiste

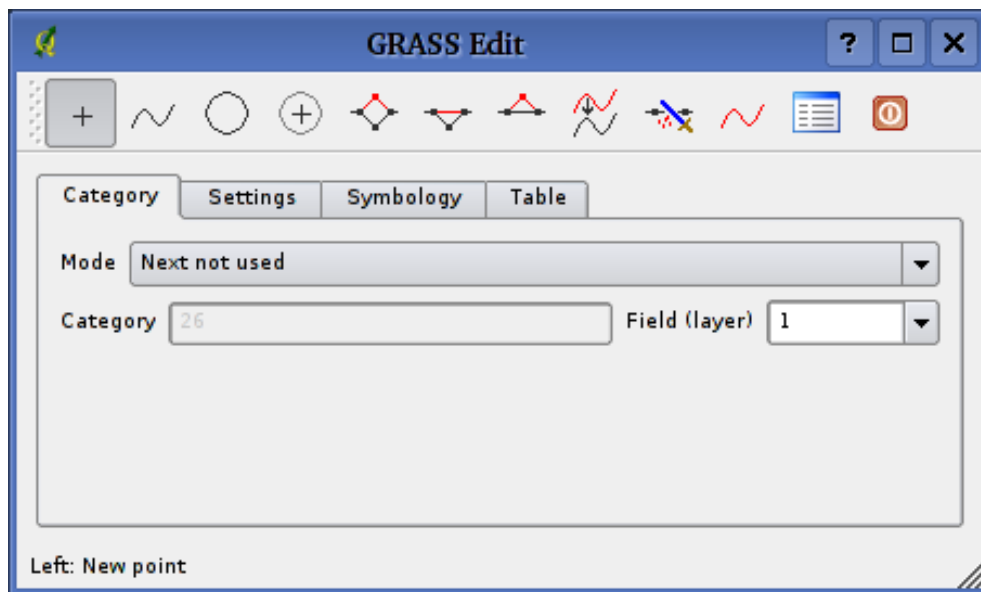
Tabelle 4 listet die Digitalisierwerkzeuge des GRASS-Plugins auf. Sie sind in der Werkzeugleiste des Editierdialogs implementiert.

### 8.5.2 Kategorie-Reiter

In diesem Reiter können Sie einstellen, in welcher Weise Kategoriewerte neuen Objekten oder neue Kategoriewerte vorhandenen Objekten zugewiesen werden sollen.

- Modus: Welcher Kategoriewert soll der Geometrie zugewiesen werden
  - Nächste nicht benutzte - der nächste Kategoriewert, der noch nicht im Vektordatensatz benutzt wird.

Abbildung 17: GRASS Editerdialog



- Manueller Eintrag - Manuellen Eintrag der Kategoriewerte im 'Kategorie'-Eingabefeld vornehmen.
- Keine Kategorie - digitalisiere Geometrie ohne Angabe eines Kategoriewertes
- Kategoriewert - eine Nummer (ID), die dem digitalisierten Objekt zugewiesen wird
- layer - Objektidentifikation (der Attributtabelle)

---

**Tip 30** ERZEUGEN VON WEITEREN 'LAYERN' MIT QGIS

Wenn Sie weitere 'Layer' Ihrem Datensatz hinzufügen wollen, so können Sie einfach eine neue Zahl in das 'Layer'-Feld eintippen und im Anschluss Return drücken. Im Reiter 'Tabelle' können Sie nun eine neue Attributtabelle erstellen und mit diesem neuen 'Layer' verbinden.













---

### 8.5.3 Einstellungsreiter

Dieser Reiter erlaubt Ihnen das Setzen des Suchradius in Bildschirmpixeln. Dies ist der Schwellenwert in Pixeln, innerhalb dessen neue Punkte oder Linien an vorhandene Knoten gesnappt werden. Dies hilft, Spalten oder Überschüsse zwischen Umrandungen zu vermeiden. Der Standardwert ist auf 10 Pixel eingestellt.



**Tabelle 4:** GRASS Editierwerkzeuge

Icon	Werkzeug	Zweck
	Neuer Punkt	Digitalisiert neuen Punkt
	Neue Linie	Digitalisiert neue Linie (zum Beenden ein anderes Werkzeug wählen)
	Neue Umrandung	Digitalisiert eine neue Umrandung (Boundary) (zum Beenden ein anderes Werkzeug wählen)
	Neuer Zentroid	Digitalisiert einen neuen Zentroid (Labelpunkt für eine existierende Fläche)
	Verschiebe Stützpunkt	Wählt einen Stützpunkt einer existierenden Linie oder Umrandung und setzt diesen an eine neue Position
	Füge Stützpunkt hinzu	Fügt einen Stützpunkt zu einer existierenden Linie hinzu
	Löscht Stützpunkt	Löscht einen Stützpunkt von einer existierenden Linie (Bestätigung der Auswahl durch einen weiteren Klick nötig.)
	Verschiebe Linie	Wählt existierende Linie und verschiebt sie an eine neue Position
	teile Linie	Teilt eine existierende Linie in zwei Teile
	Lösche Linie	Löscht eine existierende Linie (Bestätigung der Auswahl durch einen weiteren Klick nötig.)
	Editiere Attribute	Editiert Attribute eines existierenden Objekts (Beachten Sie, dass ein Objekt mehrere Feature repräsentieren kann, siehe oben)
	Beenden	Beendet die Bearbeitung (und aktualisiert die Topologie anschließend).

### 8.5.4 Symbologiereiter

Dieser Reiter erlaubt die Farbeinstellungen für die verschiedenen Geometrietypen und ihren Topologiestatus (z.B. offene/geschlossene Fläche).

### 8.5.5 Tabellenreiter

Dieser Reiter gibt Informationen über die Attributtabelle des aktuellen 'Layers'. In diesem Reiter können Sie die Datenbankverbindung des aktuellen 'Layers' hinzufügen, modifizieren oder eine neue Tabelle erstellen.

---

#### Tip 31 GRASS SCHREIBBERECHTIGUNG

---

Um einen Datensatz zu editieren, müssen Sie der Besitzer des aktuellen GRASS-Mapsets sein. Es ist nicht möglich, Karten aus einem Mapset zu editieren, das Ihnen nicht 'gehört', auch wenn Sie auf Dateisystemebene die Schreibrechte haben.

---

## 8.6 Regionswerkzeug



Die aktuelle Region (Projektausdehnung) ist in GRASS sehr wichtig für alle Rastermodule. Alle neu erzeugten Rasterdaten haben nämlich die Ausdehnung und Auflösung der aktuellen Region, egal wie groß die Standard-Region definiert ist. Die Region ist in der Datei \$LOCATION/\$MAPSET/WIND gespeichert, und definiert Nord-, Süd-, Ost- und Westausdehnung sowie Anzahl der Spalten, Anzahl der Reihen und die horizontale und vertikale räumliche Pixelauflösung.

Es ist auch möglich, die Region mit einem Schalter aus und ein zu schalten. Benutzen Sie dazu den Knopf **Zeige aktuelle GRASS-Region**

Um die aktuelle GRASS-Region zu ändern, benutzen Sie den Knopf **Aktuelle GRASS-Region editieren**. Dort haben Sie die Möglichkeit, die Region zu verändern sowie das Aussehen der Regionsdarstellung. Wenn der Dialog sichtbar ist, können Sie die Region neben der einfachen Koordinateneingabe auch interaktiv mit der Maus im Kartenbild verändern. Die Koordinaten werden dann automatisch angepasst, sobald Sie den Dialog schließen.

## 8.7 GRASS Werkzeugkasten



Der GRASS Werkzeugkasten stellt QGIS analytische Funktionen direkt aus GRASS bereit. Um den Werkzeugkasten zu nutzen, müssen Sie ein beschreibbares Mapset geöffnet haben. Andernfalls ist der Knopf ausgegraut. Dies ist nötig, da der Werkzeugkasten meistens weitere Datensätze erzeugt, die folglich in ein beschreibbares Mapset abgelegt werden müssen.

Falls Sie QGIS aus einer laufenden GRASS-Sitzung heraus gestartet haben, wird das aktuelle Mapset genutzt, da Sie ja dafür Schreibberechtigung besitzen.

Eine weitere Option besteht darin, dass Sie ein schreibbares Mapset über das Menu **Plugins->GRASS->Öffne Mapset** auswählen.

Falls Sie einen ausgegrauten GRASS Werkzeugkasten in der Menüleiste vorfinden, stellen Sie sicher, dass Sie ein beschreibbares Mapset geöffnet haben bzw. öffnen Sie ein solches.

Darüber hinaus stellt Ihnen der Werkzeugkasten auch einen Datenbrowser für Ihre GRASS-Daten in der aktuellen Location zur Verfügung.

### 8.7.1 Die Module im Werkzeugkasten

Der Werkzeugkasten bringt bereits eine kleine Auswahl aus GRASS-Modulen mit, die innerhalb von QGIS benutzt werden können. Sie sind thematisch in Blöcken organisiert. Diese Themengruppen können selber verändert, angepasst und gelöscht werden. Vgl. dazu Abschnitt 8.7.3.

Wenn Sie auf ein Modul klicken, öffnet sich ein weiterer Reiter in der Werkzeugleiste. Dieser Modulreiter hat weitere drei Unterreiter:

1. Optionen
2. Ergebnis
3. Handbuch

#### Optionen

Dieser Reiter stellt Ihnen in vereinfachter Form die unbedingt nötigen Eingabeparameter zur Verfügung, die das Modul zum Laufen benötigt. Bitte beachten Sie, dass diese Eingabefelder extra so einfach wie möglich gehalten wurden, um die Struktur klar zu halten. Falls Sie alle vom Modul unterstützten Parameter benötigen, nutzen Sie die GRASS-Eingabeshell, in der Sie das Modul ebenfalls aufrufen können.

#### Ergebnis

Dieser Reiter stellt die Ausgabe des Moduls während des Laufens dar. Nachdem Sie den Knopf 'Los' gedrückt haben, wird auf diesen Reiter gewechselt und Sie sehen die Statusausgaben des Moduls. Wenn alles funktioniert hat, sehen Sie den Ausgabetext *Successfully finished*.

#### Handbuch

Dieser Reiter enthält die Hilfeseite zu jedem GRASS-Modul. Falls Sie weitergehende Informationen zum gewählten Modul benötigten, schauen Sie in diese Hilfeseite. Dabei werden Sie feststellen, dass viele Module mehr Parameter haben, als im Reiter *Optionen* zu sehen sind. Das ist korrekt so, und auch so gewollt. Um die GUI möglichst einfach zu halten, sind nur die nötigsten Optionen in den Reiter *Optionen* übernommen worden. Natürlich stehen Ihnen alle Optionen des Moduls direkt in der GRASS-Eingabeshell zur Verfügung.

**Tip 32** ERGEBNISS DIREKT ANZEIGEN LASSEN







Wollen Sie Ihre Ergebnisse direkt in der Kartenansicht ansehen, nutzen Sie den Knopf *Ergebnis ansehen* am Ende des Modulreiters.

**8.7.2 GRASS Datei-Browser**

Eine schöne Funktion ist der GRASS Datei-Browser. In der linken Hälfte des Datei-Browsers können Sie sich alle in der Location enthaltenen Mapsets anschauen.

In der rechten Hälfte des Fensters sehen Sie weitere Metainformationen zum ausgewählten Datensatz wie beispielsweise Auflösung, BoundingBox, Datenquelle, bei Vektordaten die Attributtabelle, ...

Die Werkzeugleiste im Browser stellt Ihnen die folgenden Werkzeuge für den ausgewählten Datensatz zur Verfügung:

-  Füge Datensatz zur Karte hinzu
-  Kopiere ausgewählte Karte
-  Benenne ausgewählte Karte um
-  Lösche ausgewählte Karte
-  Setze die Region auf die Ausdehnung der ausgewählten Karte
-  Erneure die Ansicht des Browsers

Der 'Umbenennen' und 'Löschen'-Knopf steht Ihnen nur in Ihrem aktuellen Mapset zur Verfügung. Die anderen Werkzeuge funktionieren auch in den anderen Mapsets.

**8.7.3 Anpassen der Module**

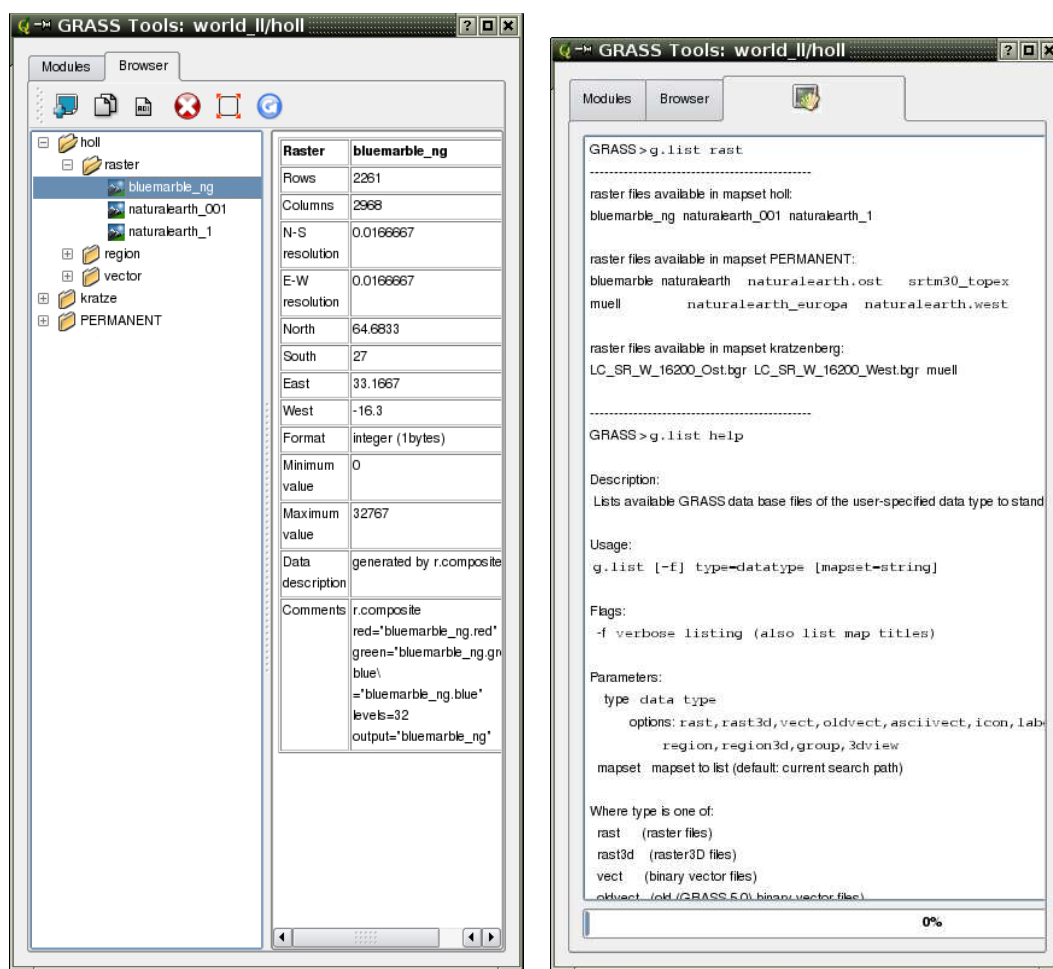
Nahezu alle GRASS-Module können in den GRASS-Werkzeugkasten integriert werden. Eine XML-Schnittstelle wertet die sehr einfachen XML-Dateien, die die Module beschreiben, aus und übernimmt die Oberflächendarstellung.

Eine kurze Einführung zur Integration weiterer Module kann im QGIS-WIKI unter der folgenden URL gefunden werden: [http://wiki.qgis.org/qgiswiki/Adding\\_New\\_Tools\\_to\\_the\\_GRASS\\_Toolbox](http://wiki.qgis.org/qgiswiki/Adding_New_Tools_to_the_GRASS_Toolbox).

Beispielhaft ist hier die XML-Datei zum Modul `v.buffer` (`v.buffer.qgm`) dargestellt:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Abbildung 18: GRASS Werkzeugkasten



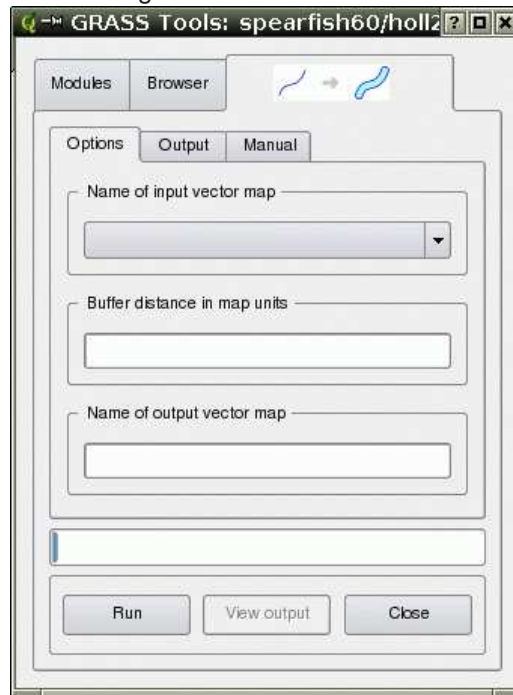
(a) GRASS Datei-Browser im Werkzeugkasten

(b) GRASS-Eingabeshell im Werkzeugkasten

```
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">
```

```
<qgisgrassmodule label="Vector buffer" module="v.buffer">
  <option key="input" typeoption="type" layeroption="layer" />
  <option key="buffer"/>
  <option key="output" />
</qgisgrassmodule>
```

Der XML-Auswerter liest diese Definition und erstellt einen neuen Reiter im Werkzeugkasten, wenn Sie das Modul auswählen.

**Abbildung 19:** Das neu integrierte Modul nach der Auswertung der XML-Datei

## 8.8 Neue GRASS-Vektoren erzeugen

Mit dieser Version von QGIS können Sie sehr einfach neue Vektordatensätze mit GRASS erzeugen.

Wählen Sie im Menü **Plugins->GRASS->Neuen Layer erstellen** aus, geben Sie einen Namen für den Datensatz ein und starten Sie mit der Digitalisierung. Falls der gewünschte Menüeintrag ausgegraut ist, überprüfen Sie, ob Sie in dem ausgewählten Mapset schreiben dürfen. Falls Sie vergessen haben, wie das geht, schauen Sie in Abschnitt 8.2 nach.

Da GRASS alle Geometrietypen in einem Datensatz verwalten kann, ist es nicht nötig, einen Geometrietyp zu wählen. Dies ist nur bei Shapedateien nötig (vgl. Abschn. 4.4.3).

Einige Hinweise zum Digitalisieren:

- Stellen Sie sicher, dass eine Attributtabelle mit den benötigten Attributspalten enthalten ist, bevor Sie mit der Digitalisierung starten. Wählen Sie dazu den 'Tabellen'-Reiter im Digitalisier-dialog.
- Wenn Sie einen Polygondatensatz erzeugen wollen, erwägen Sie evtl., den Modus auf `No category` zu stellen, denn die Umrandungen der Flächen benötigen in der Regel keinen Eintrag in der Attributtabelle. Wenn Sie alle Umrandungen digitalisiert haben, stellen Sie den Modus wieder auf `Next not used` und starten Sie mit der Digitalisierung der Zentroiden, über die die Informationen zu jeder Fläche mit der Attributtabelle verknüpft werden.

---

## 9 Erstellen einer Mapserver Map-Datei

QGIS kann benutzt werden, um eine Map-Datei für MapServer zu erstellen. Sie nutzen QGIS um die Karten zu arrangieren, zu beschriften und die Farben nach ihren Wünschen anzupassen.

Damit sie das MapServer Exportmodul benutzen können, muss Python auf Ihrem System installiert und QGIS muss mit Python-Support gebaut worden sein.

### 9.1 Erstellen einer QGIS Projektdatei

Um eine MapServer Map-Datei zu erstellen:

1. Laden Sie die gewünschten Layer in QGIS
2. Beschriften Sie die Layer und definieren Sie die Farben
3. Arrangieren Sie die Layer in der Reihenfolge, in der Sie auch im MapServer erscheinen sollen
4. Speichern Sie alles in einer QGIS Projektdatei

Damit sind wir nun an dem Punkt, an dem wir die Map-Datei erstellen können.

---

#### **Tip 33** MAPSERVER EXPORT BENÖTIGT EINE QGIS PROJEKT-DATEI

Dies ist für viele Anwender verwirrend gewesen. Das MapServer Exportmodul arbeitet auf Basis einer QGIS Projekt-Datei, **nicht** der aktuelle Inhalt des Kartenfensters und der Legende wird benutzt. Wenn Sie das MapServer Exportmodul benutzen, geben Sie eine QGIS Projekt-Datei als Eingabedatei an.

---

### 9.2 Export einer QGIS Projektdatei als Map-Datei

Das Export-Werkzeug (`msexport`) liegt im QGIS Programmordner und kann unabhängig von QGIS benutzt werden.

Von QGIS aus können Sie das Modul `Exportiere als MapServer Map...` über das Menü `Datei` erreichen.

Hier ist eine Zusammenfassung der Eingabefelder:

#### **Kartendatei**

Tragen Sie hier den Namen der Map-Datei ein, die erstellt werden soll. Sie können mit dem 'Speichern unter' Knopf auswählen, wo die Datei gespeichert werden soll.

#### **QGIS Projektdatei**

Geben Sie den kompletten Pfad zur QGIS Projektdatei (`.qgs`) an, die Sie exportieren möchten. Sie können mit dem 'Durchsuchen' Knopf nach der Projektdatei suchen.

**Name**

Ein Name für die Karte. Dieser Name wird als Prefix für alle durch den MapServer erstellten Karten verwendet.

**Breite**

Breite des Ausgabebildes in Pixeln.

**Höhe**

Höhe des Ausgabebildes in Pixeln.

**Einheiten**

Maßeinheiten für die Ausgabe

**Bildtyp**

Bildformat, das vom MapServer erstellt wird

**Vorlage**

Kompletter Pfad zur MapServer Template-Datei, die mit der Map-Datei benutzt wird

**Kopfzeile**

Kompletter Pfad zur MapServer Header-Datei, die mit der Map-Datei benutzt wird

**Fusszeile**

Kompletter Pfad zur MapServer Footer-Datei, die mit der Map-Datei benutzt wird

Nur die QGIS Projektdatei Eingaben sind notwendig, um eine Map-Datei zu erstellen. Es kann aber sein, dass diese nicht auf Anhieb funktioniert, abhängig davon, was sie sich vorstellen. Auch wenn QGIS durchaus geeignet ist, eine Map-Datei zu erstellen, ist es häufig doch notwendig, die Map-Datei manuell nachzubearbeiten. man sollte es daher eher als eine Hilfe betrachten, um die Map-Datei nicht komplett neu erstellen zu müssen.

**Beispiel zum Erstellen einer Map-Datei**

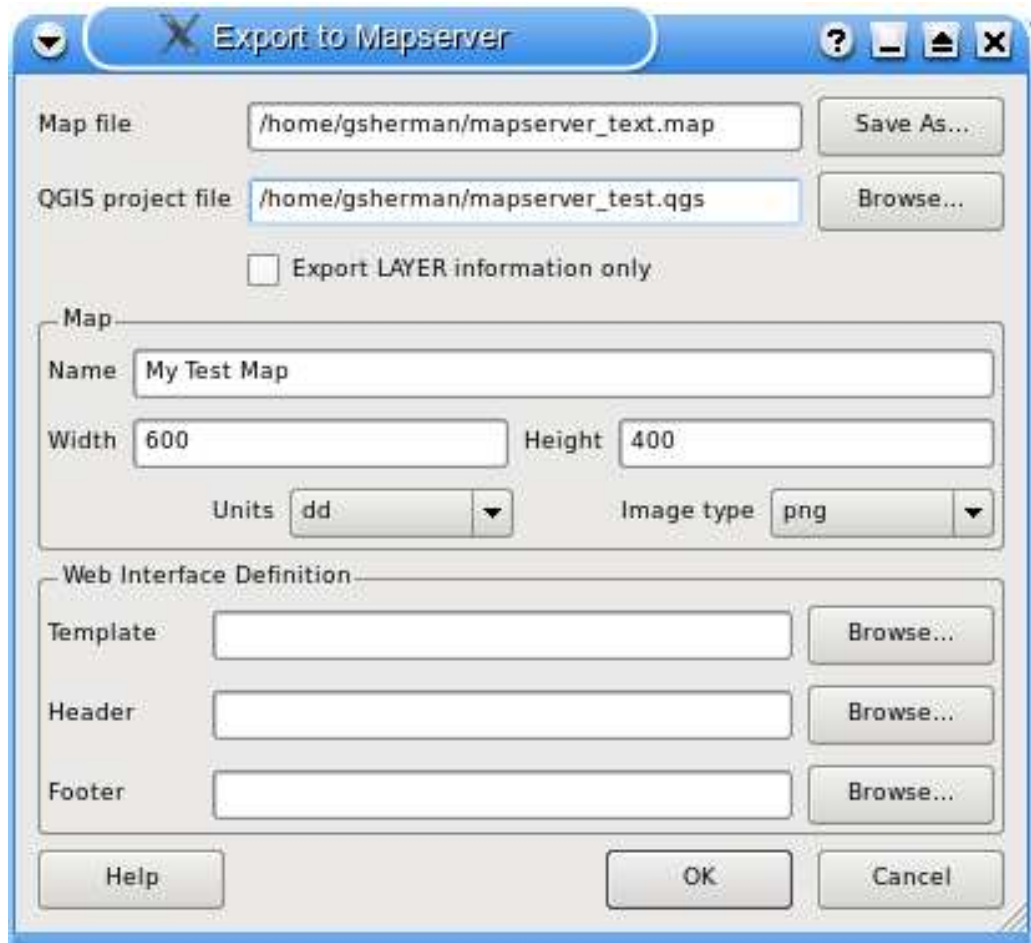
Nun wollen wir das Erstellen einer Map-Datei einmal praktisch durchführen mit den Shapes `alaska`, `lakes` und `ivers` aus den QGIS-Beispieldaten:

1. Laden Sie die Shapes `alaska`, `ivers` und `lakes` in QGIS
2. Verändern Sie die Farben und Beschriftungen nach Ihren Wünschen
3. Speichern Sie alles als Projekt mit dem Modul `Speicher Projekt` in Menü `Datei`
4. Öffnen Sie das Exportmodul, indem Sie auf `Exportiere als MapServer Map...` im Menü `Datei` auswählen
5. Geben Sie einen Namen für die Map-Datei ein
6. Suchen Sie nach der gerade gespeicherten QGIS-Projektdatei
7. Geben Sie einen Namen für die Karte ein
8. Geben Sie die Werte 600 für die Breite und 400 für die Höhe ein



9. Die Layer liegen mit den Einheiten Dezimalgrad vor, daher müssen wir die Maßeinheiten nicht ändern
10. Wählen Sie "png" als Bildtyp
11. Klicken Sie auf OK um die Map-Datei zu erstellen

**Abbildung 20:** Das 'Exportiere als MapServer Map' Module in QGIS



In dieser QGIS-Version wird nicht angezeigt, ob der Export erfolgreich war oder nicht. Dies wird in einer nächsten Version folgen.

Sie können sich die Map-Datei in einem Texteditor anschauen oder einfach das Konsolen-Tool `less` benutzen. Sie werden sehen, dass das Exportmodul alle Metadaten hinzugefügt hat, um die Map-Datei für WMS zu aktivieren.

### 9.3 Testen der Map-Datei

Wir können die Map-Datei testen, indem wir ein Bild mit dem Tool `shp2img` erzeugen. Das `shp2img` Programm ist Teil der MapServer Installation aber es ist auch in dem Programm FWTools vorhanden. Um ein Testbild zu erstellen:

- Öffnen wir eine Shell, damit wir das Programm `shp2img` aus der Kommandozeile starten können.
- Wenn die Map-Datei nicht im aktuellen Verzeichnis gespeichert wurde, wechseln Sie bitte in das entsprechende Verzeichnis mit der Map-Datei
- starten Sie `shp2img`
- Schauen Sie sich das erstellte Bild mit einem Bildbetrachter an

Unter der Annahme, dass die Map-Datei `mapserver_test.map` genannt wurde, lautet das `shp2img` Kommando:

```
shp2img -m mapserver_test.map -o mapserver_test.png
```

Das somit erstellte PNG enthält alle Layer, die wir als QGIS-Projekt abgespeichert haben. Außerdem ist die Ausdehnung des Bildes exakt diejenige, die wir in der QGIS-Projektdati abgespeichert haben.

Wenn Sie vorhaben, die Map-Datei auch für WMS-Anfragen zu nutzen, dann brauchen Sie wahrscheinlich nichts weiter zu machen. Wenn Sie es als Kartenvorlage oder individuelle Oberfläche nutzen wollen, ist wahrscheinlich noch ein bisschen handarbeit notwendig. Um zu sehen, wie einfach es ist, Karten aus QGIS ins Internet zu stellen, schauen Sie sich das 5-minütige Flashvideo von Christopher Schmidt's an.<sup>5</sup>

---

<sup>5</sup><http://openlayers.org/presentations/mappingyourdata/>

---

## 10 Print Composer

Der Print Composer bietet Layout- und Druckfunktionen in QGIS. Es ist möglich, Elemente wie ein Kartenfenster, eine Legende, einen Maßstab, Bilder und Texte zu layouten und dann zu drucken. Sie können die Größe, Position sowie die Eigenschaften jedes Elements anpassen und so ein eigenes Layout erstellen. Das Ergebnis kann dann geplottet, als Bild oder SVG exportiert werden.

Um den Print Composer zu starten, klicken Sie auf den Knopf **Drucken** in der Werkzeugleiste oder wählen Sie über die Menüleiste unter **Datei** die Funktion **Drucken**.

### 10.1 Den Print Composer benutzen

Um den Print Composer zu benutzen, laden Sie zuerst die Layer, die Sie layouten möchten in das Kartenfenster und geben Sie ihnen das für den Druck gewünschte Aussehen.

Wenn Sie den Print Composer öffnen, sehen Sie erstmal nur ein weißes Blatt vor sich, auf dem Sie dann die Elemente Kartenfenster, Legende, Maßstab, Bilder und Texte hinzufügen können. Abbildung 21 zeigt den Anfangsbildschirm des Print Composers, bevor Elemente hinzugefügt werden.

Der Print Composer hat zwei Reiter: **Allgemein** und **Eintrag**. Der **Allgemein** Reiter ermöglicht es, die Papiergröße, -ausrichtung und Auflösung der Druckkarte zu definieren. Der Reiter **Eintrag** gibt die Eigenschaften für den aktuell dargestellten Kartenausschnitt wieder. Wenn Sie ein Element im Fenster selektieren (z.B. Legende, Maßstab, Bild. usw) und dann auf den Reiter **Eintrag** klicken, können Sie dessen Eigenschaften anpassen.

Es können mehrere Elemente im Print Composer angelegt werden, also auch mehr als ein Kartenfenster mit eigener Legende. Jedes Element hat seine eigenen Eigenschaften und im Fall des Kartenfensters eine eigene Ausdehnung.

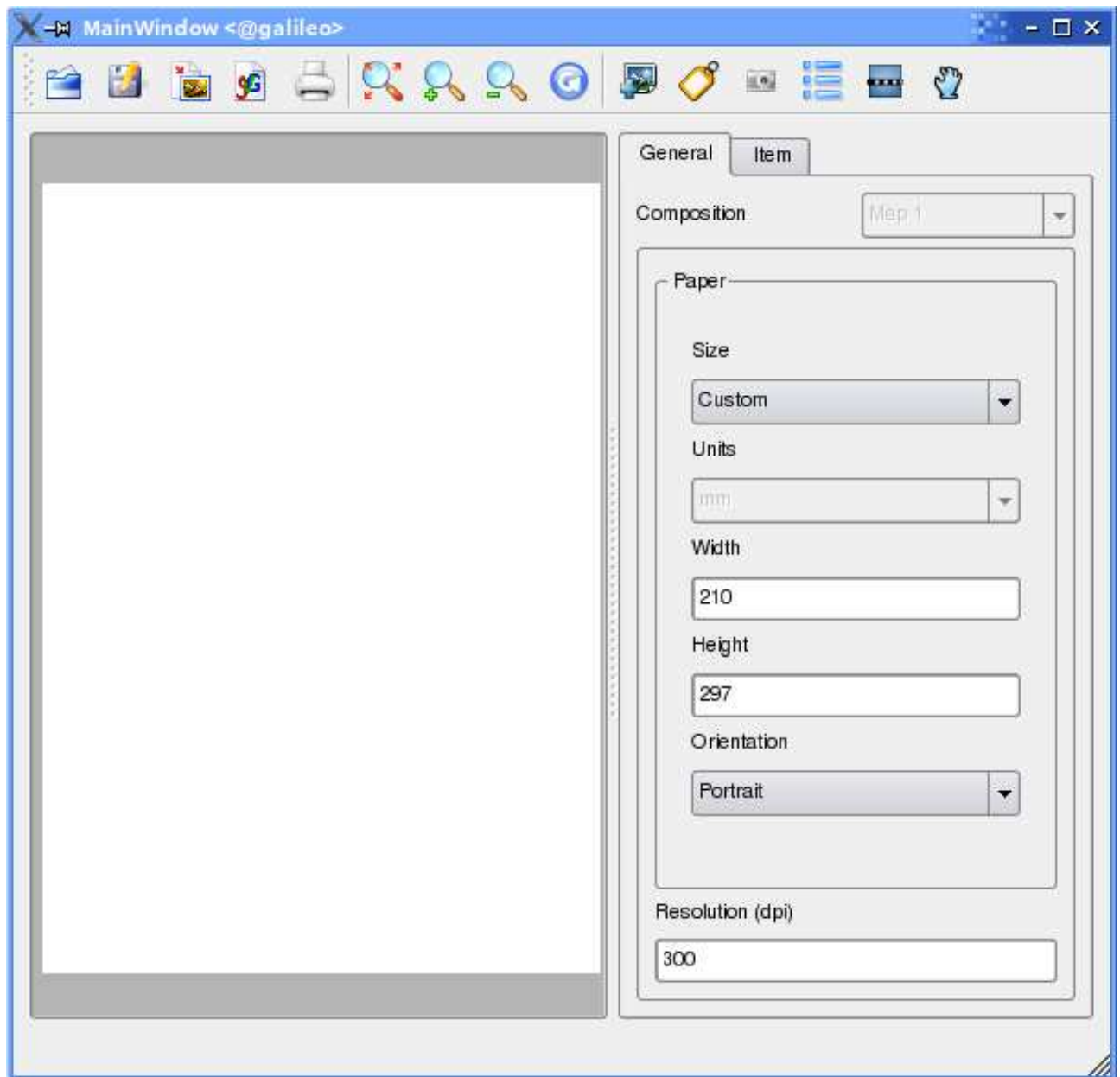
#### 10.1.1 Eine Karte zum Print Composer hinzufügen



Um Layer im Kartenfenster des Print Composers darzustellen, klicken Sie auf den Knopf **Neue Karte hinzufügen** in der Werkzeugleiste. Ziehen Sie dann mit der Maus ein Rechteck im Kartenfenster auf, um die aktuell visualisierten Layer darzustellen. Sie können die Größe des Kartenfensters im Print Composer nachträglich verändern, indem sie auf den Knopf **Selektiere/Verschiebe Eintrag** klicken, mit der Maus das Kartenfenster auswählen und die Größe verändern. Sie können die Größe auch verändern, indem Sie das Element auswählen und die Breite und Höhe dann im Reiter **Eintrag** anpassen.

Das Kartenfenster des Print Composers ist mit dem Kartenfenster von QGIS verbunden. Wenn Sie im Kartenfenster von QGIS etwas verändern (zoomen, Layer hinzufügen usw.) können Sie das Kar-

Abbildung 21: Print Composer



tenfenster des Print Composers anpassen mit dem Knopf *Ausdehnung* setzen. Außerdem können Sie auch den Maßstab der Karte definieren:

1. Wählen Sie *Maßstab* (*Ausdehnungsberechnung*) aus dem *Setzen* Menü.
2. Fügen Sie den Maßstab Nenner in die Maßstab-Box ein
3. Drücken Sie *Enter*

### 10.1.2 Weitere Elemente zum Print Composer hinzufügen



Bereits vorhandene QGIS Schablonen können geladen und angepasst werden. Um eine existierende Schablone zu laden, klicken Sie auf den Knopf *Öffne Schablone*. Wählen Sie eine aus und passen Sie diese an.



Um ein Logo, einen Nordpfeil oder irgendein anderes Bild zu ergänzen, klicken Sie auf den Knopf *Bild hinzufügen*. Das Bild wird dann in das Kartenfenster des Print Composers eingefügt und Sie können es dann verschieben.



Eine Legende kann hinzugefügt und die anzuzeigenden Vektorlayer ausgewählt werden. Um eine Legende zu ergänzen, klicken Sie auf den Knopf *Neue Vektorlegende hinzufügen*. Die Legende wird dann in das Kartenfenster des Print Composers eingefügt und Sie können es dann verschieben. Klicken Sie auf den Reiter *Eintrag*, um das Erscheinungsbild der Legende anzupassen.



Um eine Maßstabsleiste zu ergänzen, klicken Sie auf den Knopf *Füge neuen Maßstab hinzu*. Benutzen Sie den Reiter *Eintrag*, um das Erscheinungsbild der Maßstabsleiste anzupassen.



Sie können Beschriftungen ergänzen, indem Sie auf den Knopf *Neue Beschriftung hinzufügen* klicken. Benutzen Sie den Reiter *Eintrag*, um das Erscheinungsbild der Beschriftung anzupassen.

Abbildung 22 zeigt den Print Composer mit einem fertigen Layout mit mehreren Elementen.

### 10.1.3 Weitere Funktionen

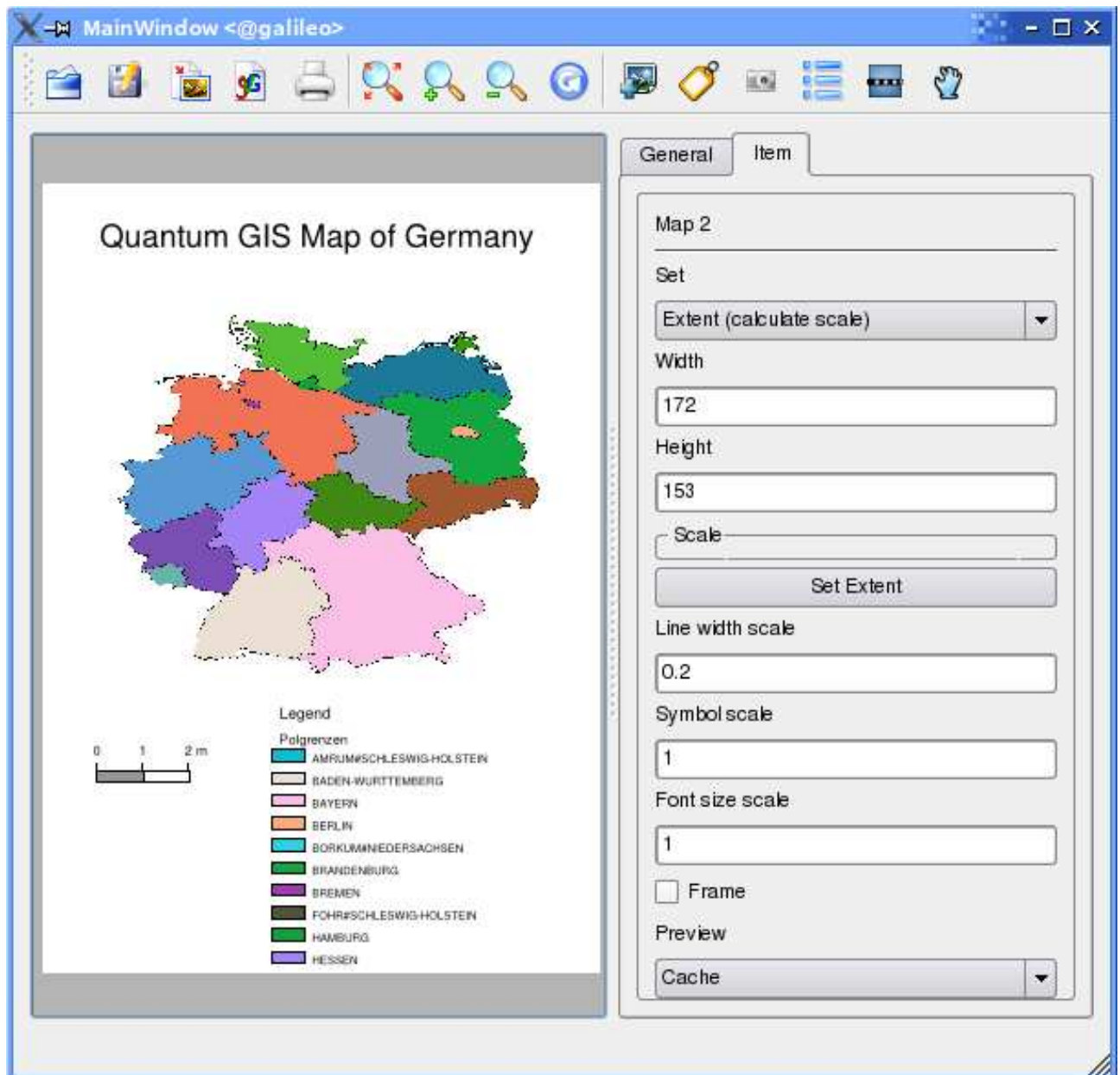
Der Print Composer bietet die Möglichkeit zur Navigation. Um in die Karte hinein zu zoomen, klicken Sie auf den Knopf *Hineinzoomen*. Das Kartenfenster des Print Composers wird mit dem Faktor 2 neu skaliert. Benutzen Sie die Bildlaufleiste, um das Kartenfenster neu zu justieren. Zum Hinauszoomen klicken Sie auf den Knopf *Hinauszoomen*.


Wenn das Kartenfenster des Print Composers nicht anzeigt, was Sie erwarten, klicken Sie auf den Knopf *Aktualisiere Ansicht*, um den Inhalt neu zu visualisieren.


### 10.1.4 Eine Ausgabe erzeugen

Mit dem Print Composer kann eine Karte gedruckt, ins PNG-Bildformat oder ins SVG exportiert werden. Die einzelnen Funktionen können Sie über die Knöpfe in der Werkzeugleiste ansprechen.

Abbildung 22: Print Composer mit Kartenfenster, Legende, Maßstab und Beschriftung



 Um eine Schablone zu sichern, klicken Sie auf den Knopf **Speichere Schablone als...** Speichern Sie die Schablone auf ihrem Rechner und benutzen Sie diese für eine andere Karte wieder.

 Um eine Karte aus dem Kartenfenster des Print Composers als PNG zu speichern, klicken Sie auf den Knopf **Exportiere als Bild**.



Um eine Karte aus dem Kartenfenster des Print Composers als SVG (Scalable Vector Graphic) zu speichern, klicken Sie auf den Knopf `Exportiere als SVG`. **Bemerkung:** In der Version 0.9.1 ist die SVG Ausgabe noch sehr einfach gehalten. Dies ist kein QGIS-Problem, sondern hängt mit der aktuellen Qt-Bibliothek zusammen und wird sich in Zukunft verbessern.

---

## 11 Plugins benutzen

### 11.1 Eine Einführung in das Nutzen der Plugins

QGIS wurde mit einer Plugin-Architektur geplant. Dies erlaubt das Einfügen weiterer Features/Funktionalitäten in das Programm. Viele der Funktionalitäten in QGIS sind bereits als Plugins implementiert.

Es wird zwischen 2 Arten von Plugins in QGIS unterschieden: den Kernplugins sowie den durch Benutzer beigesteuerten. Ein Kernplugin wird durch das QGIS Entwicklerteam betreut und ist Teil jeder QGIS Distribution. Ein von Benutzern beigesteuertes Plugin ist ein externes Plugin, das durch einen individuellen Autor gepflegt und betreut wird. Das SVN-Archiv von QGIS enthält auch einige beigesteuerte Plugins. (<http://svn.qgis.org>)

#### 11.1.1 Ein Plugin finden und installieren

Wenn Sie QGIS installieren werden alle Kernplugins mit installiert (vgl. Abschnitt 11.1.4).

Typischerweise werden die durch Benutzer beigesteuerten Plugins nur als Quellpaket verteilt und müssen selbst kompiliert werden. Grundsätzlich sind im Quellpaket Anleitungen zum Kompilieren und Installieren enthalten.

#### 11.1.2 Plugins verwalten

Das Verwalten von Plugins bedeutet maßgeblich das Laden und Entladen der Module in QGIS. Geladene Plugins werden sich von QGIS 'gemerkt' und beim nächsten Starten der Anwendung erneut geladen.

Um die Plugins zu verwalten, öffnen Sie den *Plugin Manager* aus dem Menueintrag *Plugins*. Der Pluginmanager zeigt alle verfügbaren Plugins und ihren Status (geladen oder nicht geladen). Abb. 23 zeigt den Dialog mit dem Pluginmanager

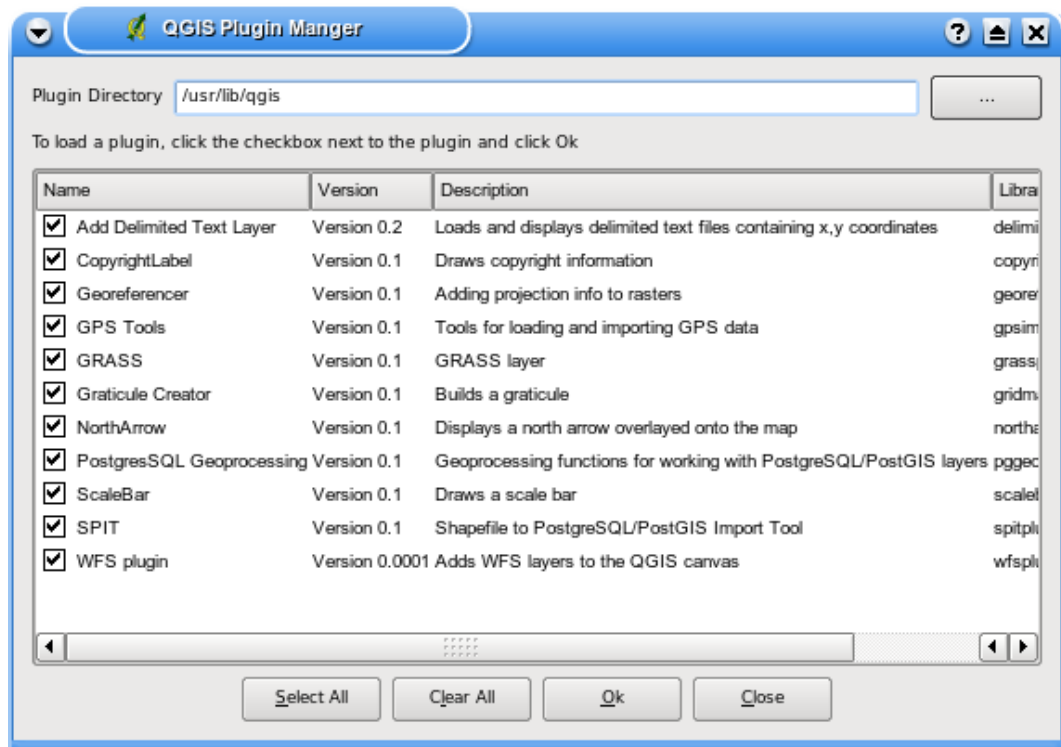
Typischerweise werden alle QGIS-Plugins im gleichen Ordner installiert. Dieser Ordner steht im Text des Plugin-Ordners. Sie können jedoch QGIS auch unter Angabe eines anderen Ordners zum Laden von Plugins von einer anderen Stelle bringen.

#### 11.1.3 Datenprovider

Datenprovider sind 'spezielle' Plugins, welche die Verbindung zur Datenquelle erstellen. Standardmäßig unterstützt QGIS Postgis-Layer und disk-basierte Datenquellen, die von der Bibliothek



Abbildung 23: Plugin Manager

**Tip 34 ABSTÜRZENDE PLUGINS**

Falls QGIS während des Startens abstürzt, kann der Grund ein kaputtes Plugin sein. Um kein Plugin zu laden, können Sie auch die gespeicherten Einstellungen direkt editieren (vgl. Abschnitt 3.8). Finden Sie dazu den Block der Plugins und ändern Sie alle Werte auf 'false'; dann wird das entsprechende Plugin nicht mehr geladen.

Auf einem GNU/Linux-System können Sie unter `$HOME/.config/QuantumGIS/qgis.conf` beispielsweise das Plugin zum Hinzufügen von Kommaseparierten Textdateien (CSV) wie folgt deaktivieren:

`Add Delimited Text Layer=false`. Tun Sie dies nun für alle Plugins im Block [Plugins] stehen. Danach können Sie vom Plugin Manager aus jedes einzelne Plugin laden und herausfinden, welches das Probleme machende Plugin ist.










GDAL/OGR unterstützt werden (siehe Anhang A.1). Ein Datenprovider Plugin erweitert QGIS beim Nutzen von anderen Datenquellen.

Datenprovider Plugins werden automatisch beim Starten im System registriert. Sie werden nicht durch den Pluginmanager verwaltet, werden jedoch im Hintergrund benutzt, wenn ein entsprechender Datenlayer in QGIS angezeigt werden soll.

### 11.1.4 Kernplugins

Derzeit enthält QGIS 9 Kernplugins, welche über den Pluginmanager geladen werden können. Tabelle 5 listet jedes Kernplugin mit einer kleinen Beschreibung des Nutzens sowie des Werkzeugleisten-symbols auf. Beachten Sie, dass das GRASS-Plugin in der Tabelle nicht enthalten ist, da es seine eigene Werkzeugleiste installiert (vgl. Abschnitt 8 um Näheres über die Funktionalität des GRASS-Plugins zu erfahren.).

**Tabelle 5:** QGIS Kernplugins

Symbol	Plugin	Beschreibung
	Copyrightschild	Zeigt ein Copyrightschild im Kartenbild
	CSV-Text	Lädt eine kommaseparierte Textdatei (CSV) mit X- und Y-Koordinaten als ein Punktlayer
	GPS Werkzeuge	Lädt und stellt GPS-Daten dar
	Gradnetzgenerator	Erzeugt ein Gradnetz (Lat/Long) und speichert es in ein Shapefile
	Maßstabsbalken	Fügt einen Maßstabsbalken dem Kartenbild hinzu
	Nordpfeil	Fügt einen Nordpfeil dem Kartenbild hinzu
	PostgreSQL Geoverarbeitung	Puffert ein PostGIS-Layer
	SPIT	Shapefile nach PostGIS Import Werkzeug - importiert Shapefiles in PostgreSQL
	Georeferenzierwerkzeug <sup>a</sup>	Georeferenziert Rasterdatenlayer

<sup>a</sup>Das Georeferenzier-Plugin steht nur zur Verfügung, wenn die gsl-Bibliotheken und Header-Dateien beim Kompilieren von QGIS installiert haben. Bitte überprüfen Sie Details im Kapitel D.3.1

---

#### **Tip 35** PLUGINEINSTELLUNGEN IN EINEM PROJEKT SPEICHERN.

Wenn Sie ein .qgs-Projekt sichern, werden alle Einstellungen, die Sie am Nordpfeil, Maßstab und Copyright vorgenommen haben mit dem Projekt abgespeichert. Beim nächsten Laden werden die Einstellungen wieder hergestellt.

---

### 11.1.5 Externe Plugins

QGIS enthält auch ein paar extern entwickelte Plugins. Diese befinden sich nicht in der Standardversion, können aber bei Bedarf kompiliert und in QGIS integriert werden.

Derzeit sind die extern entwickelten Plugins nur über SVN zugänglich. Eine Liste der aktuell verfüg-

baren Plugins können Sie hier herunterladen:

```
svn co https://svn.qgis.org/repos/qgis/trunk/external_plugins external_qgis_plugins
```

Mit diesem Befehl wird ein Ordner `external_qgis_plugins` in Ihrem aktuellen Verzeichnis erstellt. Jeder Unterordner enthält eigene Instruktionen zum Kompilieren und Installieren. Lesen Sie diese sorgfältig, um das jeweilige Plugin zu erstellen.

### 11.1.6 Plugin-Vorlagen

Wenn Sie Ihr eigenes QGIS-Plugin erstellen wollen, beinhaltet QGIS ein Skript, welches Sie bei der Integration Ihres Plugins und der Erstellung der Verzeichnisstruktur innerhalb des QGIS-Quellbaums unterstützt. Das Skript finden Sie unter `QGIS/src/plugins/plugin_builder.pl`.

Sie müssen lediglich noch die neuen Funktionen in Ihr Plugin hineinprogrammieren (und das Plugin natürlich dem QGIS Entwicklerteam zur Verfügung stellen).

Daneben hält das QGIS-Wiki (<http://wiki.qgis.org>) und der QGIS-Blog (<http://blog.qgis.org>) weitere nützliche Artikel zum Schreiben eigener Plugins bereit. Besuchen Sie die Webseite für weitere Details!

## 11.2 Benutzung der Dekorations-Plugins

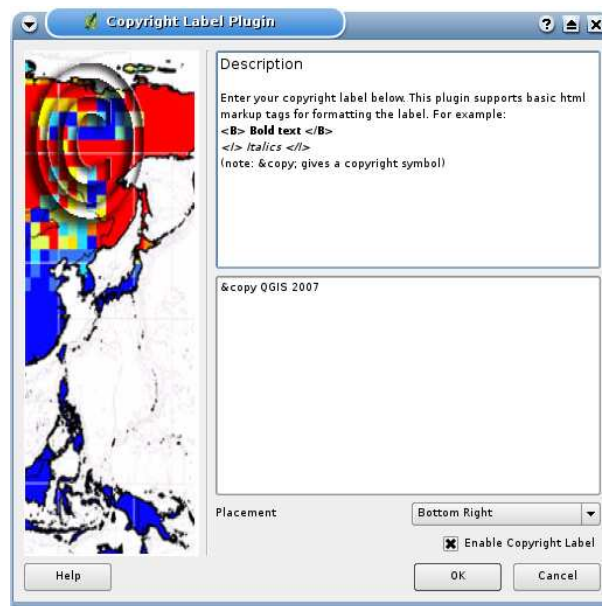
Das Dekorations Plugin beinhaltet im Detail die folgenden Einzelmodule:

- Das Urheberrechts/Copyright Plugin
- Das Nordpfeil Plugin
- sowie das Maßstabsbalken Plugin

Diese Plugins dienen der 'Verzierung' von Karten und bieten die Möglichkeit, diesen kartographische Elemente hinzuzufügen.

### 11.2.1 Das Copyright Plugin

Abbildung 24: Das Copyright Plugin



Die Bezeichnung dieses Plugins ist ein wenig irreführend, denn Sie können damit nicht nur das Copyright Zeichen, sondern auch jeden anderen beliebigen Text zu einer Karte hinzufügen.

1. Vergewissern Sie sich, dass das Plugin geladen ist.
2. Klicken Sie auf das *Copyright Label* Werkzeug in der Plugin Werkzeugleiste.
3. Geben Sie den Text ein, den Sie der Karte hinzufügen möchten. Wie im gezeigten Beispiel können Sie hierbei auch HTML Text verwenden.
4. Wählen Sie im drop-down Menü aus, wo die Beschriftung auf der Karte platziert werden soll.

5. Vergewissern Sie sich, dass das "Enable Copyright Label" Kästchen markiert ist.
6. Klicken Sie dann auf *OK*.

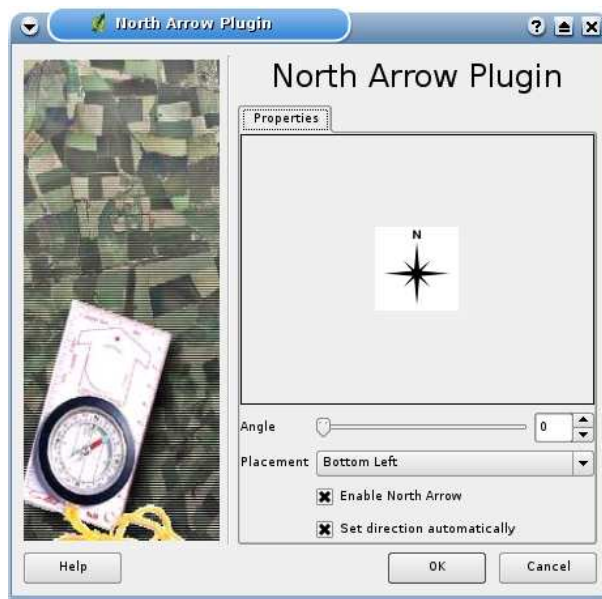
In obigem Beispiel ist die erste Zeile FETT geschrieben. Die zweite Zeile beinhaltet ein Copyright Symbol, gefolgt von unserem Firmennamen in kursiver Schrift.

### 11.2.2 Das Nordpfeil Plugin

Das Nordpfeil Plugin platziert auf unserem Kartenausschnitt einen einfachen Nordpfeil. Bislang steht hierfür lediglich eine Pfeilform zur Verfügung. Sie können die Richtung des Nordpfeiles entweder selbst anpassen oder dies automatisch von QGIS bewerkstelligen lassen. Wenn Sie es QGIS überlassen die Richtung zu bestimmen, so wägt das Programm ab in welche Richtung der Nordpfeil idealerweise zeigen sollte.

Für die Platzierung des Pfeils stehen Ihnen vier Möglichkeiten zur Verfügung. Diese entsprechen den vier Ecken Ihres Kartenausschnitts.

**Abbildung 25:** Das Nordpfeil Plugin



### 11.2.3 Das Maßstabsbalken Plugin

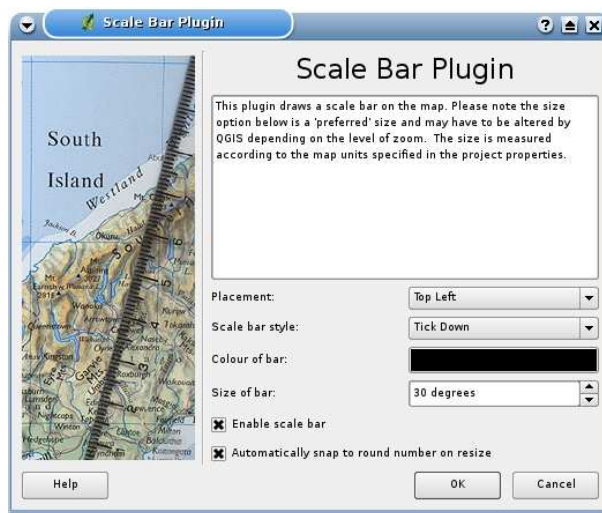
Das Maßstabsbalken Plugin fügt dem Kartenausschnitt einen Maßstabsbalken hinzu. Dabei können Sie sowohl dessen Aussehen und Platzierung, als auch seine Beschriftung bestimmen.

QGIS unterstützt hierbei jedoch lediglich die Darstellung des Balkens in denjenigen Einheiten, die auch in Ihren entsprechenden Karten verwendet werden. Dies bedeutet, dass Sie einer Karte deren Layer der Einheit Meter entsprechen, keinen Maßstabsbalken in den Einheiten Fuss oder Zoll hinzufügen können. Die Einheit des Maßstabsbalkens entspricht also immer derjenigen der Karte. Gleiches gilt, wenn Sie Dezimalzahlen (Latitude/Longitude) verwenden, auch hier können Sie keinen Maßstabsbalken verwenden, der die Distanzen in der Einheit Meter darstellt.

Das Hinzufügen eines Maßstabsbalkens:

1. Öffnen Sie die Plugin Werkzeugleiste und klicken dort auf die Schaltfläche des *Scale Bar* Werkzeugs.
2. Wählen Sie im drop-down Menü aus, wo der Balken platziert werden soll.
3. Wählen Sie das Aussehen des Balkens aus.
4. Wählen Sie die gewünscht Balkenfarbe, oder benutzen Sie die Voreinstellung schwarz.
5. Legen Sie die Größe des Balkens und dessen Beschriftung fest.
6. Vergewissern Sie sich, dass die Schaltfläche "Enable scale bar" ausgewählt ist.
7. Zusätzlich können Sie festlegen, ob bei einer Größenveränderung des Kartenausschnitts automatisch auf ganze Zahlen gerundet werden soll.
8. Klicken Sie abschließend auf *OK*.

Abbildung 26: Das Maßstabsbalken Plugin



## 11.3 Das GPS-Plugin benutzen

### 11.3.1 Was ist GPS?

GPS, das 'Global Positioning System', ist ein satellitengestütztes System, das jeder Person mit einem GPS-Empfänger erlaubt, seine exakte Position anhand von Koordinaten irgendwo auf der Welt anzeigen zu lassen. Es wird bei der Navigation von Flugzeugen, Booten oder auch von Wanderern als Hilfsmittel benutzt. Der GPS-Empfänger benutzt dabei Signale der Satelliten um Längen- und Breitengrad und manchmal auch die aktuelle Höhe zu berechnen. Die meisten Empfänger können die Positionen auch in Form von *Wegpunkten*, *Routen* oder *Spuren* speichern. QGIS stellt Wegpunkte als Punktdaten und Routen sowie Spuren als Vektorlinien dar.

### 11.3.2 GPS-Daten aus einer Datei laden

Es gibt viele verschiedene Datenformate zum Speichern von GPS-Daten. Das von QGIS unterstützte Format ist GPX (GPS eXchange format), ein Standardformat, welches Wegpunkte, Routen und Spuren in einer Datei enthalten kann.



Um eine GPX-Datei zu laden, müssen Sie das GPS Werkzeug-Plugin benutzen. Wenn dieses Plugin aktiviert ist, erscheint ein Knopf mit einem kleinen GPS-Empfänger in der Werkzeugleiste von QGIS. Klicken Sie auf den Knopf, um den Dialog *GPS Werkzeuge* zu starten (vgl. Abbildung 27).

Abbildung 27: Der Dialog *GPS Werkzeuge*



Benutzen Sie den Knopf [...], um eine GPX-Datei auszuwählen und mit der Checkbox den Dateityp



zu bestimmen, der aus der GPX-Datei geladen werden soll. Jeder Dateityp (Wegpunkte, Routen und Spuren) wird in einen eigenen Layer geladen, wenn Sie auf den Knopf *OK* klicken.

### 11.3.3 GPSTabel

Weil QGIS nur das GPX-Format unterstützt, müssen Sie einen Weg finden, um andere GPS-Datenformate nach GPX zu konvertieren. Dies können Sie in vielen Fällen mit der Freien Software GPSTabel erledigen, erhältlich unter der URL: <http://www.gpsbabel.org>. Diese Software kann auch GPS-Daten von Ihrem PC zu Ihrem GPS-Empfänger transferieren und umgekehrt. QGIS benutzt GPSTabel auch dazu, daher ist es ratsam, es zu installieren. Wenn Sie aber lediglich GPX-Dateien in QGIS laden möchten, geht dies auch ohne GPSTabel. Version 1.2.3 funktioniert auf jeden Fall mit QGIS, neuere aber wahrscheinlich auch.

### 11.3.4 GPS-Daten importieren

Um GPS-Daten aus einer Datei, die nicht im GPX-Format vorliegt zu importieren, benutzen Sie das Programm *Importiere andere Datei* im Dialog *GPS Werkzeug*. Sie wählen dann die Datei, die importiert werden soll aus, von welchem Datenformat Sie importieren möchten und wo die konvertierte GPX-Datei unter welchem Namen gespeichert werden soll.

Wenn Sie eine Datei auswählen müssen Sie auch das entsprechende Dateiformat über das Menü im Dialogfenster *Importiere andere Datei* selektieren (vgl. Abbildung 28). Nicht für alle Datenformate werden die drei GPS-Datentypen Wegpunkte, Routen und Spuren unterstützt. Manchmal sind es nur ein oder zwei.

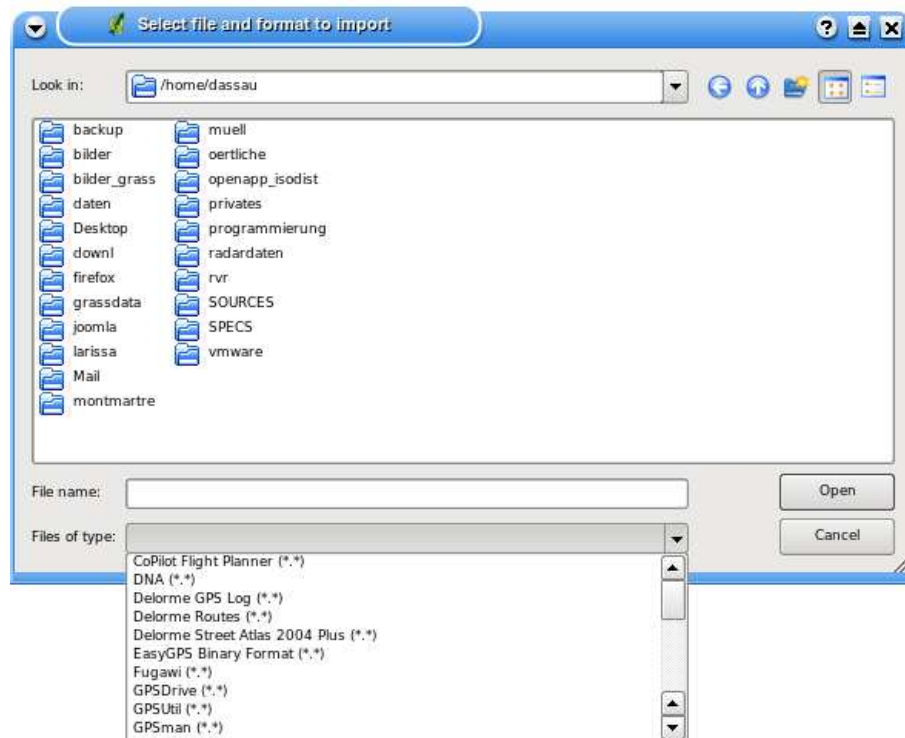
### 11.3.5 GPS-Daten von einem Empfänger herunterladen

QGIS kann mit dem Programm GPSTabel GPS-Daten direkt von einem GPS-Empfänger in einen Vektorlayer laden. Dazu können Sie das Programm *Von GPS herunterladen* verwenden (vgl. Abbildung 29), indem Sie den Typ Ihres GPS-Empfängers angeben, den Verbindungsport, den Namen der GPX-Datei, in die die Daten konvertiert werden sollen und schließlich den Namen des neuen Vektorlayers in QGIS.

Durch die Angabe des Typs Ihres GPS-Empfängers legen Sie fest, wie GPSTabel mit dem Gerät kommuniziert. Wenn kein vorhandener Typ mit Ihrem Empfänger funktioniert, können Sie einen eigenen, neuen Typ erstellen (vgl. Abschnitt 11.3.7).

Der Verbindungsport ist ein Dateiname oder ein anderer Name, den Ihr System als Referenz für den physischen Port benutzt, über den eine Verbindung zum GPS-Empfänger hergestellt wird. Unter GNU/Linux ist das etwa wie `/dev/ttyS0` oder `/dev/ttyS1` und unter Windows ist es COM1 oder COM2.



**Abbildung 28:** Dialog zum Importieren anderer GPS-Formate

Wenn Sie auf *OK* klicken, werden die Daten vom GPS-Empfängers heruntergeladen und in QGIS als Vektorlayer dargestellt.

### 11.3.6 GPS-Daten auf einen Empfänger hochladen

Sie können auch einen Vektorlayer aus QGIS auf einen GPS-Empfänger hochladen, indem Sie das Programm *In GPS hochladen* verwenden. Der Vektorlayer muss dazu ein GPX-layer sein. Sie wählen dazu einen entsprechenden Layer aus, den Typ Ihres GPS-Empfängers und den Verbindungsport. Genau wie beim Programm *Von GPS herunterladen* können Sie bei Bedarf auch einen eigenen, neuen Empfänger-Typ erstellen, wenn Ihr Gerät nicht in der Liste auftaucht.

Dieses Werkzeug ist sehr nützlich, besonders im Zusammenspiel mit den Vektorfunktionen von QGIS. Sie können eine Karte laden, ein paar Wegpunkte oder Routen digitalisieren und es dann auf Ihren GPS-Empfänger hochladen.

Abbildung 29: Das Programm 'Von GPS herunterladen'



### 11.3.7 Neue GPS-Gerätetypen definieren

Es gibt eine große Vielfalt an GPS-Empfängern, und die QGIS-Entwickler können nicht alle testen. Wenn Sie also ein Gerät haben, das nicht mit einem der bereits vorhandenen Gerätetypen funktioniert, können Sie selbst einen erstellen. Dazu klicken Sie auf den Knopf *Editiere Geräte* in den Programmen *In GPS hochladen* oder *Von GPS herunterladen*.

Um ein neues Gerät zu definieren, klicken Sie auf den Knopf *Neues Gerät*, geben einen Namen sowie die Kommandos zum hinauf- und hinunterladen der unterschiedlichen Datentypen (Wegpunkte, Routen und Spuren) ein und klicken dann auf *Gerät zum Hinaufladen*. Danach wird der Name des Gerätes in den Programmen *In GPS hochladen* und *Von GPS herunterladen* zur Verfügung stehen.

Das Kommando zum hinunterladen der unterschiedlichen Datentypen wird benutzt, um die Daten von Ihrem GPS-Empfänger in eine GPX-Datei zu transferieren. Es wird wahrscheinlich ein GPSBabel Kommando sein, kann aber auch ein anderes Kommandozeilenprogramm sein, das diese Aufgabe erfüllt. QGIS wird die Schlüsselwörter *%type*, *%in* und *%out* ersetzen, wenn das Kommando ausgeführt wird.

*%type* wird ersetzt durch “-w”, wenn Sie Wegpunkte herunterladen, “-r” wenn es eine Route ist und “-t”, wenn es sich um Spuren handelt. GPSBabel erfährt dadurch, um welchen GPS-Datentyp es sich handelt.

*%in* wird ersetzt durch den Namen des Verbindungsports und *%out* durch den Namen, den Sie für die GPX-Datei gewählt haben. Wenn Sie also einen neuen Gerätetyp mit dem Kommando “gpsbabel

`%type -i garmin -o gpx %in %out` (es handelt sich hierbei um das Standard Kommando für einen "Garmin Serial") definieren und diesen benutzen, um Wegpunkte von Port `/dev/ttyS0` in die Datei `output.gpx` zu schreiben, dann ersetzt QGIS die Schlüsselworte und startet das Kommando `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

Das Kommando hinaufladen wird benutzt, um verschiedene Datentypen (Wegpunkte, Routen und Spuren) auf Ihren GPS-Empfänger zu transferieren. Es werden dazu die gleichen Schlüsselworte benutzt, nur dass `%in` durch den Namen der hochzuladenen GPX-Datei und `%out` durch den Namen des Verbindungsports ersetzt wird. Sie können mehr über GPSTabel und seine Funktionen unter der URL: <http://www.gpsbabel.org> erlernen.

Wenn Sie einmal einen eigenen Gerätetypen erstellt haben, wird er in der Liste der GPS-Geräte angezeigt werden.

## 11.4 Das CSV-Plugin benutzen

Mit dem 'CSV'-Plugin<sup>6</sup> können Datenspalten aus einer Textdatei als ein Layer in QGIS geladen werden.

### 11.4.1 Anforderungen

Um Datenspalten aus einer Textdatei in QGIS zu laden, muss diese Textdatei bestimmte Eigenschaften aufweisen:

1. Eine Kopfzeile mit den Spaltennamen. Diese Kopfzeile muss die erste Zeile der Datei sein
2. Die Textdatei muss mindestens eine Spalte mit X- und eine mit Y-Koordinaten enthalten. Die Bezeichnungen in der Kopfzeile für diese Spalten können beliebig sein.
3. Die X- und Y-Koordinaten müssen als Zahlen angegeben sein. Das Koordinatensystem spielt keine Rolle.

Ein Beispiel für einen Textdatei könnte so aussehen:

```
name|latdec|longdec|cell|
196 mile creek|61.89806|-150.0775|tyonek d-1 ne|
197 1/2 mile creek|61.89472|-150.09972|tyonek d-1 ne|
a b mountain|59.52889|-135.28333|skagway c-1 sw|
apw dam number 2|60.53|-145.75167|cordova c-5 sw|
apw reservoir|60.53167|-145.75333|cordova c-5 sw|
apw reservoir|60.53|-145.75167|cordova c-5 sw|
aaron creek|56.37861|-131.96556|bradfield canal b-6|
aaron island|58.43778|-134.81944|juneau b-3 ne|
aats bay|55.905|-134.24639|craig d-7|
```

Einige weitere Anmerkungen zu Textdateien:

1. Die Beispieldatei verwendet | als Trennzeichen. Es können auch andere Zeichen zum Trennen der Spalten verwendet werden.
2. Die erste Zeile ist die Kopfzeile. Sie enthält die Spaltennamen name, latdec, longdec und cell
3. Anführungszeichen (") dürfen nicht als Trennzeichen benutzt werden
4. Die X-Koordinaten sind in der Spalte *longdec* enthalten
5. Die Y-Koordinaten sind in der Spalte *latdec* enthalten

<sup>6</sup>CSV steht für comma-separated-value, also Textdateien, dessen Spalten mit einem definierten Trennzeichen versehen sind

### 11.4.2 Das Plugin benutzen

Um das Plugin zu benutzen, müssen Sie QGIS starten und den Plugin Manager aufrufen, um das Plugin *Getrennter Text* zu aktivieren:

Starten Sie QGIS, öffnen Sie den Plugin Manager über das Menü *Plugins->Plugin Manager* in der Menüleiste. Der Plugin Manager zeigt eine Liste vorhandener Plugins. Sie können Plugins mit der entsprechenden Checkbox aktivieren und deaktivieren und dann auf den Knopf **OK** klicken, wie in Kapitel 11.1.2 beschrieben.


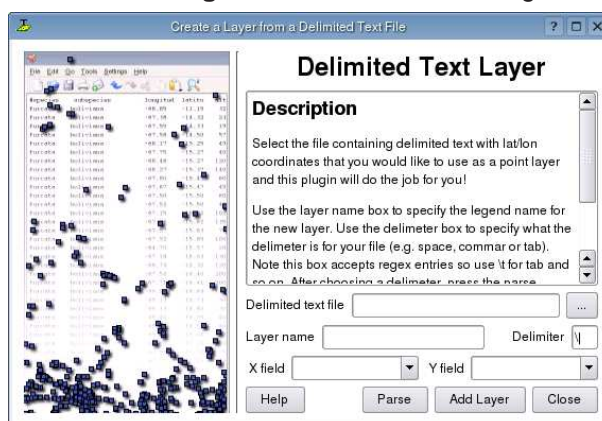

Das ausgewählte Plugin erscheint nun in der Werkzeugleiste:  Klicken Sie auf das Icon, um das Plugin *Getrennter Text* wie in Abbildung 30 zu starten.

Abbildung 30: Getrennter Text Dialog



Zuerst wählen Sie eine Textdatei aus, die Sie laden möchten:  Das Plugin versucht nun automatisch mit Hilfe des zuletzt verwendeten Trennzeichens eine Struktur (Spalten) in der Datei zu finden. Im Beispiel ist es das Trennzeichen `|` (Abbildung 31).

Für diese Textdatei ist das Trennzeichen `|` nicht richtig, sie wird durch TABs getrennt. Beachten Sie, dass die X- und Y-Felder nicht die gültigen Spaltennamen enthalten.

Damit die Textdatei richtig durchsucht werden kann, ändern Sie das Trennzeichen auf `\t`. Dies ist die reguläre Darstellung des Tabulator-Trennzeichens. Nun klicken Sie auf **durchsuchen**. Im unteren Feld ist der Inhalt nun korrekt in die vorkommenden Spalten unterteilt, wie in Abbildung 32 zu sehen.

Wählen Sie nun die Spalten für die X- und Y-Koordinaten aus und tragen einen Namen ein, unter dem die Daten in QGIS angezeigt werden sollen (siehe Abbildung 33). Um die Daten zu sehen, klicken Sie auf **Hinzufügen**. Der Layer verhält sich nun wie jeder andere Vektorlayer in QGIS.

Abbildung 31: Ausgewählte Textdatei

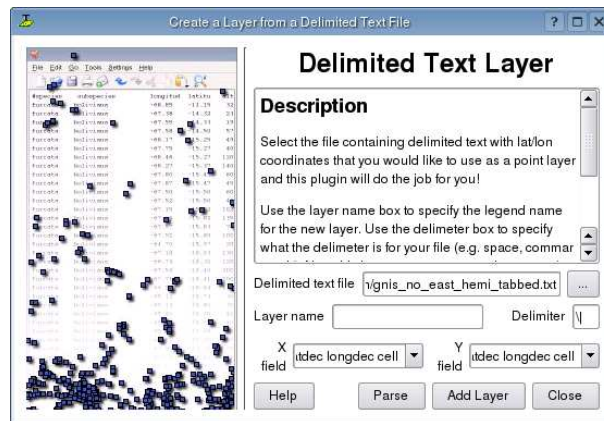


Abbildung 32: In der Textdatei gefundene Spalten

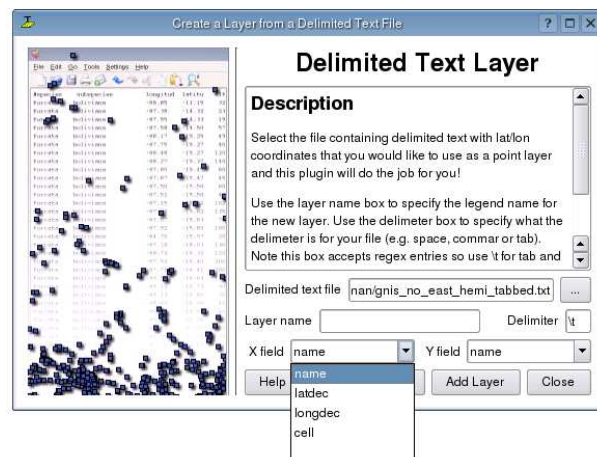
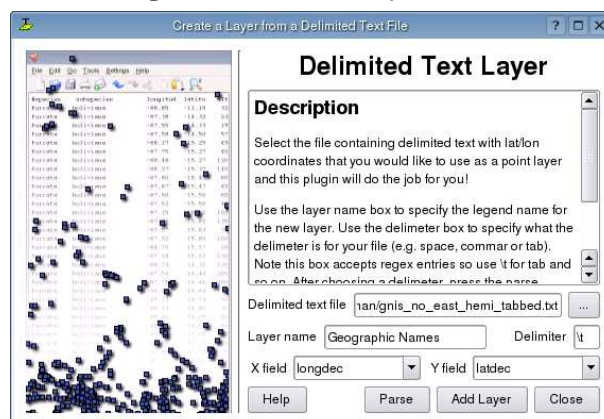


Abbildung 33: Die X- und Y-Spalten auswählen

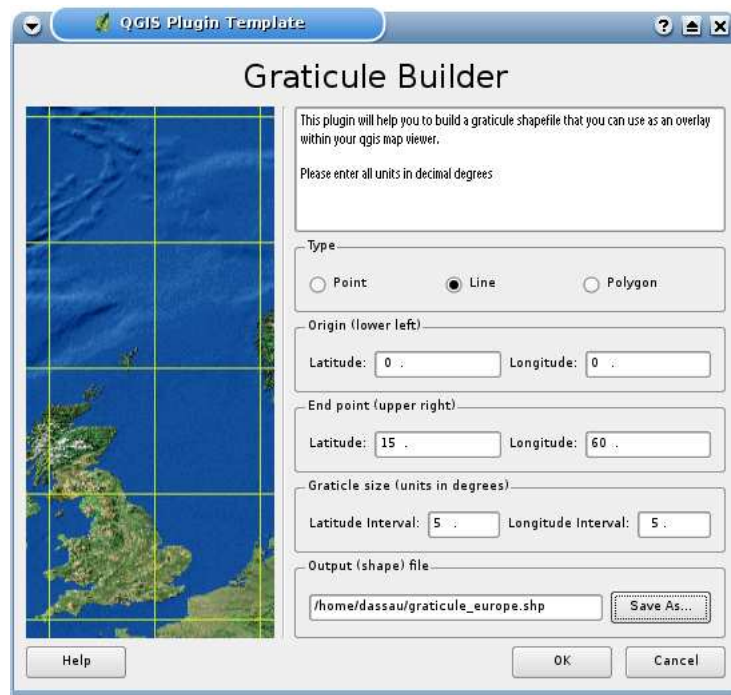




## 11.5 Die Benutzung des Gitternetz Plugins

Das Gitternetz Plugin erlaubt es, für unseren jeweils gewählten Kartenausschnitt ein 'Gitter' aus Punkten, Linien oder Polygonen zu erstellen. Sämtliche Einheiten müssen dabei in Dezimalgraden eingegeben werden. Bei der Ausgabedatei handelt es sich um eine Shape-Datei, die direkt ('on the fly') und passend zu den anderen Daten projiziert werden kann.

**Abbildung 34:** Erstellung eines Gitternetz-Layers



Hier nun ein kurzes Beispiel für die Erstellung eines Gitternetzes:

1. Stellen Sie sicher, dass das Plugin geladen ist.
2. Klicken Sie auf das Gitternetz Werkzeug in der Plugin Funktionsleiste.
3. Wählen Sie aus, ob das Gitter aus Punkten, Linien oder Polygonen bestehen soll.
4. Geben Sie die Breiten- und Längengrade für die obere linke und rechte Ecke des Gitters ein.
5. Geben Sie den Abstand an, der bei der Erstellung des Gitters verwendet werden soll. Sie können dabei unterschiedliche X und Y Werte verwenden (Längengrad, Breitengrad).
6. Geben Sie an, wo und unter welchem Namen die Shape-Datei erstellt werden soll.
7. Klicken Sie dann auf OK um das Gitternetz zu erstellen und fügen Sie dieses abschließend dem Kartenausschnitt hinzu.

## 11.6 Die Benutzung des Georeferenzierungs-Plugin

Das Georeferenzierungs-Plugin erlaubt die Erstellung von world-files für Rasterkarten. Damit das Plugin die world-file Parameter berechnen kann, wählen sie auf einer Rasterkarte Punkte aus und ordnen diesen Punkten die entsprechenden Koordinaten zu. Das Ergebnis der Georeferenzierung wird umso genauer, je mehr Koordinaten Sie dem Plugin zur Verfügung stellen.

Als Beispiel hierfür werden Sie nun ein world-file für eine topografische Karte aus der Gegend Süddakotas erstellen, welche aus dem Spearfish-Datensatz stammt. Diese Karte kann später zusammen mit den erstellten Daten in der GRASS spearfish60 location dargestellt werden. Die topografische Karte steht unter folgender Adresse zum Download bereit. [http://grass.itc.it/sampleddata/spearfish\\_toposheet.tar.gz](http://grass.itc.it/sampleddata/spearfish_toposheet.tar.gz)

Als erstes laden Sie die Datei herunter und entpacken diese mit dem tar-Befehl.

```
wget http://grass.itc.it/sampleddata/spearfish_toposheet.tar.gz
tar xvzf spearfish_toposheet.tar.gz
cd spearfish_toposheet
```

Als nächstes starten Sie QGIS, laden dort das Georeferenzierungs-Plugin und wählen dann die spearfish\_topo24.tif aus.

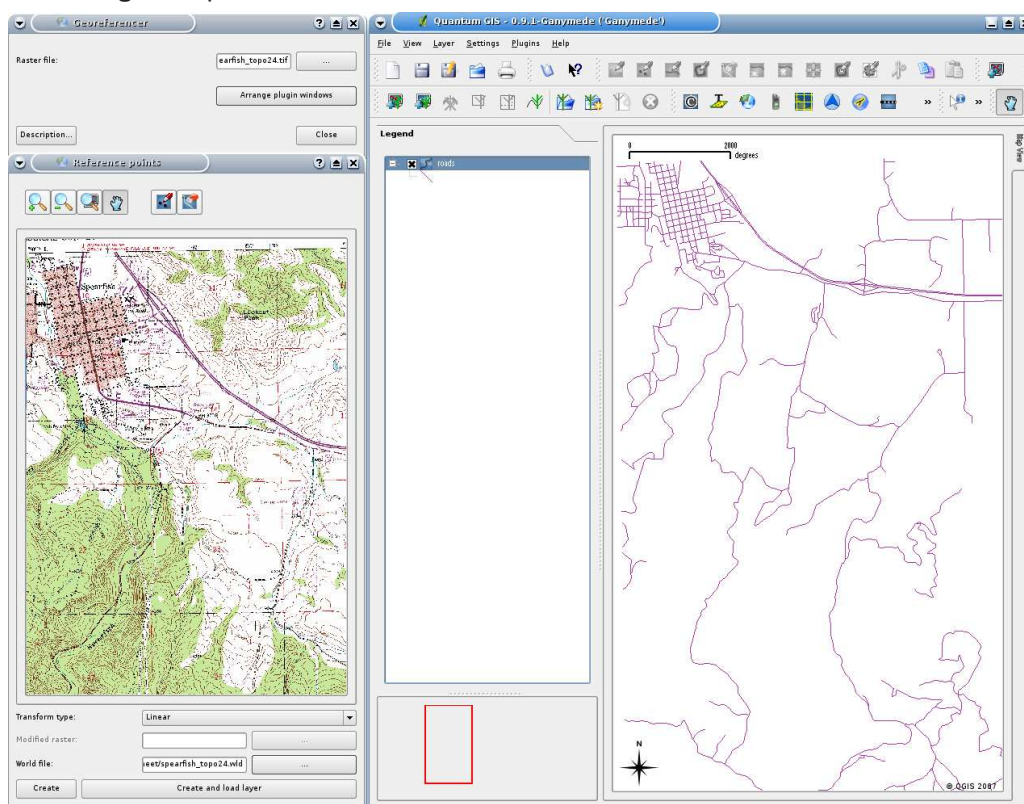
**Abbildung 35:** Ein Bild für die Georeferenzierung auswählen



Dann klicken Sie auf die Schaltfläche **Fenster anordnen** um das ausgewählte Bild im Georeferenzierer zu öffnen und auf dem Desktop gemeinsam mit dem QGIS Kartenfenster optimal anzuordnen.

Über die Schaltfläche **Add Point** können Sie nun damit beginnen dem Rasterbild Punkte und die dazugehörigen Koordinaten hinzuzufügen. Aus diesen Punktkoordinaten errechnet das Plugin dann die entsprechenden world-file Parameter. Je mehr Koordinaten Sie dem Plugin zur Berechnung zur



**Abbildung 36:** Optimales Anordnen des QGIS Kartenfensters mit dem Georeferenzierer

Verfügung stellen, desto genauer wird das Ergebnis. Für die Angabe der nötigen Punktkoordinaten stehen Ihnen zwei unterschiedliche Vorgehensweisen zur Verfügung.

1. Sie klicken auf einen Punkt in der Rasterkarte und geben die X- und Y-Koordinaten von Hand ein.
2. Sie klicken auf einen Punkt in der Rasterkarte und wählen die Schaltfläche `from map canvas` um die X- und Y-Koordinaten mit Hilfe einer georeferenzierten und bereits in QGIS geladenen Karte hinzuzufügen.

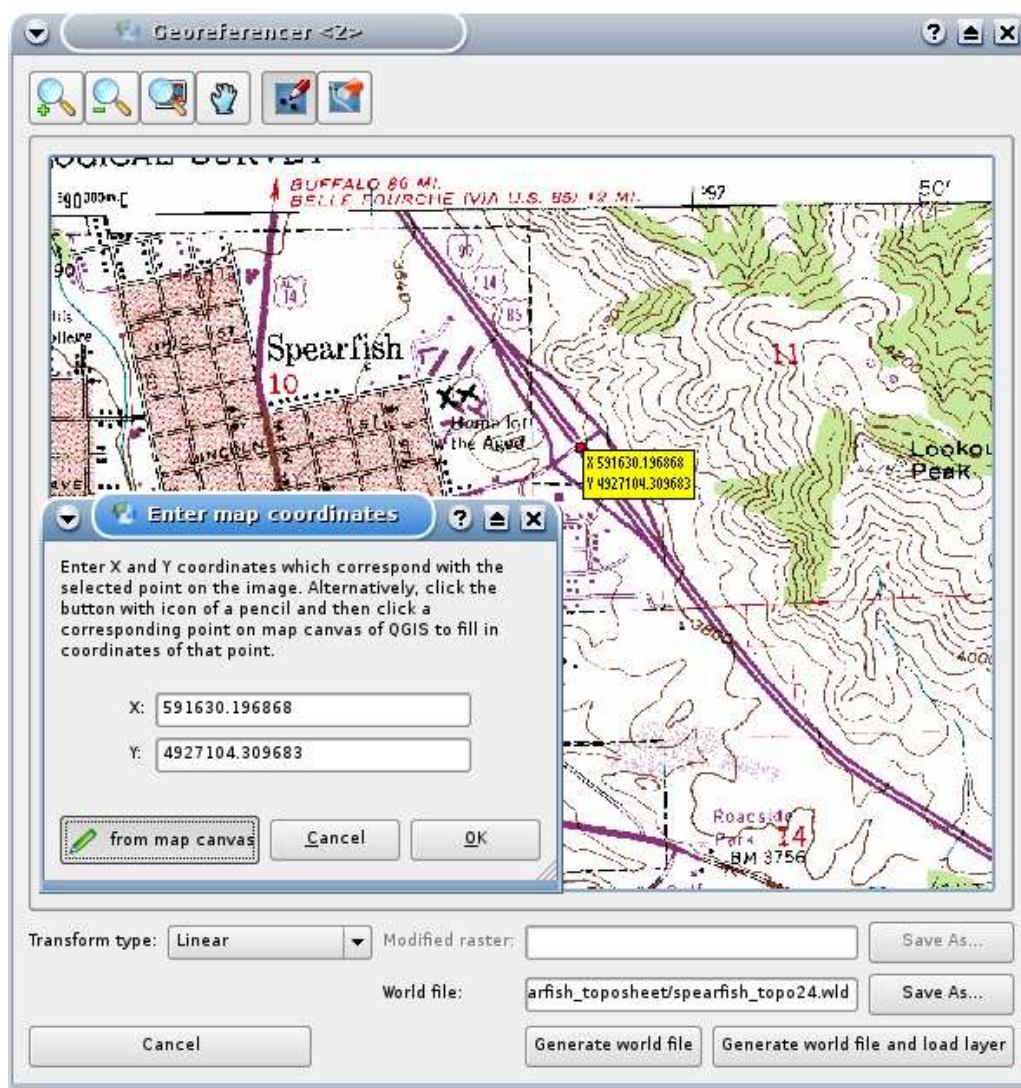
In unserem Beispiel nutzen wir die zweite Option und geben die Koordinaten der ausgewählten Punkte mit Hilfe der Karte `roads` aus der `spearfish60` location ein. Diese Daten stehen unter folgender Adresse zum Download bereit:

[http://grass.itc.it/sampled/spearfish\\_grass60data-0.3.tar.gz](http://grass.itc.it/sampled/spearfish_grass60data-0.3.tar.gz)

Für den Fall, dass Sie nicht wissen, wie Sie die `spearfish60` location über das GRASS Plugin benutzen können, finden Sie im Abschnitt 8 weitere Informationen.

Wie Sie in der Abbildung 37 sehen können, stellt Ihnen das Georeferenzierungs-Werkzeug Schaltflächen zum Vergrößern/Verkleinern, zum Verschieben, sowie zum Hinzufügen und Löschen von

Abbildung 37: Koordinaten für die Georeferenzierung setzen



Punkten in der Karte zur Verfügung.

Nachdem Sie dem Bild eine ausreichende Zahl an Punkten hinzugefügt haben, gilt es nun, für den Georeferenzierungs-Vorgang die gewünschte Koordinaten-Transformationsmethode auszuwählen und die daraus errechneten world-file Daten zusammen mit dem Tiff zu speichern. In unserem Beispiel entscheiden wir uns für eine lineare Transformation, wobei ebenso gut eine Helmert-Transformation verwendet werden könnte.

Die von uns hinzugefügten Punkte werden zusammen mit dem Rasterbild in der Datei `spearfish_topo24.tif.points` gespeichert. Dies erlaubt es uns, das Georeferenzierungs-Plugin wieder zu öffnen und neue Punkte hinzuzufügen oder bestehende Punkte zu löschen, um dadurch das Ergebnis

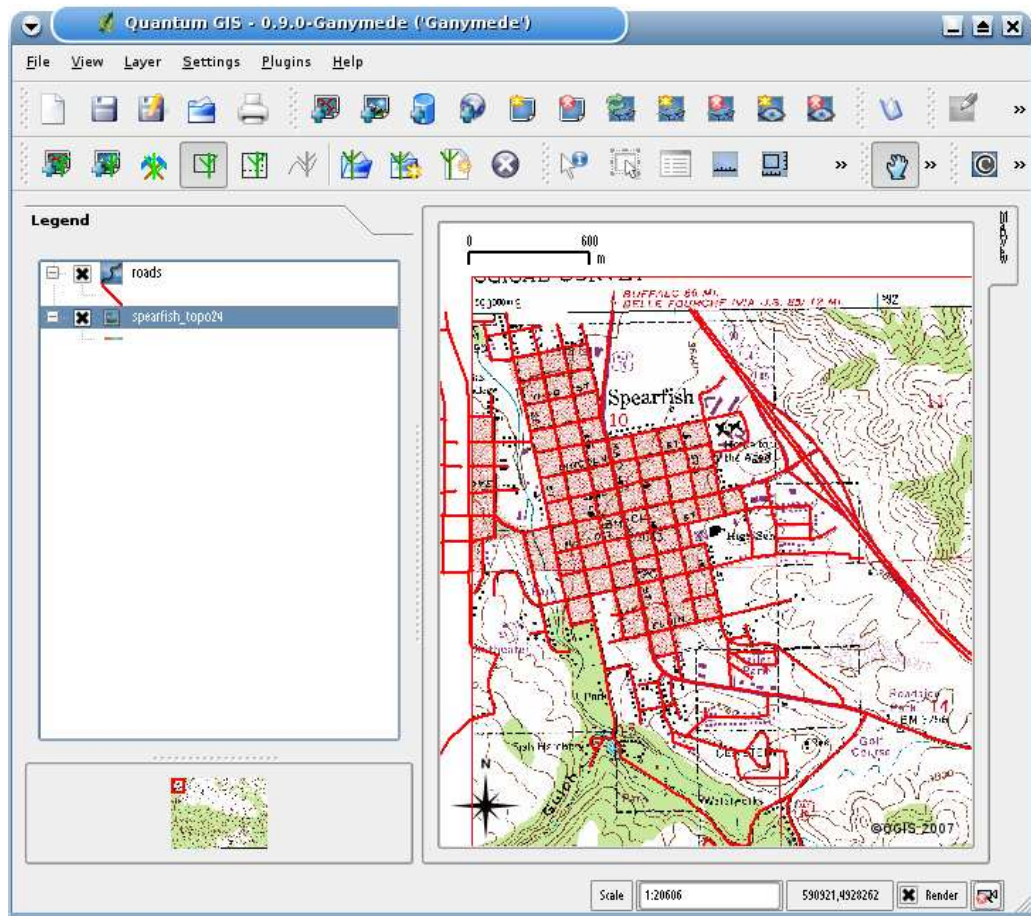
**Tip 36 WAHL DER PROJEKTION**

Die lineare (affine) Koordinaten-Transformationsmethode ist eine Transformation erster Ordnung und wird zur Skalierung, Umwandlung und Drehung von geometrisch fehlerfreien Bildern verwendet. Im Falle der Helmert-Transformation werden einem Bild - wie beim geokodieren - einfach Koordinaten-Informationen hinzugefügt. Ist ihr Bild verzerrt, so benötigen Sie Programme, die eine polynome Transformation zweiter und dritter Ordnung ermöglichen, wie z.B. GRASS GIS.

zu verbessern. Die Datei `spearfish_topo24.tif.points` zeigt beispielsweise folgende Punkte an:

mapX	mapY	pixelX	pixelY
591630.196867999969982	4927104.309682800434530	591647	4.9271e+06
608453.589164100005291	4924878.995150799863040	608458	4.92487e+06
602554.903929700027220	4915579.220743400044739	602549	4.91556e+06
591511.138448899961077	4915952.302661700174212	591563	4.91593e+06
602649.526155399973504	4919088.353569299913943	602618	4.91907e+06

Um das Rasterbild zu georeferenzieren, liegen uns 5 Koordinatenpunkte vor. Um möglichst genaue Resultate zu erhalten, ist es wichtig, die verschiedenen Punkte gleichmäßig innerhalb des Bildes zu verteilen. Zum Abschluss überprüfen wir das Ergebnis, laden die neue georeferenzierte Karte `spearfish_topo24.tif` und überlagern diese mit der Karte `roads` aus der `spearfish60` location.

**Abbildung 38:** Georeferenzierte Karte mit überlagerter Vektorkarte der spearfish location



## 11.7 Die Benutzung des Python Plugins

Das Schreiben von Plugins lässt sich mit der Programmiersprache Python wesentlich einfacher bewerkstelligen als über die Benutzung von C++. Um ein Plugin für QGIS - im folgenden als PyQGIS Plugin bezeichnet - zu erstellen, benötigen Sie QGIS 0.9, Python, PyQt und die Qt Entwickler Werkzeuge (developer tools). (2).

Wird QGIS gestartet, so überprüft das Programm auf der Suche nach C++ und den Python Plugins bestimmte Verzeichnisse. Damit eine Datei (shared library, DLL (=dynamic link library), oder Python Skript) als Plugin erkannt wird, muss diese eine bestimmte Signatur aufweisen. Im Falle von diesem Python Skript ist dies sehr einfach. QGIS überprüft hierfür im Installationsverzeichnis die folgenden Verzeichnisse:

- **Linux und andere UNIX Systeme:** `./share/qgis/python/plugins`
- **Mac OS X:** `./Contents/MacOS/share/qgis/python/plugins`
- **Windows:** `.\share\QGIS\python\plugins`

Jedes Python Plugin befindet sich dabei in einem eigenen Verzeichnis. Beim Programmstart überprüft QGIS sämtliche Unterverzeichnisse im Pfad `share/qgis/python/plugins` und initialisiert dabei sämtliche gefundenen Plugins. Nachdem dies durchgeführt wurde, erscheinen die gefundenen Plugins im QGIS-eigenen Plugin-Manager.

Lassen Sie uns nun ein Plugin erstellen, um damit eine bestehende Lücke im QGIS Interface zu schließen. Dieses Plugin wird es uns erlauben einen neuen PostGIS Layer zu erstellen und diesen in digitaler Form darzustellen. Zwar handelt es sich hierbei um ein sehr einfaches und nicht sonderlich elegantes Plugin, doch vermittelt es einen recht guten ersten Einblick darin, wie Sie in Zukunft Ihre eigenen PyQGIS Plugins erstellen können.

### 11.7.1 Die Erstellung einer Plugin-Struktur

Als erstes müssen wir uns Gedanken über den Aufbau unseres Plugins machen und dessen Struktur festlegen. In diesem Beispiel werden wir unser Plugin unter Linux erstellen, doch ist das Prinzip auf den anderen Plattformen dasselbe; es gilt lediglich einige der Systembefehle an das jeweilige System anzupassen. QGIS ist in unserem home-Verzeichnis in einem Verzeichnis mit dem Namen `qgis_09` installiert. Lassen Sie uns nun das Verzeichnis für unser Plugin erstellen.

```
mkdir ~/qgis_09/share/qgis/python/plugins/new_layer
```

Für den Anfang müssen wir im Verzeichnis `new_layer` die folgenden Dateien erstellen (Etwas später werden wir dann noch einige andere Dateien benötigen):

```
__init__.py
resources.py
resources.qrc
newlayer.py
```

### 11.7.2 Das Plugin für QGIS sichtbar machen

Das Erkennbarmachen des Plugins erfolgt mit dem `__init__.py` Skript. Im Falle unseres *NewLayer* Plugins hat das Skript folgenden Inhalt:

```
1 # load NewLayer class from file newlayer.py
2 from newlayer import NewLayer
3 def name():
4     return 'New PostGIS layer'
5 def description():
6     return 'Creates a new empty Postgis layer'
7 def version():
8     return 'Version 0.1'
9 def classFactory(iface):
10    return NewLayer(iface)
```

Diejenigen Dinge, die ein Plugin zwingend enthalten muss, sind ein Name, eine Beschreibung und eine Versionsnummer, also sämtliche Punkte die unser oben stehendes Skript beinhaltet. Jedem Schritt folgt dabei eine einfache Zeichenfolge mit den dazu passenden Informationen. Notwendig ist außerdem der sogenannte *classFactory* Schritt, der eine Verbindung zum Plugin selbst herstellen muss (Zeile 10), nachdem der **iface**-Befehl erschienen ist. Mit Hilfe dieses einfachen Programmcodes, wird QGIS unser Skript als ein Plugin erkennen.

### 11.7.3 Quellen

Um für unser Plugin nun ein schickes kleines Symbol zu erhalten, müssen wir eine Quelldatei erstellen, der wir den Namen *resources.qrc* geben. Hierbei handelt es sich lediglich um eine einfache XML-Datei, die angibt wo sich das entsprechende Symbol befindet:

```
<RCC>
  <qresource prefix='/plugins/newlayer'>
    <file>icon.png</file>
  </qresource>
</RCC>
```

Um Namenskonflikte mit anderen Plugins zu vermeiden, nutzt die Quelldatei ein Präfix. Hierbei ist die Verwendung des Pluginnamens normalerweise ausreichend. Bei der Datei *icon.png* handelt es sich um ein PNG-Bild, das in der Werkzeugleiste angezeigt wird, wenn das Plugin aktiviert ist. Für diesen Zweck können sie auch jedes Bild benutzen, sofern es eine Größe von 22x22 Pixeln besitzt und dadurch in die Werkzeugleiste passt.

Um die Quelldatei in etwas umzuwandeln, was vom Plugin genutzt werden kann, gilt es diese mit dem PyQt-Quellcompilierprogramm umzuwandeln:

```
pyrcc4 -o resources.py resources.qrc
```

Der `-o` Parameter wird zur Festlegung der Ausgabedatei verwendet. Da wir unsere Quellen erstellt haben, müssen wir einen Weg finden, um die notwendigen Information einzuholen, die für die Erstellung eines neuen Layers erforderlich sind.

#### 11.7.4 Erstellen der grafischen Benutzeroberfläche (GUI)

Normalerweise würden wir das gleiche Programm benutzen, das auch C++ Entwickler für die Erstellung einer GUI benutzen: den Qt Designer. Hierbei handelt es sich um ein visuelles Designprogramm, das es Ihnen erlaubt, Dialog- und Hauptfenster mit Hilfe von *drag & drop* Grafikobjekten zu erstellen und deren Eigenschaften festzulegen.

Für die Gestaltung unseres 'Neuer Layer' Plugins könnten wir einen großen Aufwand betreiben und Grafikobjekte für Typenfelder und andere Optionen erstellen. Da unsere Zeit hierfür jedoch begrenzt ist, werden wir uns einer anderen Möglichkeit bedienen, um die erforderlichen Informationen für das Erstellen einer Tabelle einzuholen. Dabei wird der konzeptionelle Aufbau anschaulich dargestellt und Sie können sich dann daran wagen, die Übungen des QGIS Internet Blogs zu meistern.

Um die Nutzerdaten zu erhalten, benutzen wir den *QInputDialog* Befehl aus der Qt Bibliothek. Dieser verlangt vom Nutzer die Eingabe von einer Kommandozeile. Auch wenn unser Plugin hierdurch etwas weniger elegant wird, so dient diese Methode jedoch der Veranschaulichung der einzelnen Arbeitsschritte.

Alles was wir nun noch benötigen, ist der Python Code um die Eingabedaten abzufragen und die entsprechende Tabelle zu erstellen.

#### 11.7.5 Die Erstellung des Plugins

Nun, wo wir sämtliche Vorbereitungen abgeschlossen haben, können wir mit dem Schreiben des Codes beginnen, der dann die eigentliche Arbeit verrichten soll. Beginnen wir damit, uns in der Datei *newlayer.py* einen Überblick darüber zu verschaffen, was importiert und was für das Funktionieren des Plugins bereitgestellt werden muss.

```
1 # Import the PyQt and QGIS libraries
2 from PyQt4.QtCore import *
3 from PyQt4.QtGui import *
4 from qgis.core import *
5 import pycpg
6 # Initialize Qt resources from file resources.py
7 import resources
8
9 # Our main class for the plugin
10 class NewLayer:
11
12     def __init__(self, iface):
13         # Save reference to the QGIS interface
14         self.iface = iface
15
16     def initGui(self):
17         # Create action that will start plugin configuration
18         self.action = QAction(QIcon('/:plugins/newlayer/icon.png'), \
19                               'New PosGIS Layer', self.iface.getMainWindow())
20         QObject.connect(self.action, SIGNAL('activated()'), self.run)
21
22         # Add toolbar button and menu item
23         self.iface.addToolBarIcon(self.action)
24         self.iface.addPluginMenu('&New PostGIS Layer...', self.action)
25
26     def unload(self):
27         # Remove the plugin menu item and icon
28         self.iface.removePluginMenu('&New PostGIS Layer...', self.action)
29         self.iface.removeToolBarIcon(self.action)
```

In den Zeilen 2 bis 7 importieren wir die Bibliotheken, die wir für das Plugin benötigen. Hierzu gehören die PyQt Bibliotheken, die QGIS Kernbibliothek und die Python PostgreSQL Bibliothek `psycpg`. Bei jedem Python Skript, das die QGIS Bibliotheken und PyQt benutzt, müssen die QtCore und die QtGui Bibliotheken, sowie die QGIS Kernbibliothek importiert werden. Auf diese Weise erhalten wir Zugang zu den sogenannten PyQt-Wrappern für unsere Qt-Objekte (wie etwa unseren Eingabedialog) sowie auf die QGIS Kernbibliotheken. Zudem müssen wir die zuvor erstellte Datei `resources.py` importieren, welche die genaue Symbol/Icon-Beschreibung enthält.

In Zeile 10 legen wir die Kategorie **NewLayer** fest. Mit dem `__init__` Befehl (Zeile 12 bis 14) wird unsere Kategorie festgelegt und mittels des ClassFactory Befehls in Zeile 10 von `__init__.py` an das `iface`-Objekt von QGIS angepasst. Um **iface** auch später nutzen zu können, speichern wir dieses als sogenannte 'Membervariable' ab.



In den Zeilen 16 bis 24 legen wir für das Plugin die Bestandteile der grafischen Benutzeroberfläche (GUI) fest. In Qt wird **QAction** benutzt, um eine Funktion für die Benutzeroberflächen zu erstellen, die sowohl für das Erschaffen eines Menü- als auch eines Werkzeugleisten-Elements verwendet werden kann. Im Falle unseres Plugins nutzen wir diese Funktion für beides. In Zeile 18 erstellen wir eine Funktion, die auf unsere Symbol/Icon-Quelle zurückgreift (achten Sie dabei auf das Präfix, das wir in der Datei *resources.qrc* angegeben haben). Desweiteren geben wir auch etwas Text an, der in einem Menüeintrag oder bei einem Mouseover erscheint. Abschließend gilt es noch das sogenannte 'parent' anzugeben. Im Falle eines Plugins ist dieses 'parent' das Hauptfenster von QGIS. Das **iface** Objekt, das wir während der Initialisierung gespeichert haben, ermöglicht es uns in Zeile 19 die Quellenangabe für das Hauptfensters zu erhalten.

Nachdem dieser erste Teil des Plugins nun erstellt ist, können wir diesen der Werkzeugleiste und dem *Plugins* Menü (Zeile 23 und 24) hinzufügen. Damit wird die grafische Benutzeroberfläche für das Plugin vorbereitet. Als weiteren Schritt müssen wir 'hinter uns aufräumen', wobei unser Plugin dabei nicht geladen sein darf. Um dieses Aufräumen kümmert sich der *unload* Befehl, der den Menüpunkt und das Werkzeug aus der Werkzeugleiste entfernt (Zeile 28 und 29).

Dieser erste Teil sorgt also nun für die Initialisierung und ein sauberes Starten und Beenden unseres Plugins. Lassen Sie uns nun einen Blick auf den Programmcode werfen, der die eigentliche Aufgabe verrichtet. Alle Informationen hierzu sind im *run* Programmabschnitt enthalten.

```
30 def run(self):
31     # Get the user input, starting with the table name
32     table_name = QInputDialog.getText(None, 'Table Name?', \
33         'Name for new PostGIS layer')
34     if table_name[0].length() > 0:
35         # Get the field names and types
36         fields = QInputDialog.getText(None, 'Field Names', \
37             'Fields (separate with a comma)')
38         parts = fields[0].split(',')
39         # Create the SQL statement
40         sql = 'create table ' + table_name[0] + ' (id int4 primary key, '
41         for fld in parts:
42             sql += fld + ' varchar(10), '
43         sql = sql[0:-2]
44         sql += ' )'
45         # Connect to the database
46         # First get the DSN
47         dsn = QInputDialog.getText(None, 'Database DSN', \
48             'Enter the DSN for connecting to the database (dbname=db user=user)')
49         if dsn[0].length() > 0:
50             con = psycopg.connect(str(dsn[0]))
```

```

51     curs = con.cursor()
52     curs.execute(str(sql))
53     con.commit()
54     # add the geometry column
55     curs.execute('select AddGeometryColumn('' + str(table_name[0]) + \'
56         \', \'the_geom\', 4326, \'POLYGON\', 2)\')
57     con.commit()
58     # create the GIST index
59     curs.execute('create index idx_' + str(table_name[0]) + ' on ' + \'
60         str(table_name[0]) + ' USING GIST(the_geom GIST_GEOMETRY_OPS)\')
61     con.commit()

```

Als erstes müssen wir den **QInputDialog** benutzen um den Namen der zu erstellenden Tabelle zu erhalten. Die Abfrage des Namens erfolgt in Zeile 32.

**Abbildung 39:** Geben Sie einen neuen PostGIS Tabellennamen ein



Bevor wir fortfahren überprüfen wir in Zeile 34 ob der Anwender auch tatsächlich etwas eingegeben hat.

Als nächstes müssen wir die Feldnamen abfragen. In unserem Beispiel gestalten wir diesen Punkt sehr einfach. Jedes Feld wird als varchar(10) definiert, d. h. dass jedes Feld bis zu 10 Zeichen enthalten kann. Um das Plugin voll nutzbar zu machen, müssten wir dem Anwender allerdings eine Möglichkeit bieten, die Art des Feldtyps bestimmen zu können. In Zeile 36 fordern wir den Nutzer zur Eingabe einer kommagetrennten Feldnamensliste auf.

Dann teilen wir diese Liste in ihre Bestandteile auf, um daraus die sogenannten SQL Statements zu erstellen (Zeile 38).

Zeile 40 enthält den ersten Teil des SQL Statements. Beachten Sie, dass wir die Tabelle mit einem Integer-ID Feld erstellen, das als Primärschlüssel fungiert. Dann wiederholen wir diesen Schritt für die Feldliste und hängen den entsprechenden Code an das SQL Statement (Zeile 41).

Nachdem wir sämtliche Felder zum SQL Statement hinzugefügt haben, trennen wir die ungewünschten Zeichen ab (Zeile 43) und fügen dann eine geschlossene runde Klammer ein, um damit das

**Abbildung 40:** Geben sie die Feldnamen für die neue PostGIS Tabelle ein

Statement abzuschließen (Zeile 44).

Nun sind wir soweit, um uns mit der Datenbank zu verbinden und die Tabelle zu erstellen. Um Zugang zur Datenbank zu erhalten benutzen wir `psycopg2` (<http://www.initd.org>). Für den Verbindungsaufbau müssen wir den Namen der Datenquelle (DSN) mit dem Namen der Datenbank, den Nutzer und ein eventuell notwendiges Passwort angeben. Wenn wir sowohl QGIS als auch PostgreSQL auf demselben Rechner laufen haben müssen wir normalerweise kein Passwort festlegen. In solch einem Falle sieht der DSN in etwa folgendermaßen aus:

*dbname=gis\_data user=gsherman*

Um den DSN zu erhalten, fragen wir diesen in Zeile 47 vom Anwender mit einem **QInputDialog** ab.

**Abbildung 41:** Geben Sie den DSN für die Verbindung mit der PostGIS Datenbank ein

Wenn der Nutzer einen DSN eingibt, können wir in Zeile 50 mit der Verbindung zur Datenbank fortfahren. In Zeile 51 erhalten wir von dieser Verbindung einen Cursor und führen dann das SQL Statement für die Tabellenerstellung aus und geben diese Änderung in den Zeilen 52 bis 53 an. Auf diese Weise wird zwar die Tabelle erstellt, damit diese jedoch auch als ein gültiger und nutzbarer Layer funktioniert, bedarf es aber noch einiger weiterer Dinge.

Als erstes ist eine Geometriespalte erforderlich. Ganz bewusst haben wir eine solche Spalte bei der Anlage der Tabelle noch nicht eingefügt, um diese nun mit der Funktion *AddGeometryColumn*

erstellen zu können. Diese Funktion fügt der Tabelle eine Geomtriespalte hinzu und legt für uns in der *geometry\_columns* Tabelle einen Eintrag an. In Zeile 55 legen wir den Tabellennamen, den Namen der Geomeriespalte, die SRID, den Feature-Typ und dessen Ausdehnung fest.

Als letztes müssen wir der Tabelle nun einen so genannten 'spatial index' zufügen, damit bei der Abfrage und Darstellung räumlicher Daten in QGIS eine möglichst hohe Geschwindigkeit erzielt werden kann. In Zeile 59 haben wir das SQL so 'zusammengebastelt', dass dieser Index erstellt wird. Die eigentliche Anweisung sieht dann folgendermaßen aus:

```
create index sidx_park_land on park_land
  USING GIST(the_geom GIST_GEOMETRY_OPS);
```

### 11.7.6 Fragen und Probleme

Unser Plugin ist nun fertig. Lassen Sie uns nun einen Blick darauf werfen, was daran noch falsch ist und verbessert und optimiert werden kann:

- Wir könnten ein verbessertes GUI nutzen, welches es dem Nutzer erlauben würde sämtliche erforderlichen Informationen in einem einzigen Programmdialog einzugeben.
- Der Nutzer kann keine Feldtypen festlegen.
- Die Fehlerprüfung im Dialogfenster ist begrenzt.
  - Das Plugin funktioniert nur, wenn sämtliche Felder eingegeben werden.
  - Es findet keinerlei Fehlerprüfung der Datenbankoperationen statt.
- Man erhält keine Rückmeldung wenn das Plugin die Arbeitsprozedur abgeschlossen hat.

Trotz all dieser Mängel dient es jedoch als ein ganz ursprüngliches Plugin, anhand dessen der konzeptionelle Ablauf einer Pluginerstellung recht passend dargestellt wird und das Ihnen bei den ersten Schritten ihrer eigenen Plugin Entwicklungen weiterhilft.

### 11.7.7 Hinzufügen einer Rückmeldung

Lassen Sie uns eines dieser kleinen Probleme dadurch lösen, indem wir an das Ende des Arbeitsvorgangs einige Rückmeldungen hinzufügen. Hierfür werden wir ganz einfach ein Nachrichtenfenster einfügen, welches den Nutzer darüber aufklärt, dass alle Prozesse abgeschlossen sind und zu prüfen ist, ob die entsprechende Tabelle in der Datenbank erstellt wurde.

Um dies zu bewerkstelligen geben wir nach der Zeile 61 einfach den folgenden Programmcode ein:

```
# show the user what happened
```

```
QMessageBox.information(None, 'Results', 'Table ' + str(table_name[0]) + \
' has been created. Check your database to confirm.')
```

Sobald die Tabelle erstellt worden ist, erhält der Nutzer dann die folgende Meldung:

**Abbildung 42:** Nachrichtenfenster mit den Plugin-Ergebnissen



### 11.7.8 Zusammenfassung

Das Schreiben eines QGIS Plugins in Python ist relativ einfach. Einige Plugins erfordern nicht einmal eine Grafische Benutzeroberfläche (GUI). Sie könnten beispielsweise ein Plugin schreiben das die Koordinaten eines Punktes wiedergibt auf den sie in der Karte klicken. Eine solches Plugin würde keinerlei Nutzereingabe erfordern und es könnte für die Darstellung der Ergebnisse auf eine bereits vorhandene Qt **QMessageBox** zurückgegriffen werden.

Sie können Plugins für QGIS auch in C++ schreiben, dies ist allerdings ein anderes Thema. Tutorials darüber, wie man QGIS Plugins sowohl in C++ als auch in Python schreibt, finden sie im QGIS-Blog auf:

<http://blog.qgis.org>

---

## 12 Weitere Anwendungen mit den QGIS-Bibliotheken erstellen

Ein Ziel von QGIS ist es, neben der eigentlichen Applikation, auch einen Satz von Bibliotheken zur Verfügung zu stellen, mit denen man weitere Anwendungen erstellen kann. Dieses Ziel wurde mit dem Refaktorisieren der Bibliotheken nach der Herausgabe der Version 0.8.0 realisiert. Ab der Version 0.9.0 ist nun die Entwicklung von C++ oder Python-Anwendungen auf Basis der QGIS-Bibliotheken möglich.

In diesem Kapitel werden wir einen kurzen Blick auf diese Funktionalitäten werfen und eine Standalone-Anwendung mit Python erstellen. Der QGIS-Blog stellt weitere Beispiele zum Erstellen von PyQGIS<sup>7</sup>-Anwendungen bereit.

Wir werden eine dieser Beispielanwendungen als Einstiegspunkt benutzen, um zu sehen, wie man diese Anwendungen erstellt.

Die Funktionalitäten unserer Anwendung sind folgende:

- eine Vektordatei laden
- Verschieben
- Herein- und herauszoomen
- Auf volle Ausdehnung des Themas zoomen
- eigene Farben auswählen, wenn ein Thema geladen ist

Das sind natürlich sehr wenige Funktionen. Lassen Sie uns nun mit der Erstellung der graphischen Oberfläche beginnen; dazu benutzen wir den QT Designer.

### 12.1 Erstellung der graphischen Oberfläche

Wenn wir schon nur minimalistische Funktionen einbauen, machen wir auch nur eine minimalistische Oberfläche. Mit dem QT Designer erstellen wir nun ein einfaches MainWindow ohne Menüs und Werkzeugleisten. Das erzeugt ein leeres Fenster. Zur Erstellung sind die folgenden Punkte nötig:

1. Erstellen Sie einen leeren Ordner, in dem die Anwendung entwickelt wird und wechseln Sie in diesen Ordner.
2. Starten Sie den Qt Designer
3. Der Dialog 'Neues Formular' sollte erscheinen. Falls nicht, wählen Sie *Neues Formular...* aus dem *Datei* Menü.
4. Wählen Sie 'Main Window' aus der Vorlagen/Formularliste
5. Klicken Sie *Neu von Vorlage*

---

<sup>7</sup>Eine Anwendung, die Python und QGIS benutzt.

6. Ziehen Sie das neue Fenster auf eine nutzbare Fenstergröße
7. Finden Sie das Frame widget in der Liste (unter Container) und ziehen Sie es in das Fenster 'Main Window' das sie gerade erstellt haben.
8. Klicken Sie außerhalb des Rahmens um das Hauptfenster zu selektieren.
9. Klicken Sie auf das Werkzeug *Objekte tabellarisch anordnen*. Danach sollte der Rahmen das gesamte Hauptfenster füllen.
10. Speicher Sie das Formular als *mainwindow.ui*
11. Beenden Sie Qt Designer

Nun können Sie das Formular mit dem PyQt-Interface kompilieren:

```
pyuic4 -o mainwindow_ui.py mainwindow.ui
```

Dieser Befahler erzeugt den Python Quellcode für das Hauptfenster. Als nächstes müssen wir den Anwendungscode erstellen, der das leere Fenster mit einigen Werkzeugen füllt.

## 12.2 Erzeugen des Hauptfensters (MainWindow)

Nun sind wir soweit, um die **MainWindow**-Klasse zu schreiben, die die richtige Arbeit macht. Da die Anwendung einige Zeilen lang ist, werden wir Diesen in Teilen anschauen. Wir starten mit der Import-Sektion und den Umgebungseinstellungen:

```
1 # Loosely based on:
2 #   Original C++ Tutorial 2 by Tim Sutton
3 #   ported to Python by Martin Dobias
4 #   with enhancements by Gary Sherman for FOSS4G2007
5 # Licensed under the terms of GNU GPL 2
6
7 from PyQt4.QtCore import *
8 from PyQt4.QtGui import *
9 from qgis.core import *
10 from qgis.gui import *
11 import sys
12 import os
13 # Unsere GUI importieren
14 from mainwindow_ui import Ui_MainWindow
15 # Unsere Icons und Ressourcen importieren (icons)
16 import resources
17
```

```

18 # Umgebungsvariable QGISHOME muss auf das Installationsverzeichnis von 0.9 zeigen
19 # bevor die Anwendung installiert werden kann.
20 qgis_prefix = os.getenv('QGISHOME')

```

Einiges sieht sehr ähnlich dem der Plugins aus, insbesondere die PyQt4 und QGIS-Importe. Einige Besonderheiten sind der GUI-Import in Zeile 14 und der Import unserer Ressourcen in Zeile 16.

Unsere Anwendung muss wissen, wo es die QGIS-Installation finden kann. Deswegen setzen wir die Umgebungsvariable QGISHOME, die auf das Installationsverzeichnis von QGIS 0.9.0 zeigt. In Zeile 20 weisen wir diesen Wert einer Variable zu, damit wir sie später weiterbenutzen können.

Als nächstes erstellen wir unsere **MainWindow**-Klasse, die alle Logik unserer Anwendung enthält.

```

21 class MainWindow(QMainWindow, Ui_MainWindow):
22
23     def __init__(self):
24         QMainWindow.__init__(self)
25
26         # von Qt4 benötigt, um das UI zu initialisieren
27         self.setupUi(self)
28
29         # Title setzen
30         self.setWindowTitle('FOSS4G2007 Demo App')
31
32         # Kartenfenster erzeugen
33         self.canvas = QgsMapCanvas()
34         # Hintergrundfarbe auf hellblau setzen
35         self.canvas.setCanvasColor(QColor(200,200,255))
36         self.canvas.enableAntiAliasing(True)
37         self.canvas.useQImageToRender(False)
38         self.canvas.show()
39
40         # Widget im MainWindows mit einer vertikalen Box ausrichten.
41         #
42         self.layout = QVBoxLayout(self.frame)
43         self.layout.addWidget(self.canvas)
44
45         # Actions erzeugen und jedes Werkzeug mit der entsprechenden Methode
46         # verbinden
47         self.actionAddLayer = QAction(QIcon('/:foss4g2007/mActionAddLayer.png'),
48         \
49             'Add Layer', self.frame)
50         self.connect(self.actionAddLayer, SIGNAL('activated()'), self.addLayer)

```



```

51     self.actionZoomIn = QAction(QIcon('/:foss4g2007/mActionZoomIn.png'), \
52         'Zoom In', self.frame)
53     self.connect(self.actionZoomIn, SIGNAL('activated()'), self.zoomIn)
54     self.actionZoomOut = QAction(QIcon('/:foss4g2007/mActionZoomOut.png'), \
55         'Zoom Out', self.frame)
56     self.connect(self.actionZoomOut, SIGNAL('activated()'), self.zoomOut)
57     self.actionPan = QAction(QIcon('/:foss4g2007/mActionPan.png'), \
58         'Pan', self.frame)
59     self.connect(self.actionPan, SIGNAL('activated()'), self.pan)
60     self.actionZoomFull = QAction(QIcon('/:foss4g2007/mActionZoomFullExtent.png'), \
61         'Zoom Full Extent', self.frame)
62     self.connect(self.actionZoomFull, SIGNAL('activated()'),
63         self.zoomFull)
64
65     # Werkzeugleiste erzeugen
66     self.toolbar = self.addToolBar('Map')
67     # Add the actions to the toolbar
68     self.toolbar.addAction(self.actionAddLayer)
69     self.toolbar.addAction(self.actionZoomIn)
70     self.toolbar.addAction(self.actionZoomOut);
71     self.toolbar.addAction(self.actionPan);
72     self.toolbar.addAction(self.actionZoomFull);
73
74     # Kartenwerkzeuge erzeugen
75     self.toolPan = QgsMapToolPan(self.canvas)
76     self.toolZoomIn = QgsMapToolZoom(self.canvas, False) # false = in
77     self.toolZoomOut = QgsMapToolZoom(self.canvas, True) # true = out

```

Die Zeilen 21 bis 27 sind grundsätzliche Deklarationen und Initialisierungen des **MainWindow** und die Einstellungen des UserInterface mit der Methode *setUpUi*. Das ist für alle Anwendungen nötig.

Als nächstes setzen wir den Titel für die Anwendung, sodass es etwas Interessanteres als 'MainWin-dow' (Zeile 30) anzeigt. Wenn das getan ist, können wir das Benutzerinterface weiter komplettieren. Als wir es im Designer erstellt haben, haben wir es sehr sparsam gehalten — lediglich ein Hauptfen-ster und einen Rahmen. Wir hätten bereits im Designer ein Menü und eine Werkzeugleiste erstellen können, wir machen es aber nun direkt in Python.

In Zeile 33 wir setzen wir das Kartenfenster, die Hintergrundfarbe auf Hellblau und schalten das Antialiasing ein. Außerdem nutzen wir nicht QImage zum Rendern (trauen Sie mir an diesem Punkt!). Um das Kartenfenster sichtbar zu schalten, nutzen wir die *show*-Methode.

Das Thema soll in einer vertikalen Box innerhalb des Rahmens ausgerichtet werden; in Zeile 43 wird das Kartenbild hinzugefügt.

In den Zeilen 48 bis 63 werden die Actions und Verbindungen für die Werkzeuge in der Werkzeugleiste verdrahtet. Für jedes Werkzeug erzeugen wir eine **QAction** mit einem in unserer Ressourcen-Datei definierten Icon. Dann verbinden wir das *Aktivierungssignal* vom Werkzeug zur Methode in unserer Klasse, welche die Action ausführen wird. Das ist identisch mit dem Beispiel für Plugins.

Wenn wir die Aktionen und Verbindungen verbunden haben, müssen wir sie zu der Werkzeugleiste hinzufügen. In den Zeilen 66 bis 72 tun wir dies.

Als letztes erzeugen wir 3 Kartenwerkzeuge für die Anwendung (Zeilen 75-77). Wir benutzen die Kartenwerkzeuge in dem Moment, wenn wir die Methoden definieren, die unsere Anwendung funktional machen. Lassen Sie uns auf die Methoden für die Kartenwerkzeuge schauen.

```

78  # Kartenwerkzeug auf Hereinzoomen stellen
79  def zoomIn(self):
80      self.canvas.setMapTool(self.toolZoomIn)
81
82  # Kartenwerkzeug auf Herauszoomen stellen
83  def zoomOut(self):
84      self.canvas.setMapTool(self.toolZoomOut)
85
86  # Kartenwerkzeug auf Pannen stellen
87  def pan(self):
88      self.canvas.setMapTool(self.toolPan)
89
90  # Kartenwerkzeug auf volle Ausdehnung stellen
91  def zoomFull(self):
92      self.canvas.zoomFullExtent()
```

Für jedes Kartenwerkzeug müssen wir eine Methode erstellen, die mit der Verbindung zu den Actions korrespondiert. In den Zeilen 79-88 definieren wir je eine Methode für die 3 Werkzeuge, die mit der Karte interagieren. Wenn ein Werkzeug durch Klicken in der Werkzeugleiste aktiviert wird, wird die korrespondierende Methode aufgerufen, die dem Kartenbild “mitteilt”, dass es aktiv ist. Das aktive Werkzeug entscheidet, was passieren soll, wenn auf der Karte geklickt wird.

Der Zoom auf die volle Ausdehnung ist nicht wirklich ein Kartenwerkzeug—es macht seinen Job auch ohne einen Klick auf die Karte. Wenn es aktiviert wird, rufen wir die Methode *zoomFullExtent* des Kartenfenster auf (Zeile 92). Dies komplettiert die Implementierung von allen Kartenwerkzeugen—außer dem zum Hinzufügen von Themen. Lassen Sie uns dieser Funktionalität als nächstes widmen:

```

93  # Ein OGR-Thema hinzufügen
94  def addLayer(self):
95      file = QFileDialog.getOpenFileName(self, 'Open Shapefile', '.', 'Shapefiles
```

```

96     (*.shp)')
97     fileInfo = QFileInfo(file)
98
99     # Layer hinzufügen
100     layer = QgsVectorLayer(file, fileInfo.fileName(), 'ogr')
101
102     if not layer.isValid():
103         return
104
105     # Farbe des Themas auf grau
106     symbols = layer.renderer().symbols()
107     symbol = symbols[0]
108     symbol.setFillColor(QColor.fromRgb(192,192,192))
109
110     # Thema zur Kartenregistry hinzufügen
111     QgsMapLayerRegistry.instance().addMapLayer(layer);
112
113     # Ausdehnung auf die Themenausdehnung setzen
114     self.canvas.setExtent(layer.extent())
115
116     # Kartenanwendungslayer setzen
117     cl = QgsMapCanvasLayer(layer)
118     layers = [cl]
119     self.canvas.setLayerSet(layers)

```

In der `addLayer` Methode benutzen wir den **QFileDialog** um den Namen des Shapefiles zu bekommen, das wir laden wollen. Das geschieht in Zeile 96. Beachten Sie, dass wir einen "Filter" nutzen, damit im Dialog nur die Dateien mit der Endung `.shp` angezeigt werden.

In Zeile 97 erzeugen wir ein **QFileInfo** Objekt aus dem Shapefile-Pfad. Nun ist der Layer fertig zum Erzeugen (Zeile 100). Wie benutzen das **QFileInfo** Objekt um den Dateinamen aus dem Pfad zu extrahieren und setzen es als Name für den neu erzeugten Layer ein. Um sicherzugehen, dass der Layer auch gültig ist und keine Probleme beim Laden macht, überprüfen wir dies in Zeile 102. Falls er fehlerhaft ist, wird abgebrochen und der als fehlerhaft erkannte Layer wird nicht zum Kartenbild hinzugefügt.

Normalerweise werden Kartenlayer mit einer zufälligen Farbe hinzugefügt. Hier wollen wir die Farben der Layer verändern, um eine schönere Darstellung zu erhalten. Außerdem wissen wir, dass wir den Layer `world_borders` zur Karte hinzufügen wollen; das sieht schöner auf unserem blauen Hintergrund aus. Um die Farbe zu ändern, müssen wir das Symbol, welches zum Zeichnen verwendet wird, holen und eine neue Füllfarbe setzen. Das alles wird in den Zeilen 106-108 gemacht.

All diese Dinge sind nötig, um einen neuen Layer zur Registrierung hinzuzufügen. Außerdem sind einige andere wichtige “Haushaltsdinge” erledigt worden (Zeilen 111-119). Dies ist der normale Weg um ein Thema hinzuzufügen. Das Ergebnis sind die Grenzen der Welt auf einem hellblauen Hintergrund. Das einzige, was evtl. nicht durchgeführt werden muss, ist das Setzen der Ausdehnung auf den Layer; insbesondere dann, wenn Sie noch mehrere Themen zu Ihrer Anwendung hinzufügen wollen.

Das ist das Herz der Anwendung und komplettiert gleichzeitig ihre **MainWindow**-Klasse.

## 12.3 Das Finale!

Der unten stehende Teil des Codes erzeugt das Objekt **QgsApplication** object, setzt die Pfade zur QGIS-Installation, definiert die Methode *main* und startet die Anwendung. Das einzig erwähnenswerte ist noch die Tatsache, dass wir das Anwendungsfenster zur oberen linken Ecke des Displays verschieben. Wir könnten auch die Qt API nutzen und das Fenster zentrieren...

```

120 def main(argv):
121     # erzeuge die Qt-Anwendung
122     app = QApplication(argv)
123
124     # QGIS-Bibliotheken initialisieren
125     QgsApplication.setPrefixPath(qgis_prefix, True)
126     QgsApplication.initQgis()
127
128     # Hauptfenster erstellen
129     wnd = MainWindow()
130     # Fenster zur oberen linken Ecke bewegen
131     wnd.move(100,100)
132     wnd.show()
133
134     # und los!
135     retval = app.exec_()
136
137     # beenden
138     QgsApplication.exitQgis()
139     sys.exit(retval)
140
141
142 if __name__ == '__main__':
143     main(sys.argv)

```

## 12.4 Anwendung ausführen

Nun sollten wir die Anwendung starten und sehen, was passiert. Wahrscheinlich haben Sie es, wie fast jeder Entwickler, bereits unterwegs ausprobiert.

Bevor wir die Anwendung ausführen können, müssen wir einige Umgebungsvariablen setzen. Unter GNU/Linux oder Mac OS X geht das so:

```
export LD_LIBRARY_PATH=$HOME/qgis_09/lib
export PYTHONPATH=$HOME/qgis_09/share/qgis/python
export QGISHOME=$HOME/qgis_09
```

Für Windows:

```
set PATH=C:\qgis;%PATH%
set PYTHONPATH=C:\qgis\python
set QGISHOME=C:\qgis
```

Im Falle von GNU/Linux oder Mac OS X haben wir angenommen, dass QGIS in Ihrem Heimatverzeichnis unter *qgis\_09* installiert wurde. Für Windows wurde der Installationspfad *C:\qgis* für QGIS angenommen.

Wenn nun die Anwendung startet, sieht es wie folgt aus:

**Abbildung 43:** Starten der neuen Demoanwendung



Um den Layer *world\_borders* hinzuzufügen, klicken Sie auf das Werkzeug *Thema hinzufügen* und navigieren zum Verzeichnis mit den Daten. Wählen sie das Shapefile aus und klicken Sie auf *Öffnen*, um es der Karte hinzuzufügen. Unsere eigene Farbe wird nun hinzugefügt und das Ergebnis sollte so aussehen:

Das Erstellen einer PyQGIS-Anwendung ist sehr einfach, wie sie sehen. In weniger als 150 Zeilen Code haben wir eine Anwendung erstellt, die Shapefiles laden kann und mit der man auf der Karte navigieren kann. Wenn Sie ein wenig mit der Karte herumspielen, werden Sie feststellen, dass noch

**Abbildung 44:** Hinzufügen eines Themas zur Demo-Anwendung

weitere eingebaute Funktionen funktionieren, wie beispielsweise das Mausrad-Zoomen und das Verschieben der Karte unter Zuhilfenahme der *Leertaste* bei gleichzeitigem Bewegen der Maus.

Einige komplexe Anwendungen wurden bereits mit PyQGIS erzeugt; es werden stetig mehr. Das ist sehr beachtlich, denn einige dieser Anwendungen wurden bereits vor dem offiziellen Release der QGIS-Version 0.9.0 veröffentlicht.

---

**Tip 37** DOKUMENTATION VON PYQGIS

---

Falls Sie ein Plugin oder eine PyQGIS-Anwendung schreiben wollen, werden Sie nicht umherkommen, die Dokumentation der QGIS API (<http://qgis.org>) und der PyQt Python Bindungs (<http://www.riverbankcomputing.com/Docs/PyQt4/pyqt4ref.html>) anzuschauen. Beide Dokumentationen stellen Informationen über Klassen und Methoden zur Verfügung, die Sie benötigen, um Ihre Python-Anwendung zum Leben zu erwecken.

---

---

## 13 Hilfe und Support

### 13.1 Mailinglisten

QGIS entwickelt sich ständig weiter, daher kann es vorkommen, dass es mal nicht so funktioniert, wie erwartet. Die bevorzugte und effektivste Art, Hilfe zu bekommen, besteht darin, sich in die qgis-users Mailingliste einzuschreiben.

#### **qgis-users**

Ihre Fragen erreichen eine breite Basis von Anwendern und die Antworten auf Ihre Fragen können auch anderen helfen. Sie können sich in die Mailingliste eintragen unter der URL:

<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-user>

#### **qgis-developer**

Wenn Sie ein Entwickler sind und technische Probleme haben, können Sie sich in die qgis-developer Mailingliste eintragen unter der URL:

<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-developer>

#### **qgis-commit**

Jedes Mal wenn ein Commit in den QGIS Quellcode erfolgt ist, wird eine Email an folgende Liste geschickt. Wenn Sie alle Veränderungen im Quellcode verfolgen möchten, können Sie sich unter folgender URL eintragen:

<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-commit>

#### **qgis-trac**

Diese Mailingliste stellt Nachrichten in Bezug auf das Projekt Management bereit. Dazu gehören Fehlerberichte, Aufgaben und Anfragen für neue Funktionen. Sie können sich für diese Mailingliste eintragen unter der URL:

<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-trac>

#### **qgis-doc**

Diese Liste beschäftigt sich mit Themen wie der Dokumentation, der inhaltlichen Hilfe von QGIS, dem Benutzer- und Installationshandbuch sowie Übersetzungen in verschiedene Sprachen. Wenn Sie mit an dem nächsten Benutzer- und Installationshandbuch arbeiten möchten oder es übersetzen wollen, können Sie sich für diese Mailingliste eintragen unter der URL:

<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-doc>

### qgis-psc

Diese Liste befasst sich mit den Aufgaben des Lenkungsausschusses des QGIS-Projektes. Sie können sich für diese Mailingliste eintragen unter der URL:

<http://mrcc.com/cgi-bin/mailman/listinfo/qgis-psc>

Wir heißen Sie herzlich Willkommen, sich auf jeder dieser Listen einzuschreiben und den anderen QGIS Benutzern und Entwicklern mit ihrer Erfahrung zu helfen. Beachten Sie bitte auch, dass die Mailinglisten qgis-commit und qgis-trac nur dazu erstellt wurden, um Benachrichtigungen zu verteilen und nicht für Anwenderfragen geeignet ist.

## 13.2 IRC

Wir sind außerdem im IRC präsent - Sie können uns im #qgis Kanal unter [irc.freenode.net](http://irc.freenode.net) treffen. Bitte warten Sie ein wenig auf Antworten, da die meisten nur zwischendurch mal vorbeischauchen, was gerade so passiert. Kommerzieller Support ist auch möglich. Schauen Sie dazu auf die Internetseite <http://qgis.org/content/view/90/91>.

Wenn Sie eine Diskussion im IRC verpasst haben - kein Problem! Wir loggen alle Diskussionen, damit Sie diese auch später lesen können. Lesen Sie die Logs unter der URL: <http://logs.qgis.org>.

## 13.3 BugTracker

Während die qgis-users Mailingliste nützlich ist, wenn es um allgemeine Fragen zu 'wie mache ich dies oder jenes in QGIS' geht, möchten Sie uns vielleicht auch auf richtige Fehler (Bugs) aufmerksam machen. Sie können dazu Fehlermeldungen mit Hilfe des QGIS BugTracker unter <http://svn.qgis.org/trac> erstellen. Wenn Sie ein neues Ticket für einen Fehler erstellen, geben Sie bitte auch eine Emailadresse an, über die wir weitere Informationen von Ihnen erfragen können.

Denken Sie auch bitte daran, dass ein für Sie wichtiger Fehler nicht immer die gleiche Priorität bei anderen Personen und besonders den Entwicklern hat. Einige Fehler sind sehr aufwendig zu reparieren und daher kann es schon mal ein wenig dauern, bis genügend Zeit vorhanden ist, ein Problem zu lösen.

Anfragen für neue Funktionen können auch in demselben System gestellt werden. Bitte geben Sie dann den Typ **enhancement** an.

Wenn Sie einen Fehler gefunden und selbst repariert haben, können Sie es auch als Patch schicken. Der BugTracker bietet auch hierfür einen Typ **patch**. Die Entwickler schauen sich den Patch dann an und fügen ihn zum Quellcode hinzu. Bitte haben Sie ein wenig Geduld, wenn dies ein wenig dauert.



## 13.4 Blog

Die QGIS-Gemeinschaft stellt auch einen Weblog (BLOG) unter <http://blog.qgis.org> bereit, mit vielen interessanten Artikeln für Anwender und Entwickler zum Thema QGIS. Sie sind herzlich eingeladen, selbst einen Artikel zu verfassen, nachdem Sie sich auf der Seite registriert haben!

## 13.5 Wiki

Schließlich gibt es auch ein QGIS WIKI unter <http://wiki.qgis.org>, wo Sie eine Vielzahl nützlicher Informationen über die QGIS-Entwicklung, Pläne für neue Versionen, Links zum Herunterladen von Daten oder zu vorhandenen Übersetzungen finden. Schauen Sie mal rein, da gibt es ein paar wirkliche Attraktionen!

---

## A Unterstützte Datenformate

### A.1 Unterstützte OGR-Formate

Zum Zeitpunkt der Erstellung dieses Dokuments werden folgende Datenformate durch die OGR-Bibliothek unterstützt. Formate, die unter QGIS getestet wurden, sind **fett** dargestellt.

- **Arc/Info Binary Coverage**
- Comma Separated Value (.csv)
- DODS/OPeNDAP
- **ESRI Shapefile**
- FMEObjects Gateway
- GML
- IHO S-57 (ENC)
- **Mapinfo File**
- Microstation DGN
- OGDl Vectors
- ODBC
- Oracle Spatial
- PostgreSQL<sup>8</sup>
- **SDTS**
- SQLite
- UK .NTF
- U.S. Census TIGER/Line
- VRT - Virtual Datasource

### A.2 GDAL Rasterformate

Zum Zeitpunkt der Erstellung dieses Dokuments werden folgende Datenformate durch die GDAL-Bibliothek unterstützt. Beachten Sie, dass nicht alle angegebenen Formate unter QGIS funktionieren werden. Einige benötigen kommerzielle Bibliotheken. Nur getestete Formate erscheinen im Auswahl-fenster, wenn sie eine Rasterebene laden möchten. Andere, nicht gesteste Formate können Sie laden, indem Sie den *Alle anderen Daten (\*)* Filter anwenden. Formate, die unter QGIS getestet wurden, sind **fett** dargestellt.

---

<sup>8</sup>QGIS hat eine eigene PostgreSQL Funktion. OGR sollte ohne PostgreSQL support erstellt werden

- **Arc/Info ASCII Grid**
- **Arc/Info Binary Grid (.adf)**
- Microsoft Windows Device Independent Bitmap (.bmp)
- BSB Nautical Chart Format (.kap)
- VTP Binary Terrain Format (.bt)
- CEOS (Spot for instance)
- First Generation USGS DOQ (.doq)
- New Labelled USGS DOQ (.doq)
- Military Elevation Data (.dt0, .dt1)
- ERMapper Compressed Wavelets (.ecw)
- ESRI .hdr Labelled
- ENVI .hdr Labelled Raster
- Envisat Image Product (.n1)
- EOSAT FAST Format
- FITS (.fits)
- Graphics Interchange Format (.gif)
- **GRASS Raster<sup>9</sup>**
- **TIFF / GeoTIFF (.tif)**
- Hierarchical Data Format Release 4 (HDF4)
- **Erdas Imagine (.img)**
- Atlantis MFF2e
- Japanese DEM (.mem)
- **JPEG JFIF (.jpg)**
- JPEG2000 (.jp2, .j2k)
- JPEG2000 (.jp2, .j2k)
- NOAA Polar Orbiter Level 1b Data Set (AVHRR)
- Erdas 7.x .LAN and .GIS
- In Memory Raster
- Atlantis MFF
- Multi-resolution Seamless Image Database MrSID
- NITF

---

<sup>9</sup>GRASS Rastersupport wird mit dem QGIS GRASS-Plugin bereitgestellt.

- NetCDF
- OGDl Bridge
- PCI .aux Labelled
- PCI Geomatics Database File
- Portable Network Graphics (.png)
- Netpbm (.ppm,.pgm)
- **USGS SDTS DEM (\*CATD.DDF)**
- SAR CEOS
- **USGS ASCII DEM (.dem)**
- X11 Pixmap (.xpm)

---

## B Installationsanleitung

Das folgende Kapitel stellt den Installationsvorgang von QGIS Version 0.9.1 dar. Dieser Abschnitt ist nahezu eine  $\text{\LaTeX}$  Konvertierung der Datei INSTALL.t2t aus dem SVN-Repository vom 29. November 2007.

Eine aktuelle Version ist auch im Wiki einsehbar, siehe <http://wiki.qgis.org/qgiswiki/BuildingFromSource>

### B.1 Grundsätzliche Bauhinweise

Seit Version 0.8.1 wird QGIS nicht mehr mit den Autotools gebaut. Wie viele andere Projekte auch (z.B. KDE) benutzt QGIS nun cmake zum Bauen aus den Quellen. Das configure-Skript überprüft nur, ob cmake installiert ist und bietet einige Hinweise zum Bauen von QGIS.

Für ausführlichere Informationen schauen sie in das Wiki unter [http://wiki.qgis.org/qgiswiki/Building\\_with\\_CMake](http://wiki.qgis.org/qgiswiki/Building_with_CMake)

### B.2 Übersicht über die Abhängigkeiten zum Bauen

#### Benötigte Pakete zum Bauen:

- CMake  $\geq 2.4.3$
- Flex, Bison

#### Benötigte Paket-Abhängigkeiten zur Ausführung:

- Qt  $\geq 4.2.0$
- Proj  $\geq ?$  (funktioniert mit mind. Version 4.4.x)
- GEOS  $\geq 2.2$  (3.0 wird unterstützt, evtl. funktioniert auch Version 2.1.x)
- Sqlite3  $\geq ?$  (wahrscheinlich 3.0.0)
- GDAL/OGR  $\geq ?$  (1.2.x sollte funktionieren)

#### Optionale Abhängigkeiten:

- für das GRASS Plugin - GRASS  $\geq 6.0.0$
- für das Georeferenzier-Plugin - GSL  $\geq ?$  (funktioniert mit 1.8)
- für die PostGIS-Unterstützung im SPIT Plugin - PostgreSQL  $\geq ?$
- für das GPS Plugin - expat  $\geq ?$  (1.95 ist OK)
- für den MapServer Export und PyQGIS - Python  $\geq ?$  (wahrscheinlich 2.3)

- 
- für PyQGIS - SIP  $\geq$  4.5, PyQt  $\geq$  4.1

#### **Empfohlene Pakete zur Ausführung:**

- für das GPS Plugin - gpsbabel

## **C Bauen unter Windows mit msys**

### **C.1 MSYS:**

MSYS stellt ein Unix-ähnliches Bausystem unter Windows zur Verfügung. Wir haben eine Zip-Datei erstellt, die alle Abhängigkeiten zum Bauen enthält.

Laden Sie es hier herunter:

<http://qgis.org/uploadfiles/msys/msys.zip>

und packen Sie es unter c:\msys aus.

Falls Sie Ihre eigene msys-Umgebung erstellen wollen, folgen Sie den detaillierten Erläuterungen weiter unten in diesem Dokument.

### **C.2 Qt4.3**

Laden Sie qt4.3 opensource precompiled edition exe herunter und installieren Sie es. Eine Kopie von mingw ist bereits darin enthalten:

<http://www.trolltech.com/developer/downloads/qt/windows>

Wenn das Installationsprogramm Sie nach MinGW fragt, brauchen Sie es nicht extra herunterzuladen und zu installieren. Zeigen Sie dem Installer lediglich den Ort der Installation unter c:\msys\mingw

Nach erfolgreicher Qt-Installation editieren Sie die folgende Datei:

C:\Qt\4.3.0\bin\qtvars.bat

Fügen Sie die folgenden Zeilen hinzu:

```
set PATH=%PATH%;C:\msys\local\bin;c:\msys\local\lib
set PATH=%PATH%;'C:\Program Files\Subversion\bin'
```

Außerdem empfehlen wir, die Pfade C:\Qt\4.3.0\bin\ zu Ihrer Umgebungsvariablen Path in den Windows Systemeinstellungen hinzuzufügen.

Falls Sie planen, Debugging zu machen, müssen Sie Qt mit der Debug-Option kompilieren:  
`C:\Qt\4.3.0\bin\qtvars.bat compile_debug`

Achtung: Es existiert ein Problem mit der Debug-Version von Qt 4.3; das Skript endet mit der Meldung 'mingw32-make: \*\*\* No rule to make target 'debug'. Stop.'. Um dennoch eine Debug-Version zu kompilieren, verlassen sie den Ordner src führen folgendes Kommando aus:

```
c:\Qt\4.3.0 make
```

### C.3 Flex and Bison

Anmerkung: Dieser Abschnitt kann in der Zukunft entfernt werden, da die Pakete bereits im msys-Image enthalten sind. TS

Holen Sie Flex: [http://sourceforge.net/project/showfiles.php?group\\_id=23617&package\\_id=16424](http://sourceforge.net/project/showfiles.php?group_id=23617&package_id=16424) (die Zip-Datei) und extrahieren Sie es nach `c:\msys\mingw\bin`

### C.4 Python: (optional)

Falls Sie die Python-Anbindung von QGIS nutzen wollen, sollten Sie diesen Abschnitt weiterlesen. Um die Anbindung zu Python zu kompilieren, benötigen Sie SIP und PyQt4 aus den Quellen, denn die Binär-Installationsprogramme bringen einige benötigte Header-Dateien leider nicht mit; diese sind jedoch nötig.

#### C.4.1 Herunterladen und Installieren von Python - mit dem Windows-Installationsprogramm

(Es ist unerheblich, in welchem Ordner Sie Python installieren)

<http://python.org/download/>

#### C.4.2 SIP- und PyQt4-Quellen herunterladen

<http://www.riverbankcomputing.com/Downloads/sip4/><http://www.riverbankcomputing.com/Downloads/PyQt4/>

Extrahieren Sie beide Zip-Dateien in temporäre Verzeichnisse. Versichern Sie sich, dass Sie auch die für Ihre Qt-Installation passenden Versionen erwisch haben.

### C.4.3 SIP kompilieren

```
c:\Qt\4.3.0\bin\qtvars.bat
python configure.py -p win32-g++
make
make install
```

### C.4.4 PyQt kompilieren

```
c:\Qt\4.3.0\bin\qtvars.bat
python configure.py
make
make install
```

### C.4.5 Finale Python-Anmerkungen

/!\ Nach einer erfolgreichen Installation von SIP und PyQt4 können Sie die temporären Quellverzeichnisse löschen, sie werden nicht mehr gebraucht.

## C.5 Subversion:

Falls Sie QGIS aus dem SVN Repository beziehen wollen, benötigen Sie einen Subversion-Klienten. Der angebotene Installer funktioniert gut:

<http://subversion.tigris.org/files/documents/15/36797/svn-1.4.3-setup.exe>

## C.6 CMake:

CMake ist das von QGIS benutzte Bausystem. Laden Sie es hier herunter:

<http://www.cmake.org/files/v2.4/cmake-2.4.6-win32-x86.exe>

## C.7 QGIS:

Starten Sie eine cmd.exe (Start -> Ausführen -> cmd.exe). Erstellen Sie eine Entwicklungsumgebung und wechseln Sie in das Verzeichnis.



```
md c:\dev\cpp
cd c:\dev\cpp
```

Extrahieren Sie die QGIS-Quellen aus dem SVN (hier für trunk):

```
svn co https://svn.qgis.org/repos/qgis/trunk/qgis
```

Für die Version 0.8

```
svn co https://svn.qgis.org/repos/qgis/branches/Release-0_8_0 qgis 0.8
```

## C.8 Kompilierung:

Für generelle Hintergrundinformationen lesen Sie die Information über CMake am Ende dieses Dokumentes.

Starten Sie eine cmd.exe ( Start -> Ausführen -> cmd.exe ) falls Sie noch keine gestartet haben. Setzen Sie die Pfade für den Compiler und die MSYS-Umgebung:

```
c:\Qt\4.3.0\bin\qtvars.bat
```

Der Einfachheit halber fügen Sie c:\Qt\4.3.0\bin\ zu Ihrer Umgebungsvariable Patg hinzu, sodass Sie einfach qtvars.bat eintippen können. Erzeugen Sie ein Bauverzeichnis und wechseln Sie hinein:

```
cd c:\dev\cpp\qgis
md build
cd build
```

## C.9 Konfiguration

```
cmakesetup ..
```

**Achtung:** Sie müssen die beiden Punkte '..' mit angeben!

Klicken Sie den 'Konfigurieren'-Knopf. Falls Sie nach dem Generator gefragt werden, wählen Sie 'MinGW Makefiles'.

Es existiert ein Problem mit MinGW Makefiles auf Windows 2000. Falls Sie auf dieser Plattform kompilieren, nutzen Sie stattdessen 'MSYS Makefiles' als Generator.

Alle Abhängigkeiten sollen nun automatisch gefunden werden, wenn Sie die Pfade korrekt eingestellt haben. Das einzige, was Sie noch manuell angeben müssen, ist das Installationsverzeichnis und/oder die Debug-Einstellungen (CMAKE\_INSTALL\_PREFIX).

Um die Kompatibilität zum NSIS-Paketierskript zu gewährleisten, empfehlen wir die Einstellungen beizubehalten (c:\program files\).

Wenn die Konfiguration beendet ist, klicken Sie 'OK' und beenden das Setup-Programm.

## C.10 Kompilieren und installieren

```
make  
make install
```

## C.11 qgis.exe aus dem Installationsverzeichnis starten (CMAKE\_INSTALL\_PREFIX)

Stellen Sie sicher, dass alle .dlls im gleichen Verzeichnis liegen, in dem auch die Datei qgis.exe liegt. Falls dies nicht der Fall ist, wird sich QGIS beschweren, da einige dlls nicht gefunden werden können.

Der beste Weg, um dies sicherzustellen, ist, wenn Sie parallel den offiziellen QGIS Release-Installer herunterladen und installieren (<http://qgis.org/uploadfiles/testbuilds/>). Nun kopieren Sie das Installationsverzeichnis von C:\Program Files\Quantum GIS nach c:\Program Files\qgis-0.8.1 (oder welche Version Sie gerade benutzen. Der Name sollte exakt die Versionsnummer treffen.)

Nachdem Sie diese Kopie durchgeführt haben, können Sie die eben installierte Release-Version von QGIS wieder mit dem mitgelieferten Uninstaller deinstallieren. Vergewissern Sie sich, dass das Verzeichnis Quantum GIS unterhalb von c:\Program Files komplett entfernt wurde.

Eine andere Möglichkeit ist das Ausführen von qgis.exe wenn Ihr Pfad die folgenden Verzeichnisse enthält: c:\msys\local\bin und c:\msys\local\lib. Die DLLs werden nun aus diesen Ordnern verwendet.

## C.12 Das Installationspaket erzeugen: (optional)

Laden Sie NSIS ([http://nsis.sourceforge.net/Main\\_Page](http://nsis.sourceforge.net/Main_Page)) herunter und installieren Sie es.

Nun können Sie den Windows-Explorer benutzen und in das Verzeichnis win\_build Ihres QGIS-Quellbaums wechseln. Lesen Sie die Datei README und folgen Sie den Instruktionen. Als nächstes klicken Sie mit der rechten Maustaste auf die Datei qgis.nsi und wählen Sie die Option 'Kompiliere NSIS-Skript'.

---

## D Bauen auf Mac OSX mit frameworks und cmake (QGIS > 0.8)

Mit diesem Ansatz wollen wir, soweit möglich, alle Bau-Abhängigkeiten mit den Framework-Paketen erledigen und so wenig wie möglich auf Quellpakete zurückgreifen.

### D.1 Installieren von XCODE

Wir empfehlen die Verwendung der aktuellsten Version des Paketes xcode dmg von Apples XDC-Webseite. Installieren sie XCODE nachdem Sie das ca. 941MB große Paket heruntergeladen haben.

### D.2 Installieren des Qt4 .dmg

Als Minimalanforderung benötigen Sie Qt4.2. Dennoch empfehlen wir die aktuellste Version.

`ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.3.2.dmg`

Falls Sie Debug-Bibliotheken wollen, laden Sie stattdessen das folgende dmg herunter:

`ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.3.2-debug-libs.dmg`

Von hier an werden wir nur die Release-Bibliotheken verwenden, denn das Herunterladen der Debug-Dmgs dauert länger (da die Pakete größer sind). Falls Sie jedoch planen, die Debug-Optionen zu benutzen, greifen Sie stattdessen zu den Debug-Bibliotheken. Für alle Dmgs gilt: Nach dem Download öffnen Sie das Dmg und starten den Installer. Beachten Sie, dass Sie dafür Admin-Rechte benötigen.

Nach dem Installieren müssen Sie zwei kleine Änderungen vornehmen:

Als erstes editieren Sie `/Library/Frameworks/QtCore.framework/Headers/qconfig.h` und ändern

`!\` Dies scheint seit Version 4.2.3 nicht mehr nötig zu sein.

`QT_EDITION_UNKNOWN` zu `QT_EDITION_OPENSOURCE`

Als nächstes ändern Sie den symbolischen mkspec-Link auf `macx-g++`:

```
cd /usr/local/Qt4.3/mkspecs/  
sudo rm default  
sudo ln -sf macx-g++ default
```

### D.3 Installation der Entwicklungsumgebung für die QGIS-Anhängigkeiten

Laden Sie William Kyngesburye's exzellentes All-in-One-Framework-Paket mit proj, gdal, sqlite3 etc herunter und installieren Sie es.

<http://www.kyngchaos.com/files/software/unixport/AllFrameworks.dmg>

William stellt weiterhin ein Paket für PostgreSQL/PostGIS bereits. Es ist hier erhältlich:

<http://www.kyngchaos.com/software/unixport/postgres>

Einige weitere Abhängigkeiten sind zum Zeitpunkt des Schreibens dieser Anleitung noch nicht als Framework erhältlich, sodass wir sie noch aus den Quellen nachinstallieren müssen.

#### D.3.1 Zusätzliche Abhängigkeit: GSL

Laden Sie GSL von

```
curl -O ftp://ftp.gnu.org/gnu/gsl/gsl-1.8.tar.gz
```

herunter und bauen Sie es mit dem Prefix /usr/local:

```
tar xvfz gsl-1.8.tar.gz
cd gsl-1.8
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

#### D.3.2 Zusätzliche Abhängigkeit: Expat

Laden Sie expat von

[http://sourceforge.net/project/showfiles.php?group\\_id=10127](http://sourceforge.net/project/showfiles.php?group_id=10127)

herunter und bauen Sie es mit dem Prefix /usr/local:

```
tar xvfz expat-2.0.0.tar.gz
cd expat-2.0.0
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

### D.3.3 Zusätzliche Abhängigkeit: SIP

Laden Sie SIP von

<http://www.riverbankcomputing.com/Downloads/sip4/>

herunter und bauen Sie es mit dem Prefix /usr/local:

```
tar xvfz sip-<version number>.tar.gz
cd sip-<version number>
python configure.py
make
sudo make install
cd ..
```

### D.3.4 Zusätzliche Abhängigkeit: PyQt

Stellen Sie sicher, dass Sie die aktuellste Version von Python von

<http://www.python.org/download/mac/>

haben.

Wenn Sie Probleme beim Kompilieren von PyQt mit den nachstehenden Instruktionen haben, können Sie versuchen, den Python-Pfad im Framework-Verzeichnis direkt in den Pfad einzutragen, in etwa so:

```
export PATH=/Library/Frameworks/Python.framework/Versions/Current/bin:$PATH$
```

Holen Sie sich die Python-Bindungen für Qt von

<http://www.riverbankcomputing.com/Downloads/PyQt4/GPL/>

Nun extrahieren und bauen Sie es mit dem Prefix /usr/local:

```
tar xvfz PyQt-mac<version number here>
cd PyQt-mac<version number here>
python configure.py
yes
make
sudo make install
cd ..
```

### D.3.5 Zusätzliche Abhängigkeite: Bison

Die standardmäßig für Mac OSX verfügbare Version von Bison ist zu alt. Sie benötigen eine neuere Version, die Sie hier bekommen.

```
curl -O http://ftp.gnu.org/gnu/bison/bison-2.3.tar.gz
```

Nun bauen und installieren Sie die Software mit dem Prefix /usr/local:

```
tar xvfz bison-2.3.tar.gz
cd bison-2.3
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

## D.4 CMake für Mac OSX

Holen Sie sich die aktuellste Version von hier:

<http://www.cmake.org/HTML/Download.html>

Zur Zeit des Schreibens war der folgende Link aktuell:

```
curl -O http://www.cmake.org/files/v2.4/cmake-2.4.6-Darwin-universal.dmg
```

Einmal heruntergeladen, öffnen Sie das dmg und führen den Installer aus.

## D.5 Subversion für Mac OSX

Der Link <http://sourceforge.net/projects/macsvn/> hat eine nutzbare Version von Subversion. Es existiert auch ein graphisches Frontend. Das Kommandozeilenwerkzeug können Sie hier finden:

```
curl -O http://ufpr.dl.sourceforge.net/sourceforge/macsvn/Subversion_1.4.2.zip
```

Nach dem Download betätigen Sie den Installer.

Sie benötigen auch die BerkeleyDB, welche von der gleichen <http://sourceforge.net/projects/macsvn/> erhältlich ist. Der folgende Link war beim Schreiben dieser Zeilen aktuell:

```
curl -O http://ufpr.dl.sourceforge.net/sourceforge/macsvn/Berkeley_DB_4.5.20.zip
```

Wieder einmal auspacken und den Installer ausführen.

Wir müssen noch sicher stellen, dass das SVN Kommandozeilenwerkzeug im Pfad auffindbar ist. Dazu fügen Sie folgende Zeilen an das Ende der Datei `/etc/bashrc` unter Zuhilfenahme von `sudo`:

```
sudo vim /etc/bashrc
export PATH=/usr/local/bin:$PATH:/usr/local/pgsql/bin
```

`/usr/local/bin` muss als erster Eintrag im Pfad vorhanden sein, sodass die neuere Version von Bison zuerst gefunden wird.

Nun schließen Sie Ihre Shell und öffnen gleich eine neue; dies aktualisiert Ihre Umgebungsvariablen.

## D.6 QGIS aus dem SVN auschecken

Nun wollen wir die QGIS-Quellen aus dem SVN auschecken. Dazu erzeugen wir zuerst einen Ordner, in dem wir nachfolgend arbeiten wollen.

```
mkdir -p ~/dev/cpp
cd ~/dev/cpp
```

Nun checken wir die Quellen aus:

Für Trunk:

```
svn co https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

Für den 0.8 Zweig:

```
svn co https://svn.qgis.org/repos/qgis/branches/Release-0_8_0 qgis0.8
```

Für den 0.9 Zweig:

```
svn co https://svn.qgis.org/repos/qgis/branches/Release-0_9_0 qgis0.9
```

Beim ersten Auschecken bekommen Sie möglicherweise diesen Fehler:

```
Error validating server certificate for 'https://svn.qgis.org:443':
- The certificate is not issued by a trusted authority. Use the fingerprint to
  validate the certificate manually! Certificate information:
- Hostname: svn.qgis.org
- Valid: from Apr  1 00:30:47 2006 GMT until Mar 21 00:30:47 2008 GMT
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
- Fingerprint: 2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b
  (R) eject, accept (t)emporarily or accept (p)ermanently?
```

Wir empfehlen, dieses selbst signierte Zertifikat einmal permanent mit der Taste 'p' zu akzeptieren.

## D.7 Konfiguration der Quellen

CMake unterstützt das Bauen außerhalb des Quellbaums und erzeugt ein 'build'-Verzeichnis für den Bauprozess. Einer persönlichen Konvention folgend installiere ich meine Software in meinem Homeverzeichnis in einem Ordner namens 'apps'. Falls Sie die Rechte besitzen, können Sie Ihre Software auch direkt in den Ordner /Applications bauen (auch wenn ich das nicht wirklich empfehlen kann). Die nachfolgenden Schritte gehen davon aus, dass Sie in das bereits existente Verzeichnis \${HOME}/apps installieren...

```
cd qgis
mkdir build
cd build
cmake -D CMAKE_INSTALL_PREFIX=$HOME/apps/ -D CMAKE_BUILD_TYPE=Release ..
```

Um eine spezielle GRASS-Version zu benutzen, können Sie optional die folgenden Einstellungen für CMake nutzen. (Dank an William Kyngesburye für diesen Tipp):



```
cmake -D CMAKE_INSTALL_PREFIX=${HOME}/apps/ \
-D GRASS_INCLUDE_DIR=/Applications/GRASS-6.3.app/Contents/Resources/include \
-D GRASS_PREFIX=/Applications/GRASS-6.3.app/Contents/Resources \
-D CMAKE_BUILD_TYPE=Release \
..
```

## D.8 GEOS Hinweise

In den GEOS-Headerdateien sind einige Probleme aufgetreten.

In der Datei /Library/Frameworks/GEOS.framework/Headers/io.h habe ich Zeile 61 auskommentiert.

In der Datei /Library/Frameworks/GEOS.framework/Headers/geom.h habe ich Zeile 145 auskommentiert.

## D.9 Bauen

Starten Sie nun den Bauprozess:

```
make
```

Falls das Bauen ohne Fehler abgeschlossen wurde, können Sie es installieren:

```
make install
```

## E Bauen unter GNU/Linux

### E.1 QGIS mit Qt4.x bauen

**\*Ausgangssituation:\*** Ubuntu Edgy / Debian-basierte Distribution

Diese Anleitung ist zum Selbstbauen von QGIS aus den Quellen. Eines der großen Ziele war es, zu zeigen, wie man für **\*alle\*** Abhängigkeiten Binärpakete nutzen kann. Nur QGIS selbst wird aus den Quellen gebaut. Ich verfolge diesen Ansatz, da ich alle Abhängigkeiten dem Paketmanager apt überlasse und alle Energie in die Entwicklung von QGIS setzen kann!

Dieses Dokument geht von einer frischen Installation und einem 'sauberen' System aus. Natürlich funktioniert die Anleitung auch auf bereits lange laufenden Systemen, Sie werden u.U. einige Schritte überspringen können, da Sie irrelevant für Sie sind.

## E.2 Apt vorbereiten

Alle Pakete, von denen QGIS abhängt, sind im 'universe'-Repository von Ubuntu enthalten. Dieses ist standardmäßig nicht aktiviert, Sie müssen es manuell aktivieren:

1. Editieren Sie /etc/apt/sources.list file. 2. Entkommentieren Sie alle Zeilen, die mit 'deb' starten.

Ferner sollten Sie (K)Ubuntu 'edgy' oder höher laufen lassen, um alle Abhängigkeiten verfügbar zu haben.

Nun aktualisieren Sie Ihre lokale Paketdatenbank:

```
sudo apt-get update
```

## E.3 Installieren von Qt4

```
sudo apt-get install libqt4-core libqt4-debug \
libqt4-dev libqt4-gui libqt4-qt3support libqt4-sql lsb-qt4 qt4-designer \
qt4-dev-tools qt4-doc qt4-qtconfig uim-qt gcc libapt-pkg-perl resolvconf
```

/!\ **\*Spezialhinweis:** Wenn Sie dieser Anleitung auf einem System nachfolgen, wo bereits die Qt3-Entwicklungswerkzeuge installiert sind, wird ein Konflikt zwischen Qt3 und Qt4 Werkzeugen auftreten. Zum Beispiel zeigt qmake auf die Qt3-Version und nicht auf die bei Qt4 enthaltende. Ubuntu's Qt3 und Qt4-Pakete sind so entwickelt, dass sie koexistent nebeneinander installiert sein können. Falls Sie beide Versionen installiert haben, sehen Sie, dass es drei qmake-Versionen gibt:

```
/usr/bin/qmake -> /etc/alternatives/qmake
/usr/bin/qmake-qt3
/usr/bin/qmake-qt4
```

Selbiges gilt natürlich für die Qt Binärdateien. Sie werden bemerken, dass das kanonische 'qmake' durch das Werkzeug 'apt alternatives' koordiniert wird. Ferner müssen wir die Qt4-Einstellungen als Standard einstellen, bevor wir mit dem Bauen von QGIS starten können. Um wieder zu den Qt3-Einstellungen zurückzukommen, gehen Sie einfach den umgekehrten Weg.

Sie können apt alternatives zur Korrektur der Qt4-Version benutzen:

```
sudo update-alternatives --config qmake
sudo update-alternatives --config uic
sudo update-alternatives --config designer
sudo update-alternatives --config assistant
```

```
sudo update-alternatives --config qtconfig
sudo update-alternatives --config moc
sudo update-alternatives --config lupdate
sudo update-alternatives --config lrelease
sudo update-alternatives --config linguist
```

Nutzen Sie den einfachen Kommandozeilendialog, der nach jeder Zeile ausgeführt wird, um die Qt4-Version der entsprechenden Anwendung auszuwählen.

## E.4 Installation weitere Abhängigkeiten

```
sudo apt-get install gdal-bin libgdal1-dev libgeos-dev proj \
libgdal-doc libhdf4g-dev libhdf4g-run python-dev \
libgs10-dev g++ libjasper-1.701-dev libtiff4-dev subversion \
libsqlite3-dev sqlite3 ccache make libpq-dev flex bison cmake txt2tags \
python-qt4 python-qt4-dev python-sip4 sip4 python-sip4-dev
```

! \ Debian Benutzer sollten libgdal-dev anstatt libgdal1-dev benutzen.

! \ **\*Hinweis:** Für Python werden die Anbindungen SIP  $\geq 4.5$  und PyQt4  $\geq 4.1$  benötigt! Einige stabile GNU/Linux Distributionen (z.B.: Debian Etch oder SuSE) haben aber nur SIP  $< 4.5$  und PyQt4  $< 4.1$ . Um Python-Unterstützung zu bekommen, müssen Sie die Pakete aus den Quellen installieren.

## E.5 GRASS-spezifische Schritte

! \ **\*Hinweis:** Falls Sie die GRASS-Unterstützung nicht bauen möchten, können Sie diesen Abschnitt überspringen.

Sie können GRASS für Dapper installieren:

```
sudo apt-get install grass libgrass-dev libgdal1-grass
```

! \ Evtl. müssen Sie die zu installierende GRASS-Version direkt angeben, z.B. libgdal1-1.3.2-grass

## E.6 Einrichtung von ccache (optional)

Sie sollten ccache einrichten, um die Kompilierzeit zu reduzieren:

```
cd /usr/local/bin
sudo ln -s /usr/bin/ccache gcc
sudo ln -s /usr/bin/ccache g++
```

## E.7 Einrichten der Entwicklungsumgebung

Als (persönliche) Konvention richte ich meine Entwicklungsumgebung in meinem Home-Verzeichnis unter `$HOME/dev/<Sprache>` ein. In diesem Fall wird eine C++-Umgebung wie folgt eingerichtet:

```
mkdir -p ${HOME}/dev/cpp
cd ${HOME}/dev/cpp
```

Alle weiteren Pfade beziehen sich nun auf diesen Ordner

## E.8 QGIS-Quellen auschecken

Es existieren zwei Wege um die Quellen auszuchecken. Der anonyme Weg sollte gewählt werden, wenn Sie nicht vorhaben, Änderungen an den Quellen zurück in das Repository zu schreiben. Entwickler mit Schreibberechtigungen sollten den Quellcode mit ihrer Entwicklerkennung auschecken, um Änderungen am Code zurückschreiben zu können.

### 1. Anonymer Checkout

```
cd ${HOME}/dev/cpp
svn co https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

### 2. Entwickler Checkout

```
cd ${HOME}/dev/cpp
svn co --username <IhrBenutzername> https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

Das erste Mal, wenn Sie QGIS auschecken, werden Sie gefragt, ob Sie das HTTPS-Zertifikat von qgis.org akzeptieren möchten. Dazu sollten Sie 'p' drücken, um es permanent zu akzeptieren.

```
Error validating server certificate for 'https://svn.qgis.org:443':
- The certificate is not issued by a trusted authority. Use the
  fingerprint to validate the certificate manually! Certificate
  information:
```

```
- Hostname: svn.qgis.org
- Valid: from Apr  1 00:30:47 2006 GMT until Mar 21 00:30:47 2008 GMT
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
- Fingerprint:
  2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b (R)eject,
  accept (t)emporarily or accept (p)ermanently?
```

## E.9 Start des Kompiliervorgangs

Programme werden bei mir unterhalb von ~/apps installiert. Dies separiert die Ubuntu-Pakete von den aus den Quellen gebauten Paketen. Auf diese Weise können Ubuntu Binärpakete und die selbst-kompilierte Version von QGIS nebeneinander existieren. Dazu empfehlen wir Folgendes:

```
mkdir -p ${HOME}/apps
```

Nun kann der Bau-Ordner erstellt und cmake aufgerufen werden:

```
cd qgis
mkdir build
cd build
cmake ..
```

Nach dem Aufrufen von cmake (Bitte die beiden Punkte .. beachten, Sie sind notwendig!) wird ein Menü erscheinen, in dem Sie verschiedene Configure-Optionen zum Bauen eingestellt werden können. Falls Sie keinen Root-Zugriff haben (oder Ihre Binärinstallation Ihres Paketmanagers nicht überschreiben wollen), setzen Sie CMAKE\_BUILD\_PREFIX auf einen Pfad, in dem Sie Schreibrechte haben. (Ich empfehle \$HOME/apps). Nun drücken Sie 'c' um den Konfiguriervorgang zu starten, 'e' für das Eliminieren jeglicher Fehlermeldungen, die erscheinen könnten. 'g' erstellt letztendlich die Makefiles. Bitte beachten Sie, dass in manchen Fällen ein mehrmaliges Drücken von 'c' nötig ist, bis der Knopf 'g' verfügbar wird. Nach der 'g' Erzeugung drücken Sie 'q' um cmakes interaktiven Dialog zu beenden.

Nun das Bauen:

```
make
make install
```

Je nach Plattform kann das Bauen ein kleine Weile dauern.

## E.10 QGIS ausführen

Nun können Sie versuchen, QGIS auszuführen:

```
$HOME/apps/bin/qgis
```

Wenn alles korrekt durchgelaufen ist, sollte QGIS nun starten und auf Ihrem Bildschirm erscheinen.

## F Erzeugen eine MSYS-Umgebung zum Kompilieren von QGIS

### F.1 Initiale Einrichtung

#### F.1.1 MSYS

Diese Umgebung bringt eine Menge Entwicklerwerkzeuge aus der UNIX-Welt nach Windows, die zum Kompilieren nötig sind.

Hier herunterladen:

<http://puzzle.dl.sourceforge.net/sourceforge/mingw/MSYS-1.0.11-2004.04.30-1.exe>

Installieren Sie unter `c:\msys`

Alle nachfolgenden Programme, die Sie kompilieren, werden in Unterverzeichnissen von `c:\msys` abgelegt.

#### F.1.2 MinGW

Hier Herunterladen:

<http://puzzle.dl.sourceforge.net/sourceforge/mingw/MinGW-5.1.3.exe>

Installieren Sie es unter `c:\msys\mingw`

Es reicht wenn Sie nur die Komponenten `g++` und `mingw-make` herunterladen und installieren.

#### F.1.3 Flex und Bison

Flex und Bison sind Werkzeuge, die Parser erzeugen; sie sind beide für die GRASS- und QGIS-Kompilation nötig.

Laden Sie die folgenden Pakete herunter:

<http://gnuwin32.sourceforge.net/downlinks/flex-bin-zip.php>

<http://gnuwin32.sourceforge.net/downlinks/bison-bin-zip.php>

<http://gnuwin32.sourceforge.net/downlinks/bison-dep-zip.php>

Entpacken Sie alles unter `c:\msys\local`

## F.2 Abhängigkeiten installieren

### F.2.1 Ausgangssituation fertigstellen

Paul Kelly hat eine tolle Vorarbeit gemacht und vorkompilierte Bibliotheken für GRASS zusammengestellt. Das Paket enthält derzeit:

- zlib-1.2.3
- libpng-1.2.16-noconfig
- xdr-4.0-mingw2
- freetype-2.3.4
- fftw-2.1.5
- PDCurses-3.1
- proj-4.5.0
- gdal-1.4.1

Es kann hier heruntergeladen werden:

<http://www.stjohnspoint.co.uk/grass/wingrass-extralibs.tar.gz>

Darüber hinaus hat er einige Notizen hinterlassen, wie man diese Bibliotheken baut (für die, die es interessiert):

<http://www.stjohnspoint.co.uk/grass/README.extralibs>

Das gesamte Paket sollte unter `c:\msys\local` entpackt werden.

### F.2.2 GDAL, erster Schritt

Seit QGIS die Bibliothek GDAL mit GRASS-Unterstützung benötigt, müssen wir GDAL aus den Quellen kompilieren - Paul Kellys Paket hat keine GRASS-Unterstützung in GDAL. Die Idee ist die folgen-

de:

1. GDAL ohne GRASS kompilieren
2. GRASS kompilieren
3. GDAL mit GRASS kompilieren

So, nun erstmal die GDAL-Quellen herunterladen:

<http://download.osgeo.org/gdal/gdal141.zip>

Entpacken Sie dies in das folgende Verzeichnis: `c:\msys\local\src`.

Starten Sie die MSYS-Konsole, wechseln Sie in das Verzeichnis `gdal-1.4.1` und führen Sie die folgenden Kommandos aus. Sie können dies auch in eine Datei `build-gdal.sh` schreiben und alle auf einmal ausführen. Das Rezept ist von Paul Kellys Notizen extrahiert; es stellt lediglich sicher, dass DLLs erzeugt werden und diese auch dynamisch gelinkt sind...

```
CFLAGS='-O2 -s' CXXFLAGS='-O2 -s' LDFLAGS=-s ./configure --without-libtool
--prefix=/usr/local --enable-shared --disable-static --with-libz=/usr/local
--with-png=/usr/local
make
make install
rm /usr/local/lib/libgdal.a
g++ -s -shared -o ./libgdal.dll -L/usr/local/lib -lz -lpng ./frmts/o/*.o
./gcore/*.o ./port/*.o ./alg/*.o ./ogr/ogrsf_frmts/o/*.o
./ogr/ogrgeometryfactory.o ./ogr/ogrpoint.o ./ogr/ogrcurve.o
./ogr/ogrlinestring.o ./ogr/ogrlinearring.o ./ogr/ogrpolygon.o
./ogr/ogrutils.o ./ogr/ogrgeometry.o ./ogr/ogrgeometrycollection.o
./ogr/ogrmultipolygon.o ./ogr/ogrsurface.o ./ogr/ogrmultipoint.o
./ogr/ogrmultilinestring.o ./ogr/ogr_api.o ./ogr/ogrfeature.o
./ogr/ogrfeaturedefn.o ./ogr/ogrfeaturequery.o ./ogr/ogrfeaturestyle.o
./ogr/ogrfielddefn.o ./ogr/ogrspatialreference.o ./ogr/ogr_srsnode.o
./ogr/ogr_srs_proj4.o ./ogr/ogr_fromepsg.o ./ogr/ogrct.o ./ogr/ogr_opt.o
./ogr/ogr_srs_esri.o ./ogr/ogr_srs_pci.o ./ogr/ogr_srs_usgs.o
./ogr/ogr_srs_dict.o ./ogr/ogr_srs_panorama.o ./ogr/swq.o
./ogr/ogr_srs_validate.o ./ogr/ogr_srs_xml.o ./ogr/ograssemblepolygon.o
./ogr/ogr2gmlgeometry.o ./ogr/gml2ogrgeometry.o
install libgdal.dll /usr/local/lib
cd ogr
g++ -s ogrinfo.o -o ogrinfo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s ogr2ogr.o -o ogr2ogr.exe -lgdal -L/usr/local/lib -lpng -lz -lgdal
g++ -s ogrtindex.o -o ogrtindex.exe -lgdal -L/usr/local/lib -lpng -lz -lgdal
```



```
install ogrinfo.exe ogr2ogr.exe ogrtindex.exe /usr/local/bin
cd ../apps
g++ -s gdalinfo.o -o gdalinfo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_translate.o -o gdal_translate.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdaladdo.o -o gdaladdo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdalwarp.o -o gdalwarp.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_contour.o -o gdal_contour.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdaltindex.o -o gdaltindex.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_rasterize.o -o gdal_rasterize.exe -L/usr/local/lib -lpng -lz -lgdal
install gdalinfo.exe gdal_translate.exe gdaladdo.exe gdalwarp.exe
gdal_contour.exe gdaltindex.exe gdal_rasterize.exe /usr/local/bin
```

Schlussendlich editieren Sie `gdal-config` in `c:\msys\local\bin` um die statische Bibliotheksreferenz mit `-lgdal` zu ersetzen:

```
CONFIG_LIBS='-L/usr/local/lib -lpng -lz -lgdal'
```

Die GDAL-Bauprozedur kann stark vereinfacht werden, wenn das Werkzeug `libtool` mit einem Patch benutzt wird:

```
./configure --with-ngpython --with-xerces=/local/ --with-jasper=/local/ --with-grass=/local/grass-6.3.cvs/
--with-pg=/local/pgsql/bin/pg_config.exe
```

Nun sollte `libtool` gefixt werden: `mv libtool libtool.orig cat libtool.orig | sed 's/max_cmd_len=8192/max_cmd_len=32768/g' > libtool`

`Libtool` unter Windows nimmt ein Zeilenlängenlimit von 8192 an. Dadurch schlägt das Linking fehl. Dieser Fix ist lediglich ein Workaround.

`Make` und `make install` sollten nun problemlos laufen.

### F.2.3 GRASS

Holen Sie sich die Quellen aus dem CVS/SVN oder nutzen Sie einen wöchentlichen Snapshot, siehe:

<http://grass.itc.it/devel/cvs.php>

In der MSYS-Konsole wechseln Sie in das Quellverzeichnis wo Sie die Quellen entpackt oder ausgecheckt haben (z.B. `c:\msys\local\src\grass-6.3.cvs`)

Lassen Sie die folgenden Kommandos laufen:

```
export PATH='/usr/local/bin:/usr/local/lib:$PATH'
./configure --prefix=/usr/local --bindir=/usr/local
--with-includes=/usr/local/include --with-libs=/usr/local/lib --with-cxx
--without-jpeg --without-tiff --without-postgres --with-opengl=windows
--with-fftw --with-freetype --with-freetype-includes=/usr/local/include/freetype2
--without-x --without-tcltk --enable-x11=no --enable-shared=yes
--with-proj-share=/usr/local/share/proj
make
make install
```

GRASS sollte unter `c:\msys\local\grass-6.3.cvs` installiert worden sein.

Diese beiden Links könnten bei der Kompilierung behilflich sein:

- [http://grass.gdf-hannover.de/wiki/WinGRASS\\_Current\\_Status](http://grass.gdf-hannover.de/wiki/WinGRASS_Current_Status)
- <http://geni.ath.cx/grass.html>

### F.2.4 GDAL, zweiter Schritt

An dieser Stelle werden wir die vormalig benutzten GDAL-Quellen erneut benutzen und mit einer leicht geänderten Konfiguration kompilieren.

Vorher müssen die GDAL-Quellen noch repariert werden, damit die GRASS-Unterstützung auch eingebaut werden kann:

[http://trac.osgeo.org/gdal/attachment/ticket/1587/plugin\\_patch\\_grass63.diff](http://trac.osgeo.org/gdal/attachment/ticket/1587/plugin_patch_grass63.diff)

(Sie können es mit der Hand patchen oder `patch.exe` in `c:\msys\bin` benutzen)

Nun wechseln Sie in der MSYS-Konsole in den GDAL-Quellbaum und führen die gleichen Kommandos wie in Schritt eins aus, mit der folgenden Ausnahme:

1. beim `./configure` fügen Sie dieses Argument hinzu: `-with-grass=/usr/local/grass-6.3.cvs`
2. beim Aufruf von `g++` in Zeile 5 des Makefiles (die `libgdal.dll` erzeugt), fügen Sie folgende Argumente hinzu: `-L/usr/local/grass-6.3.cvs/lib -lgrass_vect -lgrass_dig2 -lgrass_dgl -lgrass_rtree -lgrass_linkm -lgrass_dbmiclient -lgrass_dbmibase -lgrass_I -lgrass_gproj -lgrass_vask -lgrass_gmath -lgrass_gis -lgrass_datetime`

Editieren Sie nun `gdal-config` und ändern die Zeile mit `CONFIG_LIBS` zu

```
CONFIG_LIBS='-L/usr/local/lib -lpng -L/usr/local/grass-6.3.cvs/lib
-lgrass_vect -lgrass_dig2 -lgrass_dgl -lgrass_rtree -lgrass_linkm
```

```
-lgrass_dbmiclient -lgrass_dbmibase -lgrass_I -lgrass_gproj -lgrass_vask  
-lgrass_gmath -lgrass_gis -lgrass_datetime -lz -L/usr/local/lib -lgdal'
```

Nun sollte GDAL mit GRASS Rasterdaten arbeiten.

### F.2.5 GEOS

Laden Sie die Quellen herunter:

<http://geos.refractory.net/geos-2.2.3.tar.bz2>

Entpacken Sie diese unter `c:\msys\local\src`

Zur Kompilierung musste ich die Datei `source/headers/timeval.h` in Zeile 13 patchen. Von:

```
#ifdef _WIN32
```

nach:

```
#if defined(_WIN32) && defined(_MSC_VER)
```

Nun kann in der MSYS-Konsole der Bauvorhang angestoßen werden:

```
./configure --prefix=/usr/local  
make  
make install
```

### F.2.6 SQLITE

Sie können vorkompilierte DLLs benutzen, die Sie hier herunterladen können:

[http://www.sqlite.org/sqlitedll-3\\_3\\_17.zip](http://www.sqlite.org/sqlitedll-3_3_17.zip)

Kopieren Sie `sqlite3.dll` daraus nach `c:\msys\local\lib`

Nun laden Sie folgendes Archiv herunter:

[http://www.sqlite.org/sqlite-source-3\\_3\\_17.zip](http://www.sqlite.org/sqlite-source-3_3_17.zip)

Kopieren Sie daraus die Datei `sqlite3.h` nach `c:\msys\local\include`

### F.2.7 GSL

Quellen herunterladen von:

<ftp://ftp.gnu.org/gnu/gsl/gsl-1.9.tar.gz>

Entpacken Sie dieses nach `c:\msys\local\src`

und kompilieren Sie in der MSYS-Konsole:

```
./configure  
make  
make install
```

### F.2.8 EXPAT

Quellcode herunterladen:

<http://dfn.dl.sourceforge.net/sourceforge/expat/expat-2.0.0.tar.gz>

Nach `c:\msys\local\src` entpacken und kompilieren:

```
./configure  
make  
make install
```

### F.2.9 POSTGRES

PostgreSQL wird als vorkompiliertes Binärpaket installiert. Nutzen Sie den folgenden Link:

<http://wwwmaster.postgresql.org/download/mirrors-ftp?file=%2Fbinary%2Fv8.2.4%2Fwin32%2Fpostgresql-8.2.4-1-binaries-no-installer.zip>

Kopieren Sie den Inhalt des `pgsql`-Ordners aus dem Archiv nach `c:\msys\local`

## F.3 Aufräumen

Nun ist die MSYS-Umgebung präpariert. Nicht mehr benötigte Dinge in `c:\msys\local\src` können nun gelöscht werden, da sie eine Menge Platz einnehmen und nicht mehr benötigt werden.

---

## G Bauen mit MS Visual Studio

/!\ Dieser Abschnitt beschreibt einen Bauprozess, wo alle Abhängigkeiten selber gebaut werden. Im nachfolgenden Abschnitt (vgl. Abschn. H.2) wird ein einfacherer Weg beschritten; alle Abhängigkeiten sind bereits paketierte. Lediglich Visual Studio Express muss noch eingerichtet werden, um QGIS damit zu kompilieren.

Beachten Sie, dass diese Anleitung nicht GRASS und die Python-Plugins beinhaltet.

### G.1 Visual Studio einrichten

Dieser Abschnitt beschreibt das Einrichten von Visual Studio, um damit QGIS zu bauen.

#### G.1.1 Express Edition

Die freie Express Edition hat leider keine SDK, welche die Header und weitere benötigte Dateien beinhaltet. Diese sind aber nötig, um QGIS zu bauen. Die Plattform SDK kann wie folgt installiert werden:

<http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/>

Danach müssen Sie `<vsinstalldir>\Common7\Tools\vsvars` wie folgt editieren:

Fügen Sie `%PlatformSDKDir%\Include\atl` und `%PlatformSDKDir%\Include\mf c` zum Eintrag `@set INCLUDE`.

Dies wird weitere Header zu Ihrem System INCLUDE-Pfad hinzufügen. Beachten Sie jedoch, dass dies nur funktioniert, sofern Sie den Visual Studio Command-Prompt zum Bauen benutzen. Nahezu alle Abhängigkeiten werden dann gebaut. Weiterhin müssen Sie aus den oben beschriebenen Dateien einige Bibliotheken löschen, die bei Visual Studio Express nicht dabei sind:

<http://www.codeproject.com/wtl/WTLEXPRESS.asp>

#### G.1.2 Alle Editionen

Sie benötigen weiterhin `stdint.h` und `unistd.h`. `unistd.h` ist in der GnuWin32 Version von flex & bison Binärpaketen enthalten (kommt später). `stdint.h` ist hier zu finden:

<http://www.azillionmonkeys.com/qed/pstdint.h>.

Kopieren Sie beide Dateien nach `<vsinstalldir>\VC\include`.

## G.2 Herunterladen/Installieren der Abhängigkeiten

Dieser Abschnitt beschreibt das Herunterladen und Installieren der verschiedenen QGIS-Abhängigkeiten.

### G.2.1 Flex und Bison

Flex und Bison sind Werkzeuge zum Erzeugen von Parsern, sie werden für GRASS und auch für die QGIS Kompilierung benötigt.

Laden Sie die folgenden Pakete herunter und führen Sie den Installer aus:

<http://gnuwin32.sourceforge.net/downlinks/flex.php>

<http://gnuwin32.sourceforge.net/downlinks/bison.php>

### G.2.2 PostgreSQL Unterstützung in Qt

Wenn Sie Qt mit PostgreSQL-Unterstützung bauen wollen, müssen Sie PostgreSQL herunterladen und installieren, damit es später mit Qt gelinkt werden kann.

Laden Sie `.../binary/v8.2.5/win32/postgresql-8.2.5-1.zip` vom PostgreSQL-Spiegel Ihrer Wahl herunter und installieren Sie es.

PostgreSQL wird derzeit mit MinGW gebaut bringt dessen Header und Bibliotheken mit. Die Header können in Visual C++ benutzt werden, während die Bibliotheken nur als DLL und Archiv (.a) ausgeliefert werden und deshalb nicht direkt mit Visual C++ benutzt werden können.

Um nutzbare Bibliotheken zu erstellen, nutzen Sie das sed-Skript (als `mkdef.sed`) im PostgreSQL-lib-Verzeichnis:

```
/Dump of file / {
s/Dump of file \([^\ ]*\)$ /LIBRARY \1/p
a\
EXPORTS
}
/[^\ ]*ordinal hint/,/^[^\ ]*Summary/ {
/^[^\ ]*[0-9]\+/ {
s/^[^\ ]*[0-9]\+ [^\ ]*[0-9A-Fa-f]\+ [^\ ]*[0-9A-Fa-f]
\+ [^\ ]*\([^\ ]*\)=\+ \).*$ / \1/p
}
}
```

Führen Sie es in der Visual Studio C++ Kommandozeile aus (aus dem Programme Menü):

```
cd c:\Program Files\PostgreSQL\8.2\bin
dumpbin /exports ..\bin\libpq.dll | sed -nf ../lib/mkdef.sed >..\lib\libpq.def
cd ../lib
lib /def:libpq.def /machine:x86
```

Sed muss dazu in Ihrem Suchpfad sein (z.B. von cygwin oder msys).

Das war es schon fast. Sie müssen nur noch die include und lib-Pfade zu den Variablen INCLUDE und LIB in vcvars.bat hinzufügen.

### G.2.3 Qt

Bauen Sie Qt wie folgt:

[http://wiki.qgis.org/qgiswiki/Building\\_QT\\_4\\_with\\_Visual\\_C%2B%2B\\_2005](http://wiki.qgis.org/qgiswiki/Building_QT_4_with_Visual_C%2B%2B_2005)

### G.2.4 Proj.4

Holen Sie proj.4 hier:

<http://proj.maptools.org/>

In der Visual Studio Kommandozeile (Umgebungsvariablen müssen dazu korrekt gesetzt sein), führen Sie den folgenden Befehl aus:

```
nmake -f makefile.vc
```

Installieren Sie, indem Sie die Variable PROJ\_DIR auf Ihre Bedürfnisse hin setzen.

```
set PROJ_DIR=c:\lib\proj
```

```
mkdir %PROJ_DIR%\bin
mkdir %PROJ_DIR%\include
mkdir %PROJ_DIR%\lib
```

```
copy src\*.dll %PROJ_DIR%\bin
copy src\*.exe %PROJ_DIR%\bin
copy src\*.h %PROJ_DIR%\include
copy src\*.lib %PROJ_DIR%\lib
```

Eine Batch-Datei können diese Befehle auch fassen.

### G.2.5 GSL

gsl-Quellen hier herunterladen:

<http://david.geldreich.free.fr/downloads/gsl-1.9-windows-sources.zip>

Bauen Sie GLS mit der Datei `gsl.sln`.

### G.2.6 GEOS

Laden Sie GEOS aus dem SVN (svn checkout <http://svn.refrations.net/geos/trunk> geos) herunter. Editieren Sie `geos\source\makefile.vc` wie folgt:

Entkommentieren Sie die Zeilen 333 und 334. Dies erlaubt das Kopieren von `version.h.vc` nach `version.h`.

Entkommentieren Sie die Zeilen 338 und 339.

Benennen Sie `geos_c.h.vc` nach `geos_c.h.in` in den Zeilen 338 und 339 um. Dies erlaubt das Kopieren von `geos_c.h.in` zu `geos_c.h`.

In der Visual Studio Kommandozeile (Umgebungsvariablen müssen dazu korrekt gesetzt sein), führen Sie den folgenden Befehl aus:

```
nmake -f makefile.vc
```

Installieren Sie, indem Sie die Variable `GEOS_DIR` auf Ihre Bedürfnisse hin setzen.

```
set GEOS_DIR='c:\lib\geos'
```

```
mkdir %GEOS_DIR%\include
```

```
mkdir %GEOS_DIR%\lib
```

```
mkdir %GEOS_DIR%\bin
```

```
xcopy /S/Y source\headers\*.h %GEOS_DIR%\include
```

```
copy /Y capi\*.h %GEOS_DIR%\include
```

```
copy /Y source\*.lib %GEOS_DIR%\lib
```

```
copy /Y source\*.dll %GEOS_DIR%\bin
```

Eine Batch-Datei können diese Befehle auch fassen.



### G.2.7 GDAL

Laden Sie GDAL aus dem SVN (svn checkout <https://svn.osgeo.org/gdal/branches/1.4/gdal>) herunter.

Editieren Sie `nmake.opt` auf Ihre Bedürfnisse, diese Datei ist sehr gut dokumentiert.

In der Visual Studio Kommandozeile (Umgebungsvariablen müssen dazu korrekt gesetzt sein), führen Sie den folgenden Befehl aus:

```
nmake -f makefile.vc
```

und

```
nmake -f makefile.vc devinstall
```

### G.2.8 PostGIS

Laden Sie PostGIS und die Windows-Version von PostgreSQL hier herunter:

<http://postgis.refractory.net/download/>

Beachten Sie, dass Sie nicht die PostGIS-Version im PostgreSQL-Installer installieren sollten.

### G.2.9 Expat

Expat gibt es hier:

[http://sourceforge.net/project/showfiles.php?group\\_id=10127](http://sourceforge.net/project/showfiles.php?group_id=10127)

Sie benötigen `expat-win32bin-2.0.1.exe`.

Einfach den Installer ausführen und expat installieren.

### G.2.10 CMake

Laden Sie CMake hier herunter:

<http://www.cmake.org/HTML/Download.html>

Sie benötigen `cmake-<version>-win32-x86.exe`. Einfach ausführen reicht zum Installieren von CMake.

### G.3 QGIS mit CMAKE bauen

QGIS-Quellen aus dem SVN herunterladen (svn co <https://svn.qgis.org/repos/qgis/trunk/qgis> qgis).

erzeugen Sie ein 'build'-Verzeichnis im ausgecheckten QGIS-Ordner. Darin werden alle Dateien während des Bauens abgelegt.

Starten Sie Start→All Programs→CMake→CMake.

Navigieren Sie zum QGIS-Ordner.

Im Feld 'Wo soll das Binär-Programm gebaut werden' wählen Sie den vormalige erzeugten Ordner 'build' aus.

In der 'Cache Value'-Liste füllen Sie die \*\_INCLUDE\_DIR und \*\_LIBRARY Einträge aus.

Klicken Sie den Konfigurieren-Knopf. Auf die Frage, welche Makefiles Sie erzeugen wollen, wählen Sie Visual Studio 8 2005 und klicken OK.

Nun sollte die Konfiguration ohne Fehler komplett sein. Falls doch Fehler auftraten, korrigieren Sie die Pfade zu Header und Bibliotheken; diese sind meist fehlerträchtig. Fehlerhafte Einträge werden in Rot in der Liste dargestellt.

Wenn alles OK und ohne Fehler ist, klicken Sie OK zum erzeugen der Projekt-Dateien.

Mit Visual Studio 2005, öffnen Sie qgis.sln im 'build'-Ordner.

Bauen Sie das ALL\_BUILD Projekt. Dies erzeugt QGIS mit allen Plugins.

Installieren Sie QGIS in dem Sie das INSTALL Projekt bauen. Standardmäßig wird das nach c:\Program Files\qgis<version> installiert. (Dies können Sie ändern, indem Sie die Variable CMAKE\_INSTALL\_PREFIX in CMake entsprechend anpassen).

Ferner müssen Sie alle Abhängigkeits-DLLs in das QGIS-Verzeichnis kopieren oder wahlweise alle Verzeichnisse in Ihrem Pfad (PATH) haben.

## H Mit MSVC Express unter Windows bauen

/!\ Achtung: Bauen mit MSVC ist noch eine Baustelle. Insbesondere die folgenden Punkte funktionieren noch nicht: python, grass, postgis connections.

/!\ Dieser Abschnitt des Dokumentes ist noch nicht freigegeben, daher wird er noch nicht ins Deutsche übersetzt.

Tim Sutton, 2007

## H.1 System preparation

I started with a clean XP install with Service Pack 2 and all patches applied. I have already compiled all the dependencies you need for gdal, expat etc, so this tutorial wont cover compiling those from source too. Since compiling these dependencies was a somewhat painful task I hope my pre-compiled libs will be adequate. If not I suggest you consult the individual projects for specific build documentation and support. Lets go over the process in a nutshell before we begin:

- \* Install XP (I used a Parallels virtual machine)
- \* Install the premade libraries archive I have made for you
- \* Install Visual Studio Express 2005 sp1
- \* Install the Microsoft Platform SDK
- \* Install command line subversion client
- \* Install library dependencies bundle
- \* Install Qt 4.3.2
- \* Check out QGIS sources
- \* Compile QGIS
- \* Create setup.exe installer for QGIS

## H.2 Install the libraries archive

Half of the point of this section of the MSVC setup procedure is to make things as simple as possible for you. To that end I have prepared an archive that includes all dependencies needed to build QGIS except Qt (which we will build further down). Fetch the archive from:

[http://qgis.org/uploadfiles/msvc/qgis\\_msvc\\_deps\\_except\\_qt4.zip](http://qgis.org/uploadfiles/msvc/qgis_msvc_deps_except_qt4.zip)

Create the following directory structure:

```
c:\dev\cpp\
```

And then extract the libraries archive into a subdirectory of the above directory so that you end up with:

```
c:\dev\cpp\qgislibs-release
```

/!\ Note that you are not obliged to use this directory layout, but you should adjust any instructions that follow if you plan to do things differently.

## H.3 Install Visual Studio Express 2005

First thing we need to get is MSVC Express from here:

<http://msdn2.microsoft.com/en-us/express/aa975050.aspx>

The page is really confusing so dont feel bad if you cant actually find the download at first! There are six coloured blocks on the page for the various studio family members (vb / c# / j# etc). Simply choose your language under the 'select your language' combo under the yellow C++ block, and your download will begin. Under internet explorer I had to disable popup blocking for the download to be able to commence.

Once the setup commences you will be prompted with various options. Here is what I chose :

\* Send usage information to Microsoft (No) \* Install options: \* Graphical IDE (Yes) \* Microsoft MSDN Express Edition (No) \* Microsoft SQL Server Express Edition (No) \* Install to folder: C:\Program Files\Microsoft Visual Studio 8\ (default)

It will need to download around 90mb of installation files and reports that the install will consume 554mb of disk space.

## H.4 Install Microsoft Platform SDK2

Go to this page:

<http://msdn2.microsoft.com/en-us/express/aa700755.aspx>

Start by using the link provided on the above page to download and install the platform SDK2.

The actual SDK download page is once again a bit confusing since the links for downloading are hidden amongst a bunch of other links. Basically look for these three links with their associated 'Download' buttons and choose the correct link for your platform:

PSDK-amd64.exe	1.2 MB	Download
PSDK-ia64.exe	1.3 MB	Download
PSDK-x86.exe	1.2 MB	Download

When you install make sure to choose 'custom install'. These instructions assume you are installing into the default path of:

C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\

We will go for the minimal install that will give us a working environment, so on the custom installation screen I made the following choices:

Configuration Options  
+ Register Environmental Variables (Yes)  
Microsoft Windows Core SDK

---

+ Tools	(Yes)
+ Tools (AMD 64 Bit)	(No unless this applies)
+ Tools (Intel 64 Bit)	(No unless this applies)
+ Build Environment	
+ Build Environment (AMD 64 Bit)	(No unless this applies)
+ Build Environment (Intel 64 Bit)	(No unless this applies)
+ Build Environment (x86 32 Bit)	(Yes)
+ Documentation	(No)
+ Redistributable Components	(Yes)
+ Sample Code	(No)
+ Source Code	(No)
+ AMD 64 Source	(No)
+ Intel 64 Source	(No)
Microsoft Web Workshop	(Yes) (needed for shlwapi.h)
+ Build Environment	(Yes)
+ Documentation	(No)
+ Sample Code	(No)
+ Tools	(No)
Microsoft Internet Information Server (IIS) SDK	(No)
Microsoft Data Access Services (MDAC) SDK	(Yes) (needed by GDAL for odbc)
+ Tools	
+ Tools (AMD 64 Bit)	(No)
+ Tools (AMD 64 Bit)	(No)
+ Tools (x86 32 Bit)	(Yes)
+ Build Environment	
+ Tools (AMD 64 Bit)	(No)
+ Tools (AMD 64 Bit)	(No)
+ Tools (x86 32 Bit)	(Yes)
+ Documentation	(No)
+ Sample Code	(No)
Microsoft Installer SDK	(No)
Microsoft Table PC SDK	(No)
Microsoft Windows Management Instrumentation	(No)
Microsoft DirectShow SDK	(No)
Microsoft Media Services SDK	(No)
Debuggin Tools for Windows	(Yes)

/!\ Note that you can always come back later to add extra bits if you like.

/!\ Note that installing the SDK requires validation with the Microsoft Genuine Advantage application. Some people have a philosophical objection to installing this software on their computers. If you are one of them you should probably consider using the MINGW build instructions described elsewhere

in this document.

The SDK installs a directory called

`C:\Office10`

Which you can safely remove.

After the SDK is installed, follow the remaining notes on the page link above to get your MSVC Express environment configured correctly. For your convenience, these are summarised again below, and I have added a couple more paths that I discovered were needed:

- 1) open Visual Studio Express IDE
- 2) Tools -> Options -> Projects and Solutions -> VC++ Directories
- 3) Add:

Executable files:

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin`

Include files:

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include`

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\atl`

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\mfc`

Library files: `C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib`

- 4) Close MSVC Express IDE
- 5) Open the following file with notepad:

`C:\Program Files\Microsoft Visual Studio 8\VC\VCProjectDefaults\corewin_express.vsprops`

and change the property:

`AdditionalDependencies='kernel32.lib'`

To read:

`AdditionalDependencies='kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib  
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib'`

The notes go on to show how to build a mswin32 application which you can try if you like - I'm not going to recover that here.

## H.5 Edit your vsvars

Backup your vsvars32.bat file in

C:\Program Files\Microsoft Visual Studio 8\Common7\Tools

and replace it with this one:

```
@SET VSINSTALLDIR=C:\Program Files\Microsoft Visual Studio 8
@SET VCINSTALLDIR=C:\Program Files\Microsoft Visual Studio 8\VC
@SET FrameworkDir=C:\WINDOWS\Microsoft.NET\Framework
@SET FrameworkVersion=v2.0.50727
@SET FrameworkSDKDir=C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0
@if '%VSINSTALLDIR%'==' ' goto error_no_VSINSTALLDIR
@if '%VCINSTALLDIR%'==' ' goto error_no_VCINSTALLDIR

@echo Setting environment for using Microsoft Visual Studio 2005 x86 tools.

@rem
@rem Root of Visual Studio IDE installed files.
@rem
@set DevEnvDir=C:\Program Files\Microsoft Visual Studio 8\Common7\IDE

@set PATH=C:\Program Files\Microsoft Visual Studio 8\Common7\IDE;
C:\Program Files\Microsoft Visual Studio 8\VC\BIN;
C:\Program Files\Microsoft Visual Studio 8\Common7\Tools;
C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\bin;
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727;
C:\Program Files\Microsoft Visual Studio 8\VC\VCPackages;%PATH%
@rem added by Tim
@set PATH=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Bin;%PATH%
@set INCLUDE=C:\Program Files\Microsoft Visual Studio 8\VC\INCLUDE;%INCLUDE%
@rem added by Tim
@set INCLUDE=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Include;%INCLUDE%
@set INCLUDE=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Include\mfc;%INCLUDE%
@set INCLUDE=%INCLUDE%;C:\dev\cpp\qgislibs-release\include\postgresql
@set LIB=C:\Program Files\Microsoft Visual Studio 8\VC\LIB;
C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\lib;%LIB%
```

```
@rem added by Tim
@set LIB=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
      R2\Lib;%LIB%
@set LIB=%LIB%;C:\dev\cpp\qgislibs-release\lib
@set LIBPATH=C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727

@goto end

:error_no_VSINSTALLDIR
@echo ERROR: VSINSTALLDIR variable is not set.
@goto end

:error_no_VCINSTALLDIR
@echo ERROR: VCINSTALLDIR variable is not set.
@goto end

:end
```

## H.6 Environment Variables

Right click on 'My computer' then select the 'Advanced' tab. Click environment variables and create or augment the following "System" variables (if they dont already exist):

Variable Name:	Value:
EDITOR	vim
INCLUDE	C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\.
LIB	C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib\.
LIB_DIR	C:\dev\cpp\qgislibs-release
PATH	C:\Program Files\CMake 2.4\bin; %SystemRoot%\system32; %SystemRoot%; %SystemRoot%\System32\Wbem; C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin\.; C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin\WinNT\; C:\Program Files\svn\bin;C:\Program Files\Microsoft Visual



```
Studio 8\VC\bin;  
C:\Program Files\Microsoft Visual Studio 8\Common7\IDE;  
'c:\Program Files\Microsoft Visual Studio 8\Common7\Tools';  
c:\Qt\4.3.2\bin;  
'C:\Program Files\PuTTY'  
QTDIR      c:\Qt\4.3.2  
SVN_SSH    'C:\\Program Files\\PuTTY\\plink.exe'
```

== Building Qt4.3.2 ==

You need a minimum of Qt 4.3.2 here since this is the first version to officially support building the open source version of Qt for windows under MSVC.

Download Qt 4.x.x source for windows from

<http://www.trolltech.com>

Unpack the source to

c:\Qt\4.x.x\

=== Compile Qt ===

Open the Visual Studio C++ command line and cd to c:\Qt\4.x.x where you extracted the source and enter:

```
configure -platform win32-msvc2005  
nmake  
nmake install
```

Add -qt-sql-odbc -qt-sql-psql to the configure line if your want odbc and PostgreSQL support build into Qt.

/!\ Note: For me in some cases I got a build error on qsscreenshot.pro. If you are only interested in having the libraries needed for building Qt apps, you can probably ignore that. Just check in c:\Qt\4.3.2\bin to check all dlls and helper apps (assistant etc) have been made.

=== Configure Visual C++ to use Qt ===

After building configure the Visual Studio Express IDE to use Qt:

- 1) open Visual Studio Express IDE
- 2) Tools -> Options -> Projects and Solutions -> VC++ Directories
- 3) Add:

Executable files: \$(QTDIR)\bin

Include files: \$(QTDIR)\include \$(QTDIR)\include\Qt \$(QTDIR)\include\QtCore  
\$(QTDIR)\include\QtGui \$(QTDIR)\include\QtNetwork \$(QTDIR)\include\QtSvg  
\$(QTDIR)\include\QtXml \$(QTDIR)\include\Qt3Support \$(LIB\_DIR)\include (needed during  
qgis compile to findstdint.h andunistd.h)

Library files: \$(QTDIR)\lib

Source Files: \$(QTDIR)\src

Hint: You can also add

QString = t=<d->data, su>, size=<d->size, i>

to AutoExp.DAT in C:\Program Files\Microsoft Visual Studio 8  
\Common7\Packages\Debugger before

[Visualizer]

That way the Debugger will show the contents of QString when you point at or watch a variable in the debugger. There are probably much more additions - feel free to add some - I just needed QString and took the first hit in google I could find.

== Install Python ==

Download <http://python.org/ftp/python/2.5.1/python-2.5.1.msi> and install it.

== Install SIP ==

Download <http://www.riverbankcomputing.com/Downloads/sip4/sip-4.7.1.zip>

and extract it into your c:\dev\cpp directory.

From a Visual C++ command line cd to the directory where you extract SIP and run:

```
c:\python25\python configure.py -p win32-msvc2005 nmake nmake install
```

== Install PyQt4 ==

Download <http://www.riverbankcomputing.com/Downloads/PyQt4/GPL/PyQt-win-gpl-4.3.1.zip>

and extract it into your c:\dev\cpp directory.

From a Visual C++ command line cd to the directory where you extracted PyQt4 and run:

```
c:\python25\python configure.py -p win32-msvc2005 nmake nmake install ""
```

## H.7 Install CMake

Download and install cmake 2.4.7 or better, making sure to enable the option

Update path for all users

## H.8 Install Subversion

You "must" install the command line version if you want the CMake svn scripts to work. Its a bit tricky to find the correct version on the subversion download site as they have som misleadingly named similar downloads. Easiest is to just get this file:

<http://subversion.tigris.org/downloads/1.4.5-win32/apache-2.2/svn-win32-1.4.5.zip>

Extract the zip file to

C:\Program Files\svn

And then add

C:\Program Files\svn\bin

To your path.

## H.9 Initial SVN Check out

Open a cmd.exe window and do:

```
cd \  
cd dev  
cd cpp  
svn co https://svn.qgis.org/repos/qgis/trunk/qgis
```

At this point you will probably get a message like this:

```
C:\dev\cpp>svn co https://svn.qgis.org/repos/qgis/trunk/qgis  
Error validating server certificate for 'https://svn.qgis.org:443':  
- The certificate is not issued by a trusted authority. Use the  
  fingerprint to validate the certificate manually!  
Certificate information:  
- Hostname: svn.qgis.org  
- Valid: from Sat, 01 Apr 2006 03:30:47 GMT until Fri, 21 Mar 2008 03:30:47 GMT  
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US  
- Fingerprint: 2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b  
(R)eject, accept (t)emporarily or accept (p)ermanently?
```

Press 'p' to accept and the svn checkout will commence.

## H.10 Create Makefiles using cmakesetup.exe

I won't be giving a detailed description of the build process, because the process is explained in the first section (where you manually build all dependencies) of the windows build notes in this document. Just skip past the parts where you need to build GDAL etc, since this simplified install process does all the dependency provisioning for you.

```
cd qgis
mkdir build
cd build
cmakesetup ..
```

Cmakesetup should find all dependencies for you automatically (it uses the LIB\_DIR environment to find them all in c:\dev\cpp\qgislibs-release). Press configure again after the cmakesetup gui appears and when all the red fields are gone, and you have made any personalisations to the setup, press ok to close the cmake gui.

Now open Visual Studio Express and do:

File -> Open -> Project / Solution

Now open the cmake generated QGIS solution which should be in :

```
c:\dev\cpp\qgis\build\qgisX.X.X.sln
```

Where X.X.X represents the current version number of QGIS. Currently I have only made release built dependencies for QGIS (debug versions will follow in future), so you need to be sure to select 'Release' from the solution configurations toolbar.

Next right click on ALL\_BUILD in the solution browser, and then choose build.

Once the build completes right click on INSTALL in the solution browser and choose build. This will by default install qgis into c:\program files\qgisX.X.X.

## H.11 Running and packaging

To run QGIS you need to at the minimum copy the dlls from c:\dev\cpp\qgislibs-release\bin into the c:\program files\qgisX.X.X directory.

---

# I GNU Public License

## GNU GENERAL PUBLIC LICENSE

Deutsche Übersetzung der Version 2, Juni 1991. Den offiziellen englischen Originaltext finden Sie unter <http://www.gnu.org/licenses/gpl.html>. Diese Übersetzung wird mit der Absicht angeboten, das Verständnis der GNU General Public License (GNU GPL) zu erleichtern. Es handelt sich jedoch nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.

Diese Übersetzung wurde ursprünglich erstellt von Katja Lachmann Übersetzungen im Auftrag der S.u.S.E. GmbH - <http://www.suse.de>. Sie wurde überarbeitet von Peter Gerwinski, G-N-U GmbH - <http://www.g-n-u.de> (31. Oktober 1996, 4. Juni 2000).

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Es ist jedermann gestattet, diese Lizenzurkunde zu vervielfältigen und unveränderte Kopien zu verbreiten; Änderungen sind jedoch nicht erlaubt.

### Vorwort

Die meisten Softwarelizenzen sind daraufhin entworfen worden, Ihnen die Freiheit zu nehmen, die Software weiterzugeben und zu verändern. Im Gegensatz dazu soll Ihnen die GNU General Public License, die Allgemeine Öffentliche GNU-Lizenz, ebendiese Freiheit garantieren. Sie soll sicherstellen, daß die Software für alle Benutzer frei ist. Diese Lizenz gilt für den Großteil der von der Free Software Foundation herausgegebenen Software und für alle anderen Programme, deren Autoren ihr Werk dieser Lizenz unterstellt haben. Auch Sie können diese Möglichkeit der Lizenzierung für Ihre Programme anwenden. (Ein anderer Teil der Software der Free Software Foundation unterliegt stattdessen der GNU Lesser General Public License, der Kleineren Allgemeinen Öffentlichen GNU-Lizenz.)

Die Bezeichnung "freie" Software bezieht sich auf Freiheit, nicht auf den Preis. Unsere Lizenzen sollen Ihnen die Freiheit garantieren, Kopien freier Software zu verbreiten (und etwas für diesen Service zu berechnen, wenn Sie möchten), die Möglichkeit, die Software im Quelltext zu erhalten oder den Quelltext auf Wunsch zu bekommen. Die Lizenzen sollen garantieren, daß Sie die Software ändern oder Teile davon in neuen freien Programmen verwenden dürfen - und daß Sie wissen, daß Sie dies alles tun dürfen.

Um Ihre Rechte zu schützen, müssen wir Einschränkungen machen, die es jedem verbieten, Ihnen diese Rechte zu verweigern oder Sie aufzufordern, auf diese Rechte zu verzichten. Aus diesen Einschränkungen folgen bestimmte Verantwortlichkeiten für Sie, wenn Sie Kopien der Software verbreiten oder sie verändern.

Beispielsweise müssen Sie den Empfängern alle Rechte gewähren, die Sie selbst haben, wenn Sie - kostenlos oder gegen Bezahlung - Kopien eines solchen Programms verbreiten. Sie müssen sicherstellen, daß auch die Empfänger den Quelltext erhalten bzw. erhalten können. Und Sie müssen ihnen diese Bedingungen zeigen, damit sie ihre Rechte kennen.

Wir schützen Ihre Rechte in zwei Schritten: (1) Wir stellen die Software unter ein Urheberrecht (Copyright), und (2) wir bieten Ihnen diese Lizenz an, die Ihnen das Recht gibt, die Software zu vervielfältigen, zu verbreiten und/oder zu verändern.

---

Um die Autoren und uns zu schützen, wollen wir darüberhinaus sicherstellen, daß jeder erfährt, daß für diese freie Software keinerlei Garantie besteht. Wenn die Software von jemand anderem modifiziert und weitergegeben wird, möchten wir, daß die Empfänger wissen, daß sie nicht das Original erhalten haben, damit irgendwelche von anderen verursachte Probleme nicht den Ruf des ursprünglichen Autors schädigen.

Schließlich und endlich ist jedes freie Programm permanent durch Software-Patente bedroht. Wir möchten die Gefahr ausschließen, daß Distributoren eines freien Programms individuell Patente lizenzieren - mit dem Ergebnis, daß das Programm proprietär würde. Um dies zu verhindern, haben wir klargestellt, daß jedes Patent entweder für freie Benutzung durch jedermann lizenziert werden muß oder überhaupt nicht lizenziert werden darf.

Es folgen die genauen Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung:

## **Allgemeine Öffentliche GNU-Lizenz**

### **Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung**

§0. Diese Lizenz gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Copyright-Inhabers darauf hinweist, daß das Werk unter den Bestimmungen dieser General Public License verbreitet werden darf. Im folgenden wird jedes derartige Programm oder Werk als "das Programm" bezeichnet; die Formulierung "auf dem Programm basierendes Werk" bezeichnet das Programm sowie jegliche Bearbeitung des Programms im urheberrechtlichen Sinne, also ein Werk, welches das Programm, auch auszugsweise, sei es unverändert oder verändert und/oder in eine andere Sprache übersetzt, enthält. (Im folgenden wird die Übersetzung ohne Einschränkung als "Bearbeitung" eingestuft.) Jeder Lizenznehmer wird im folgenden als "Sie" angesprochen.

Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von dieser Lizenz nicht berührt; sie fallen nicht in ihren Anwendungsbereich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt, und die Ausgaben des Programms unterliegen dieser Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, daß die Ausgabe durch die Ausführung des Programmes erfolgte). Ob dies zutrifft, hängt von den Funktionen des Programms ab.

§1. Sie dürfen auf beliebigen Medien unveränderte Kopien des Quelltextes des Programms, wie sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, daß Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschluß veröffentlichen, alle Vermerke, die sich auf diese Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen und desweiteren allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie dieser Lizenz zukommen lassen.

Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen. Wenn Sie es wünschen, dürfen Sie auch gegen Entgelt eine Garantie für das Programm anbieten.

§2. Sie dürfen Ihre Kopie(n) des Programms oder eines Teils davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen von Paragraph 1 vervielfältigen und verbreiten, vorausgesetzt, daß zusätzlich alle im folgenden genannten Bedingungen erfüllt werden:

1. Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.
2. Sie müssen dafür sorgen, daß jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den

---

Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.

3. Wenn das veränderte Programm normalerweise bei der Ausführung interaktiv Kommandos einliest, müssen Sie dafür sorgen, daß es, wenn es auf dem üblichsten Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdruckt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, daß es keine Gewährleistung gibt (oder anderenfalls, daß Sie Garantie leisten), und daß die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muß der Benutzer darauf hingewiesen werden, wie er eine Kopie dieser Lizenz ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muß Ihr auf dem Programm basierendes Werk auch keine solche Meldung ausgeben).

Diese Anforderungen gelten für das bearbeitete Werk als Ganzes. Wenn identifizierbare Teile des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise als unabhängige und eigenständige Werke für sich selbst zu betrachten sind, dann gelten diese Lizenz und ihre Bedingungen nicht für die betroffenen Teile, wenn Sie diese als eigenständige Werke weitergeben. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen weitergeben, das ein auf dem Programm basierendes Werk darstellt, dann muß die Weitergabe des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf das gesamte Ganze ausgedehnt werden - und somit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor.

Somit ist es nicht die Absicht dieses Abschnittes, Rechte für Werke in Anspruch zu nehmen oder Ihnen die Rechte für Werke streitig zu machen, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht, die Rechte zur Kontrolle der Verbreitung von Werken, die auf dem Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben.

Ferner bringt auch das einfache Zusammenlegen eines anderen Werkes, das nicht auf dem Programm basiert, mit dem Programm oder einem auf dem Programm basierenden Werk auf ein- und demselben Speicher- oder Vertriebsmedium dieses andere Werk nicht in den Anwendungsbereich dieser Lizenz.

§3. Sie dürfen das Programm (oder ein darauf basierendes Werk gemäß Paragraph 2) als Objectcode oder in ausführbarer Form unter den Bedingungen der Paragraphen 1 und 2 kopieren und weitergeben - vorausgesetzt, daß Sie außerdem eine der folgenden Leistungen erbringen:

1. Liefern Sie das Programm zusammen mit dem vollständigen zugehörigen maschinenlesbaren Quelltext auf einem für den Datenaustausch üblichen Medium aus, wobei die Verteilung unter den Bedingungen der Paragraphen 1 und 2 erfolgen muß. Oder:
2. Liefern Sie das Programm zusammen mit einem mindestens drei Jahre lang gültigen schriftlichen Angebot aus, jedem Dritten eine vollständige maschinenlesbare Kopie des Quelltextes zur Verfügung zu stellen - zu nicht höheren Kosten als denen, die durch den physikalischen Kopiervorgang anfallen -, wobei der Quelltext unter den Bedingungen der Paragraphen 1 und 2 auf einem für den Datenaustausch üblichen Medium weitergegeben wird.  
Oder:
3. Liefern Sie das Programm zusammen mit dem schriftlichen Angebot der Zurverfügungstellung des Quelltextes aus, das Sie selbst erhalten haben. (Diese Alternative ist nur für nicht-kommerzielle Verbreitung zulässig und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot gemäß Absatz b erhalten haben.)

Unter dem Quelltext eines Werkes wird diejenige Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet "der komplette Quelltext": Der Quelltext aller im Programm enthaltenen Module einschließlich aller zugehörigen Modulschnittstellen-Definitionsdateien



---

sowie der zur Compilation und Installation verwendeten Skripte. Als besondere Ausnahme jedoch braucht der verteilte Quelltext nichts von dem zu enthalten, was üblicherweise (entweder als Quelltext oder in binärer Form) zusammen mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) geliefert wird, unter dem das Programm läuft - es sei denn, diese Komponente selbst gehört zum ausführbaren Programm.

Wenn die Verbreitung eines ausführbaren Programms oder von Objectcode dadurch erfolgt, daß der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quelltext als Verbreitung des Quelltextes, auch wenn Dritte nicht dazu gezwungen sind, den Quelltext zusammen mit dem Objectcode zu kopieren.

§4. Sie dürfen das Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es nicht durch diese Lizenz ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter dieser Lizenz. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter dieser Lizenz erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.

§5. Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da Sie sie nicht unterzeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie diese Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr Einverständnis mit dieser Lizenz und mit allen ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werks.

§6. Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den hier festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen der Durchsetzung der hierin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung dieser Lizenz durch Dritte durchzusetzen.

§7. Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverletzung oder aus einem anderen Grunde (nicht auf Patentfragen begrenzt) Bedingungen (durch Gerichtsbeschluß, Vergleich oder anderweitig) auferlegt werden, die den Bedingungen dieser Lizenz widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen dieser Lizenz. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in dieser Lizenz und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann dürfen Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die gebührenfreie Weiterverbreitung des Programms durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, sowohl das Patentrecht als auch diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten.

Sollte sich ein Teil dieses Paragraphen als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Paragraph seinem Sinne nach angewandt werden; im übrigen soll dieser Paragraph als Ganzes gelten.

Zweck dieses Paragraphen ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Paragraph hat einzig den Zweck, die Integrität des Vertriebssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zu dem großen Angebot der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet; es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluß.

---

Dieser Paragraph ist dazu gedacht, deutlich klarzustellen, was als Konsequenz aus dem Rest dieser Lizenz betrachtet wird.

§8. Wenn die Verbreitung und/oder die Benutzung des Programms in bestimmten Staaten entweder durch Patente oder durch urheberrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter diese Lizenz gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, in der diese Staaten ausgeschlossen werden, so daß die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet diese Lizenz die Beschränkung, als wäre sie in diesem Text niedergeschrieben.

§9. Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, können aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden.

Jede Version dieser Lizenz hat eine eindeutige Versionsnummer. Wenn in einem Programm angegeben wird, daß es dieser Lizenz in einer bestimmten Versionsnummer oder "jeder späteren Version" ("any later version" 201d) unterliegt, so haben Sie die Wahl, entweder den Bestimmungen der genannten Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

§10. Wenn Sie den Wunsch haben, Teile des Programms in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; wir machen zu diesem Zweck gelegentlich Ausnahmen. Unsere Entscheidung wird von den beiden Zielen geleitet werden, zum einen den freien Status aller von unserer freien Software abgeleiteten Werke zu erhalten und zum anderen das gemeinschaftliche Nutzen und Wiederverwenden von Software im allgemeinen zu fördern.

### **Keine Gewährleistung**

§11. Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, "wie es ist", ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich - aber nicht begrenzt auf - Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

§12. In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich - aber nicht beschränkt auf - Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

## I.1 Quantum GIS Qt Ausnahme der GPL

### Original (Englisch):

In addition, as a special exception, the QGIS Development Team gives permission to link the code of this program with the Qt library, including but not limited to the following versions (both free and commercial): Qt/Non-commercial Windows, Qt/Windows, Qt/X11, Qt/Mac, and Qt/Embedded (or with modified versions of Qt that use the same license as Qt), and distribute linked combinations including the two. You must obey the GNU General Public License in all respects for all of the code used other than Qt. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

### Übersetzung (Deutsch):

Zusätzlich, als eine besondere Ausnahme, gibt das QGIS Development Team das Recht, den Quellcode dieses Programms mit der Qt-Bibliothek, einschliesslich aber nicht begrenzt auf die folgenden Versionen (frei und kommerziell): Qt/Non-commercial Windows, Qt/Windows, Qt/X11, Qt/Mac, und Qt/Embedded (oder mit modifizierten Qt-Versionen, welche die gleiche Lizenz wie Qt nutzen) zu verlinken und zu vertreiben. Für jeglichen Quellcode ausser dem Qt-Quellcode, müssen Sie die GNU General Public License in jeder Hinsicht befolgen. Wenn Sie diesen Text ändern, können Sie diese Ausnahmen erweitern zu Ihrer Version dieses Textes, aber Sie sind dazu nicht verpflichtet. Wenn Sie es nicht möchten, löschen Sie diese Ausnahme aus Ihrer Version.

## **Literatur**

- [1] T. Mitchell. Web mapping illustrated, published by o'reilly, 2005.
- [2] G. Sherman. Shuffling quantum gis into the open source gis stack, free and open source software for geospatial (foss4g) conference victoria, bc, 2007.

## **Web-Referenzen**

- [3] GRASS GIS. <http://grass.itc.it>, 2006.
- [4] PostGIS. <http://postgis.refrations.net/>, 2006.
- [5] Web Map Service (1.1.1) Implementation Specification. <http://portal.opengeospatial.org>, 2002.
- [6] Web Map Service (1.3.0) Implementation Specification. <http://portal.opengeospatial.org>, 2004.