

---

# **QGIS User Guide**

*Release 2.6*

**QGIS Project**

22.05.2015



<b>1</b>	<b>Präambel</b>	<b>3</b>
<b>2</b>	<b>Gebrauch der Dokumentation</b>	<b>5</b>
2.1	GUI Schreibstile . . . . .	5
2.2	Text oder Tastatur Schreibstile . . . . .	5
2.3	Betriebssystemspezifische Anweisungen . . . . .	6
<b>3</b>	<b>Vorwort</b>	<b>7</b>
<b>4</b>	<b>Funktionalitäten</b>	<b>9</b>
4.1	Daten visualisieren . . . . .	9
4.2	Daten erkunden, abfragen und Karten layouts . . . . .	9
4.3	Daten erstellen, editieren, verwalten und exportieren . . . . .	10
4.4	Daten analysieren . . . . .	10
4.5	Karten im Internet veröffentlichen . . . . .	10
4.6	Erweiterte QGIS Funktionalität durch Erweiterungen . . . . .	10
4.7	Python-Konsole . . . . .	11
4.8	Bekannte Probleme . . . . .	12
<b>5</b>	<b>Was ist neu in QGIS 2.6</b>	<b>13</b>
5.1	Anwendungen und Projektoptionen . . . . .	13
5.2	Datenprovider . . . . .	13
5.3	Kartenzusammenstellung . . . . .	13
5.4	QGIS Server . . . . .	14
5.5	Darstellung . . . . .	14
5.6	Benutzeroberfläche . . . . .	14
<b>6</b>	<b>Der erste Einstieg</b>	<b>15</b>
6.1	Installation . . . . .	15
6.2	Beispieldaten . . . . .	15
6.3	Ein erstes Übungsbeispiel . . . . .	16
6.4	QGIS Starten und Beenden . . . . .	17
6.5	Optionen der Kommandozeile . . . . .	17
6.6	QGIS Projekte . . . . .	19
6.7	Ausgabe . . . . .	20
<b>7</b>	<b>QGIS GUI</b>	<b>21</b>
7.1	Menüleiste . . . . .	22
7.2	Werkzeuggeste . . . . .	28
7.3	Legende . . . . .	29
7.4	Kartenfenster . . . . .	31
7.5	Statusleiste . . . . .	32

<b>8</b>	<b>Allgemeine Werkzeuge</b>	<b>33</b>
8.1	Tastenkürzel . . . . .	33
8.2	Hilfe . . . . .	33
8.3	Layeranzeige kontrollieren . . . . .	34
8.4	Messen . . . . .	35
8.5	Objekte abfragen . . . . .	37
8.6	Dekorationen . . . . .	39
8.7	Beschriftungstools . . . . .	42
8.8	Räumliche Lesezeichen . . . . .	44
8.9	Layer/Gruppen einbinden . . . . .	44
<b>9</b>	<b>QGIS Konfiguration</b>	<b>47</b>
9.1	Bedienfelder und Werkzeugkästen . . . . .	47
9.2	Projekteigenschaften . . . . .	47
9.3	Optionen . . . . .	49
9.4	Anpassung . . . . .	57
<b>10</b>	<b>Arbeiten mit Projektionen</b>	<b>59</b>
10.1	Überblick zur Projektionsunterstützung . . . . .	59
10.2	Bestimmung einer globalen Projektion . . . . .	59
10.3	On-The-Fly (OTF) Projektion . . . . .	61
10.4	Eigenes Koordinatenbezugssystem definieren . . . . .	62
10.5	Standard Datumtransformationen . . . . .	64
<b>11</b>	<b>QGIS Browser</b>	<b>65</b>
<b>12</b>	<b>Arbeiten mit Vektordaten</b>	<b>67</b>
12.1	Unterstützte Datenformate . . . . .	67
12.2	Die Symbolbibliothek . . . . .	81
12.3	Vektorlayereigenschaften . . . . .	84
12.4	Ausdrücke . . . . .	115
12.5	Editierfunktionen . . . . .	122
12.6	Abfrageeditor . . . . .	141
12.7	Feldrechner . . . . .	143
<b>13</b>	<b>Arbeiten mit Rasterdaten</b>	<b>147</b>
13.1	Arbeiten mit Rasterdaten . . . . .	147
13.2	Dialogfenster Rasterlayereigenschaften . . . . .	148
13.3	Rasterrechner . . . . .	156
<b>14</b>	<b>Arbeiten mit OGC Daten</b>	<b>161</b>
14.1	QGIS als OGC Datenclient . . . . .	161
14.2	QGIS als OGC Datenserver . . . . .	170
<b>15</b>	<b>Arbeiten mit GPS Daten</b>	<b>177</b>
15.1	GPS Plugin . . . . .	177
15.2	Live GPS tracking . . . . .	181
<b>16</b>	<b>GRASS GIS Integration</b>	<b>187</b>
16.1	GRASS Plugin starten . . . . .	187
16.2	GRASS Layer visualisieren . . . . .	188
16.3	Information zur GRASS-Datenbank . . . . .	188
16.4	Daten in eine GRASS LOCATION importieren . . . . .	191
16.5	Das GRASS Vektormodell . . . . .	192
16.6	Einen neuen GRASS Vektorlayer erstellen . . . . .	193
16.7	Digitalisieren und Editieren eines GRASS Vektorlayers . . . . .	193
16.8	Einstellung der GRASS Region . . . . .	195
16.9	Die GRASS Werkzeugkiste . . . . .	197



<b>17 QGIS Verarbeitung Umgebung</b>	<b>209</b>
17.1 Einführung	209
17.2 Die Werkzeugkiste	209
17.3 Die Grafische Modellierung	221
17.4 Die Stapelprozeß Schnittstelle	227
17.5 Verarbeitung Algorithmen von der Konsole aus verwenden	230
17.6 Das Protokoll	235
17.7 Writing new Processing algorithms as python scripts	237
17.8 Handing data produced by the algorithm	239
17.9 Communicating with the user	239
17.10 Documenting your scripts	239
17.11 Beispielskripte	239
17.12 Best practices for writing script algorithms	240
17.13 Pre- and post-execution script hooks	240
17.14 Konfiguration externer Anwendungen	240
17.15 The QGIS Commander	247
<b>18 Processing providers and algorithms</b>	<b>251</b>
18.1 GDAL algorithm provider	251
18.2 LAStools	284
18.3 Modeler Tools	309
18.4 OrfeoToolbox algorithm provider	311
18.5 QGIS algorithm provider	386
18.6 R algorithm provider	440
18.7 SAGA algorithm provider	450
18.8 TauDEM algorithm provider	621
<b>19 Druckzusammenstellung</b>	<b>653</b>
19.1 Erste Schritte	654
19.2 Darstellung	659
19.3 Zusammenstellungselemente	660
19.4 Elemente verwalten	689
19.5 Schritte rückgängig machen und wiederherstellen	691
19.6 Atlas-Erzeugung	691
19.7 Eine Ausgabe erzeugen	694
19.8 Die Druckzusammenstellung verwalten	695
<b>20 Erweiterungen</b>	<b>697</b>
20.1 QGIS Erweiterungen	697
20.2 QGIS Kernplugins verwenden	702
20.3 Koordinaten aufnehmen Plugin	702
20.4 DB Manager Plugin	703
20.5 Dxf2Shape Konverter Plugin	704
20.6 eVis Plugin	705
20.7 fTools Plugin	717
20.8 GDALTools Plugin	721
20.9 Georeferenzier Plugin	724
20.10 Interpolationsplugin	730
20.11 Offline-Bearbeitung Plugin	731
20.12 Oracle-Spatial-GeoRaster Plugin	731
20.13 Rastergeländeanalyse-Erweiterung	735
20.14 Heatmap-Erweiterung	736
20.15 MetaSearch Katalog Client	741
20.16 Straßengraph Plugin	744
20.17 Räumliche Abfrage Plugin	747
20.18 SPIT Plugin	747
20.19 SQL-Anywhere Plugin	748
20.20 Topologieprüfung Erweiterung	748
20.21 Zonenstatistikerweiterung	752

<b>21 Hilfe und Support</b>	<b>753</b>
21.1 Mailinglisten . . . . .	753
21.2 IRC . . . . .	754
21.3 BugTracker . . . . .	754
21.4 Blog . . . . .	755
21.5 Plugins . . . . .	755
21.6 Wiki . . . . .	755
<b>22 Anhang</b>	<b>757</b>
22.1 GNU General Public License . . . . .	757
22.2 GNU Free Documentation License . . . . .	761
<b>23 Literatur und Internetreferenzen</b>	<b>769</b>
<b>Stichwortverzeichnis</b>	<b>771</b>

·  
·



---

## Präambel

---

Dieses Werk ist das offizielle Handbuch zur Benutzung und Installation der Software QGIS. Die in diesem Werk genannten Soft- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen. QGIS 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 ist mit reStructuredText gesetzt. Es ist als reST Quellcode erhältlich unter github und kann online als HTML und PDF angeschaut bzw. heruntergeladen werden. Weitere Informationen darüber wie Sie an diesem Dokument mitwirken und übersetzen können finden Sie auf <http://www.qgis.org/wiki/>.

### Verweise in diesem Dokument

Das Dokument enthält interne und externe Verweise. Wenn Sie auf einen internen Verweis klicken dann springen Sie innerhalb des Dokuments währenddessen sich wenn Sie auf einen externen Verweis klicken eine Internetadresse öffnet. Im PDF sind interne Verweise blau und externe Verweise grün dargestellt. Klicken Sie auf einen grünen Verweis dann wird mit Ihrem Webbrowser eine Seite im Internet geöffnet. In der HTML Version sind die Farben der Verweise identisch.

### Autoren des englischsprachigen User Guides:

Tara Athan	Radim Blazek	Godofredo Contreras	Otto Dassau	Martin Dobias
Peter Ersts	Anne Ghisla	Stephan Holl	N. Horning	Magnus Homann
Werner Macho	Carson J.Q. Farmer	Tyler Mitchell	K. Koy	Lars Luthman
Claudia A. Engel	Brendan Morely	David Willis	Jürgen E. Fischer	Marco Hugentobler
Larissa Junek	Diethard Jansen	Paolo Corti	Gavin Macaulay	Gary E. Sherman
Tim Sutton	Alex Bruy	Raymond Nijssen	Richard Duivenvoorde	Andreas Neumann
Astrid Emde	Yves Jacolin	Alexandre Neto	Andy Schmid	Hien Tran-Quang

Copyright (c) 2004 - 2014 QGIS Development Team

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

### Lizenz des Dokuments

Es wird die Erlaubnis gewährt, dieses Dokument zu kopieren, zu verteilen und/oder zu modifizieren, unter den Bestimmungen der GNU Free Documentation License, Version 1.3 oder jeder späteren Version, veröffentlicht von der Free Software Foundation; ohne unveränderliche Abschnitte, ohne vordere Umschlagtexte und ohne hintere Umschlagtexte. Eine Kopie der Lizenz wird im Kapitel *GNU Free Documentation License* bereitgestellt.



---


## Gebrauch der Dokumentation

---

Dieser Abschnitt beschreibt die in diesem Handbuch benutzten einheitlichen Schreibstile.

### 2.1 GUI Schreibstile

Die GUI Schreibstile sollen das Erscheinungsbild der Grafischen Benutzeroberfläche nachahmen. Im Allgemeinen gibt ein Stil das einfache Erscheinungsbild wieder, so dass der Benutzer die GUI nach etwas das wie die Instruktionen im Handbuch aussieht absuchen kann.

- Menü Optionen: *Layer* → *Rasterlayer hinzufügen* oder *Einstellungen* → *Werkzeugkasten* → *Digitalisierung*
- Werkzeug:  Rasterlayer hinzufügen
- Knopf : **[Speicher als Standard]**
- Titel einer Dialogbox: *Layerereigenschaften*
- Reiter: *Allgemein*
- Kontrollkästchen:  *Darstellen*
- Radio Button:  *Postgis SRID*  *EPSG ID*
- Wähle eine Zahl:
- Wähle ein Wort:
- Suche nach einer Datei:
- Wähle eine Farbe:
- Schieberegler:
- Eingabetext:

Ein Schatten zeigt, dass dieses GUI Element mit der Maus anwählbar ist.

### 2.2 Text oder Tastatur Schreibstile

Dieses Handbuch enthält auch Stile die sich auf Texte, Tastaturbefehle und Code beziehen um verschiedene Einheiten, wie Klassen und Methoden anzuzeigen. Diese Stile entsprechen nicht dem tatsächlichen Aussehen von Texten oder Codezeilen innerhalb von QGIS.



- Querverweise: <http://qgis.org>
- Tastenkombinationen: Drücken Sie `Strg+B`, was heisst dass Sie die `Strg`-Taste drücken und halten sollen und dann die `B`-Taste drücken sollen.

- Name einer Datei: `lakes.shp`
- Name einer Klasse: **New Layer**
- Methode: `classFactory`
- Server: `myhost.de`
- User Text: `qgis --help`



Kodezeilen werden durch eine Schriftart mit festgelegter Breite angezeigt:

```
PROJCS["NAD_1927_Albers",  
  GEOGCS["GCS_North_American_1927",
```


## 2.3 Betriebssystemspezifische Anweisungen


GUI Sequenzen und kleine Textzeilen werden wie folgt formatiert: Klicken Sie   Datei **X** QGIS → QGIS beenden. Dies beschreibt dass Sie unter Linux, Unix und Windows Betriebssystemen erst das Datei-Menü und dann 'QGIS beenden' klicken sollen während Sie unter dem Macintosh OS X Betriebssystem erst das QGIS-Menü klicken und dann 'Verlassen' klicken sollen.

Größere Texte können als Liste formatiert werden:

-  Mache dies
-  Mache das
- **X** Mache etwas anderes

oder als Paragraph

 **X** Mache dies und dies und dies. Dann mache dies und dies und dies, und dies und dies und dies, und dies und dies und dies.

 Tun Sie das. Dann tun sie das und das und das, das und das und , das und das und das, das und das und das, das und das und das.

Abbildungen innerhalb der Dokumentation können unter verschiedenen Betriebssystemen erstellt worden sein. Das jeweilige Betriebssystem wird dabei am Ende der Abbildungsüberschrift mit einem Icon angezeigt.

.



---

## Vorwort

---

Willkommen in der wunderbaren Welt der Geographischen Informationssysteme (GIS)!

QGIS 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://qt.nokia.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.

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 *GNU General Public License* eingesehen werden.

---

### **Tipp: Aktuellste Dokumentation**

Die aktuellste Version dieses Dokuments können Sie immer im Dokumentationsbereich der QGIS Internetseite unter <http://www.qgis.org/en/docs/> finden.

---



---

## Funktionalitäten

---

QGIS bietet viele allgemeine GIS Funktionen die durch Kernfunktionen und Erweiterungen bereitgestellt werden. Eine kurze Zusammenfassung von sechs allgemeinen Kategorien von Funktionen und Erweiterungen wird unten dargestellt, gefolgt von ersten Einblicken in die integrierte Python-Konsole.

### 4.1 Daten visualisieren

Es ist möglich, Vektor- und Rasterdaten in unterschiedlichen Formaten und aus verschiedenen Projektionen anzuschauen und zu überlagern, ohne die Daten selbst in irgendeiner Art und Weise konvertieren zu müssen. Zu den unterstützten Datenformaten gehören z.B.:

- Tabellen und Views aus räumlichen Datenbanken wie PostGIS, SpatiaLite und MS SQL Spatial, Oracle Spatial, Vektorformate die durch die installierte OGR Bibliothek unterstützt werden, darunter ESRI Shape-dateien, MapInfo, SDTS, GML und viele mehr. Siehe Kapitel *Arbeiten mit Vektordaten*.
- Raster- und Bilddatenformate die von der installierten GDAL (Geospatial Data Abstraction Library) Bibliothek unterstützt werden wie z.B. GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG und viele mehr. Siehe Kapitel *Arbeiten mit Rasterdaten*.
- GRASS Raster- und Vektordaten aus GRASS Datenbanken (Location/Mapset). Siehe Kapitel *GRASS GIS Integration*.
- Online Geodaten welche als OGC Webservice, darunter WMS, WMTS, WCS, WFS und WFS-T, bereitgestellt werden. Siehe Kapitel *Arbeiten mit OGC Daten*.

### 4.2 Daten erkunden, abfragen und Karten layouts

Sie können Karten zusammenstellen und interaktiv räumliche Daten mit einer benutzerfreundlichen GUI erkunden. Die vielen in der GUI erhältlichen hilfreichen Werkzeuge beinhalten:

- QGIS Browser
- Spontanreprojektion
- DB Manager
- Drucklayouts erstellen mit dem Map Composer
- Kartenübersichtsfenster
- Räumliche Bookmarks
- Beschriftungswerkzeuge
- Identifizieren/Selektieren von Objekten
- Editieren/Visualisieren/Suchen von Attributdaten

- Datendefiniertes Beschriften von Objekten
- Datendefinierte Vektor- und Rastersymbolisierungswerkzeuge
- Atlas Kartenzusammenstellung mit Gradnetz-Layern
- Nordpfeil, Maßstab und Urheberrechtsnachweis
- Unterstützung für das Speichern und Wiederherstellen von Projekten

### 4.3 Daten erstellen, editieren, verwalten und exportieren

Sie können Vektor- und Rasterlayer erstellen, bearbeiten und in zahlreiche Formate exportieren. QGIS bietet die folgenden an:

- Digitalisierungswerkzeuge für OGR-unterstützte Formate und GRASS Vektorlayer
- Fähigkeit Shapedateien und GRASS Vektorlayer zu erstellen und zu bearbeiten
- Georeferenzierungsplugin um Bilder zu geocodieren
- GPS Werkzeuge um das GPX Format zu importieren und exportieren und andere GPS Format in GPS Format zu konvertieren oder um direkt in ein GPS Gerät runter/hochzuladen (Unter Linux ist usb: zur Liste der GPS Geräte hinzugefügt worden.)
- Unterstützung für das Darstellen und Bearbeiten von OpenStreetMap Daten
- Fähigkeit räumliche Datenbanktabellen aus Shapedateien mittels des DB Manager Plugins zu erstellen
- Verbesserte Handhabung von räumlichen Datenbanktabellen
- Werkzeuge um Vektorattributtabelle zu verwalten
- Möglichkeit Screenshots als georeferenzierte Bilder zu speichern

### 4.4 Daten analysieren

Sie können räumliche Datenanalysen auf räumliche Datenbanken und andere OGR-unterstützte Formate anwenden. QGIS bietet zur Zeit Vektoranalysen, Sampling, Geoverarbeitung sowie Geometrie und Datenbankmanagementwerkzeuge. Sie können auch die integrierten GRASS Werkzeuge, die die komplette GRASS Funktionalität von mehr als 400 Modulen beinhalten, benutzen. (Siehe Kapitel *GRASS GIS Integration* .). Oder Sie arbeiten mit dem Verarbeitungs Plugin, das eine mächtige räumliche Analyseumgebung zur Verfügung stellt mit dem man native oder Drittanbieter-Algorithmen aus QGIS heraus aufrufen kann, z.B. GDAL, SAGA, fTools und weitere. (Siehe Kapitel *Einführung*.)

### 4.5 Karten im Internet veröffentlichen

QGIS kann als WMS, WMTS, WMS-C oder WFS und WFS-T Client und als WMS, WCS oder WFS Server benutzt werden. (Siehe Kapitel *Arbeiten mit OGC Daten*.) Zusätzlich können Sie Ihre Daten im Internet veröffentlichen in dem Sie einen mit UMN MapServer oder GeoServer installierten Webserver verwenden.

### 4.6 Erweiterte QGIS Funktionalität durch Erweiterungen

QGIS kann an Ihre speziellen Bedürfnisse mit seiner erweiterbaren Pluginarchitektur und Bibliotheken die zum Erstellen von Plugins benutzt werden können angepasst werden. Sie können sogar neue Anwendungen mit C++ oder Python erstellen!

## 4.6.1 Kern Plugins

Kernplugins sind:

1. Koordinaten aufnehmen (Nehmen Sie Mauskoordinaten in verschiedenen KBS auf)
2. DB Manager (Austauschen, Bearbeiten und Darstellen von Layern und Tabellen; Ausführen von SQL Abfragen.
3. Diagram Overlay (Place diagrams on vector layers)
4. Dxf2Shp-Konverter (DXF Dateien zu Shapedateien konvertieren)
5. eVis (Events visualisieren)
6. fTools (Vektordaten analysieren und verwalten)
7. GDALTools (Integration von GDALTools in QGIS)
8. GDAL-Georeferenzierung (Rasterdateien Projektionsinformationen hinzufügen mit GDAL)
9. GPS Werkzeuge (GPS Daten laden und importieren)
10. GRASS (GRASS GIS integrieren)
11. Heatmap (Erzeuge ein Heatmap-Raster aus Punktdaten)
12. Interpolationserweiterung (Interpolation basierend auf Stützpunkten von Vektorlayern)
13. Offline-Bearbeitung (Ermöglicht Offline-Bearbeitung und Synchronisation mit Datenbanken)
14. Oracle Spatial GeoRaster
15. Verarbeitung (vorher SEXTANTE)
16. Rastergeländeanalyse-Erweiterung (rasterbasierte Geländeanalyse)
17. Straßengraph-Erweiterung (Analysiert ein kürzeste-Wege-Netzwerk)
18. Räumliche Abfrage Plugin
19. SPIT (Importiert Shapedateien nach PostgreSQL/PostGIS)
20. SQL Anywhere Erweiterung (Speichern von Vektorlayern in einer SQL Anywhere - Datenbank)
21. Topologie-Prüfung (Topologische Fehler in Vektorlayern finden)
22. Zonenstatistikerweiterung (Berechnet Anzahl, Summe und Mittel eines Rasters für jedes Polygon eines Vektorlayers)

## 4.6.2 Externe Python Plugins

QGIS bietet eine steigende Anzahl von externen Python Plugins, die von der Community bereitgestellt werden, an. Diese Plugins werden in dem offiziellen Plugins Repository vorgehalten und können auf einfache Art und Weise mit der Python Plugin Installation installiert werden. Siehe Kapitel *Der Erweiterungen Dialog*.

## 4.7 Python-Konsole

Um Skripte zu erstellen ist es möglich die integrierte Python-Konsole zu benutzen die mit dem Menü *Erweiterungen* → *Python-Konsole* geöffnet werden kann. Die Konsole öffnet sich als nicht-modales Utility-Fenster. Für das Zusammenspiel mit der QGIS Umgebung gibt es die `qgis.utils iface` Variable, die eine Instanz von `QgsInterface` ist. Diese Schnittstelle ermöglicht den Zugang zum Kartenfenster, Menüs, Werkzeugleisten und anderen Teilen der QGIS Anwendung.

Weitere Informationen zum Arbeiten mit der Python-Konsole und das Programmieren von QGIS Plugins und Anwendungen finden Sie unter [http://www.qgis.org/html/en/docs/pyqgis\\_developer\\_cookbook/index.html](http://www.qgis.org/html/en/docs/pyqgis_developer_cookbook/index.html).

## 4.8 Bekannte Probleme

### 4.8.1 Begrenzung der Anzahl von geöffneten Dateien

Wenn Sie ein großes QGIS Projekt öffnen und Sie sicher sind dass alle Layer gültig sind aber einige Layer als schlecht markiert sind stehen Sie wahrscheinlich vor diesem Problem. Linux (und auch andere Bs) hat eine prozessbezogene Begrenzung von geöffneten Dateien. Ressourcengrenzen beziehen sich auf einen Prozess und werden vererbt. Der `ulimit` Befehl, den die Shell zur Verfügung stellt, verändert die Begrenzung nur für den aktuellen Shell-Prozess; die neue Begrenzung wird von allen untergeordneten Prozessen geerbt.

Sie können sich die Informationen zu `ulimit` anzeigen lassen indem Sie folgenden Befehl eingeben

```
user@host:~$ ulimit -aS
```

Sie können die aktuell zulässige Anzahl von geöffneten Dateien pro Prozess mit dem folgenden Befehl in der Konsole sehen

```
user@host:~$ ulimit -Sn
```

Um die Begrenzungen für eine **vorhandene Sitzung** zu ändern, können Sie so etwas verwenden

```
user@host:~$ ulimit -Sn #number_of_allowed_open_files
user@host:~$ ulimit -Sn
user@host:~$ qgis
```

#### Um es für immer zu beheben

Unter den meisten Linux Systemen werden die Ressourcenbegrenzungen beim Login durch das `pam_limits` Modul gemäß den Einstellungen in `etc/security/limits.conf` oder `/etc/security/limits.d/*.conf` eingestellt. Sie müssten in der Lage sein die Dateien zu editieren wenn Sie Root-Rechte haben (auch über `sudo`), die Änderungen jedoch werden erst wirksam wenn Sie sich erneut anmelden.

Mehr Informationen:

<http://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/> <http://linuxaria.com/article/open-files-in-linux?lang=en>

.

---

## Was ist neu in QGIS 2.6

---

Diese Veröffentlichung enthält neue Features und erweitert die Programmierschnittstelle vorhergehender Versionen. Wir empfehlen dass Sie diese Version statt der vorherigen Veröffentlichungen benutzen.

Diese Version enthält hunderte Fehlerkorrekturen und viele neue Funktionen und Verbesserungen die in diesem Handbuch beschrieben werden. Sie können dazu auch den visuellen Changelog unter <http://changelog.linfinity.com/qgis/version/2.6.0/> überprüfen.

### 5.1 Anwendungen und Projektoptionen

- **Projektdateiname in Eigenschaften:** Sie können jetzt den vollständigen Pfad für das QGIS-Projekt im Projekteigenschaften Dialog sehen.

### 5.2 Datenprovider

- **DXF Exportwerkzeugverbesserungen:**
  - Baumansicht und Attributauswahl für die Layerzuweisung im Dialog
  - Unterstützung für Flächenfüllung/SCHRAFFUR
  - Texte als MTEXT statt TEXT ausgeben (einschließlich Text, Neigung und Dicke)
  - Unterstützung für RGB-Farbe, wenn es keine genaue Farbübereinstimmung gibt
  - AutoCAD 2000-DXF (R15) statt R12 nutzen

### 5.3 Kartenzusammenstellung

- **Ausmaße in der Karte anzeigen: Auf das Element** in den Elementeigenschaften gibt es jetzt zwei zusätzliche Knöpfe die es Ihnen ermöglichen (1) den Kartenausschnitt entsprechend dem Ausmaß Ihres Kartenelementes einzustellen und (2) in der Kartenansicht die aktuell für das Kartenelement festgelegten Ausmaße darzustellen.
- **Unterstützung für mehrere Gitter:** Es ist jetzt möglich mehr als ein Gitter in Ihr Kartenelement einzufügen. Jedes Gitter ist vollständig anpassbar und kann einem unterschiedlichen KBS zugeordnet werden. Das heißt zum Beispiel dass Sie jetzt ein Kartenlayout sowohl mit geografischem wie projiziertem Gitter erstellen können.
- **Selektiver Export:** Sie können bei jedem Element Ihres Druckzusammenstellungslayouts unter den Darstellungsoptionen das Objekt vom Kartenexport ausschließen.

## 5.4 QGIS Server

## 5.5 Darstellung

## 5.6 Benutzeroberfläche



---

## Der erste Einstieg

---

Dieses Kapitel gibt einen schnellen Überblick über das Installieren von QGIS, einige Beispieldaten von der QGIS Internetseite und das erste Starten sowie eine einfache Sitzung bei der Raster- und Vektorlayer visualisiert werden.

### 6.1 Installation

Die Installation von QGIS ist sehr einfach. Für MS Windows und Mac OS X sind Standardinstallationspakete erhältlich. Für viele GNU/Linux Betriebssysteme stehen Binärpakete (rpm und deb) oder Softwarerepositories die Ihrem Installationsmanager hinzugefügt werden können zur Verfügung. Holen Sie sich die neuesten Informationen über Binärpakete von der QGIS Internetseite unter <http://download.qgis.org>.

#### 6.1.1 Kompilieren des Quellcodes

Wenn Sie QGIS aus dem Quellcode installieren lesen Sie bitte in den Installationsanweisungen nach. Werden mit dem QGIS Quellcode in einer Datei mit dem Namen `INSTALL` betrieben. Sie können Sie auch online unter <http://htmlpreview.github.io/?https://raw.githubusercontent.com/qgis/QGIS/master/doc/INSTALL.html> finden.

#### 6.1.2 Installation auf externen Medien


Unter QGIS können Sie eine `--configpath` Einstellung die den Standardpfad für die Benutzerkonfiguration überschreibt (z.B. `~/qgis2` unter Linux) und **QSettings** zwingt dieses Verzeichnis auch zu benutzen. So können Sie z.B. eine QGIS Installation auf einem Flashlaufwerk mit allen Plugins und Einstellungen durchführen. Siehe Kapitel *Menü System* für zusätzliche Informationen.

### 6.2 Beispieldaten

Die Dokumentation zeigt eine Reihe von Beispielen, die auf den Geodaten des QGIS Beispieldatensatzes basieren.

Während der Installation unter Windows gibt es die Option, den QGIS Beispieldatensatz mit herunterzuladen. Wenn die Option ausgewählt wurde, werden die Daten nach Eigene Dateien in den Ordner GIS Database heruntergeladen. Mit dem Windows Explorer können Sie die Daten bei Bedarf nachträglich in ein anderes Verzeichnis verschieben. Wenn Sie die Option bei der Installation nicht ausgewählt haben, können Sie

- bereits auf Ihrem Rechner vorhandene GIS Daten verwenden
- den QGIS Beispieldatensatz nachträglich von [http://download.osgeo.org/qgis/data/qgis\\_sample\\_data.zip](http://download.osgeo.org/qgis/data/qgis_sample_data.zip) herunterladen
- QGIS deinstallieren, wieder neu installieren und dabei die entsprechende Option auswählen, wenn die oben angesprochenen Optionen nicht funktionieren.

 Für GNU/Linux und Mac OSX wird momentan noch kein fertiges Installationspaket für den Beispieldatensatz als rpm, deb or dmg bereitgestellt. Sie müssen die Datei `qgis_sample_data` als ZIP- oder TAR-Archiv von der URL [http://download.osgeo.org/qgis/data/qgis\\_sample\\_data.zip](http://download.osgeo.org/qgis/data/qgis_sample_data.zip) herunterladen und auf Ihrem Rechner entpacken.

Der QGIS Beispieldatensatz enthält Geodaten von Alaska und deckt sämtliche Übungen und Screenshots dieser Dokumentation ab, inklusive einer kleinen GRASS GIS Datenbank. Das Koordinatenbezugssystem ist Albers Equal Area mit der Maßeinheit 'feet' (EPSG-Code 2964).




```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]],
TOWGS84[-3,142,183,0,0,0,0],
AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]],
AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

Wenn Sie QGIS überwiegend als grafische Benutzeroberfläche für GRASS GIS verwenden möchten, finden Sie weitere GRASS GIS Beispiel-Locations (z.B. Spearfish oder South Dakota) auf der offiziellen GRASS Website unter der URL: <http://grass.osgeo.org/download/sample-data/>.


## 6.3 Ein erstes Übungsbeispiel






Nachdem Sie QGIS installiert und den Beispieldatensatz heruntergeladen und entpackt haben, beginnen wir mit einem einfachen und kurzen Beispiel. Ziel ist es, einen Raster- und einen Vektorlayer zu laden und wir verwenden dazu den Rasterlayer `qgis_sample_data/raster/landcover.img` und den Vektorlayer `qgis_sample_data/gml/lakes.gml`.

### 6.3.1 QGIS starten

-  Starten Sie QGIS indem Sie "QGIS" in die Kommandozeile eingeben oder wenn Sie ein vorkompiliertes Binärpaket verwenden indem Sie das Menü Anwendungen benutzen.
-  Starten Sie QGIS über das Start Menü, das QGIS Desktop Icon oder durch doppelklicken auf eine evtl. bereits vorhandene QGIS Projektdatei.
-  Doppelklicken Sie auf das QGIS Icon in Ihrem Programmordner.

### 6.3.2 Laden eines Raster- und Vektorlayers aus dem Beispieldatensatz




1. Drücken Sie auf den  Rasterlayer hinzufügen Knopf.
2. Wechseln Sie in den Ordner `qgis_sample_data/raster/`, wählen Sie die ERDAS IMG Datei `landcover.img` und klicken Sie auf **[Öffnen]**.

3. Wenn die Datei nicht aufgelistet ist, prüfen Sie in der *Dateien des Typs*  Combobox im unteren Bereich des Dialogs, ob der richtige Datentyp, in diesem Fall “Erdas Imagine Images (\*.img, \*.IMG)” eingestellt ist.
4. Nun drücken Sie auf den  Vektorlayer hinzufügen Knopf.
5. Als *Quellentyp* sollte im neuen *Vektorlayer hinzufügen* Dialog *Datei* ausgewählt sein. Klicken Sie jetzt **[Durchsuchen]** um den Vektorlayer auszusuchen.
6. Wechseln Sie in den Ordner `qgis_sample_data/gml/`, wählen Sie ‘Geography Markup Language [GML] [OGR] (.gml,GML)’ aus dem *Filter*  Pulldown Menü, dann wählen Sie die GML Datei `lakes.gml` und klicken auf **[Öffnen]**. Im Dialog *Vektorlayer hinzufügen* dann nochmal auf **[Öffnen]** klicken. Der *Koordinatenbezugssystem-Auswahl* Dialog öffnet sich mit ausgewähltem *NAD27 / Alaska Albers*; klicken Sie **[OK]**.
7. Zoomen Sie in einen Bereich in dem sich ein paar Seen befinden.
8. Doppelklicken Sie auf `lakes` in der Legende. Der Dialog *Layereigenschaften* öffnet sich.
9. Klicken Sie auf das Menü *Stil* und wählen Sie Blau als Füllfarbe.
10. Klicken Sie auf das Menü *Beschriftungen* und aktivieren Sie das  *Layer beschriften mit* Kontrollkästchen um das Beschriften einzuschalten. Wählen Sie das Feld “NAMES” als Feld das Beschriftungen enthält.
11. Um die Lesbarkeit der Beschriftungen zu verbessern können Sie einen weißen Puffer um sie bilden indem Sie links “Buffer” klicken, das Kontrollkästchen  *Textpuffer zeichnen* aktivieren und 3 als Puffergröße wählen.
12. Drücken Sie nun auf den Knopf **[Anwenden]**, prüfen Sie, ob das Ergebnis gut aussieht und bestätigen Sie dann mit einem Klick auf **[OK]**.

Sie sehen, wie einfach es ist, Raster- und Vektorlayer in QGIS zu visualisieren. Gehen Sie nun weiter zu den folgenden Kapiteln, um mehr über die vorhandenen Funktionalitäten, Einstellungsmöglichkeiten und ihre Benutzung zu erfahren.


## 6.4 QGIS Starten und Beenden

In Kapitel *Ein erstes Übungsbeispiel* wurde bereits kurz gezeigt, wie QGIS gestartet wird. Dies wird hier wiederholt und Sie werden sehen, dass QGIS darüber hinaus noch eine Reihe von Kommandozeilenoptionen zur Verfügung stellt.

-  Wenn QGIS bereits in einem ausführbaren Pfad installiert ist, können Sie QGIS in einem Kommandozeilenfenster mit dem Befehl: `qgis` starten, oder durch einen Doppelklick auf das QGIS Icon auf dem Desktop oder im Anwendungen Menü.
-  Starten Sie QGIS über das Start Menü, das QGIS Desktop Icon oder durch doppelklicken auf eine evtl. bereits vorhandene QGIS Projektdatei.
-  Doppelklicken Sie auf das QGIS Icon in Ihrem Programmordner. Wenn Sie QGIS aus der Shell starten wollen, verwenden Sie `/Pfad-zu-den-ausführbaren-Dateien/Contents/MacOS/Qgis`.

Um QGIS zu beenden klicken Sie im Menü   *Datei*  *QGIS* → *Beenden* oder benutzen Sie das Tastenkürzel `Strg+Q`.

## 6.5 Optionen der Kommandozeile

 QGIS Wenn Sie QGIS in der Kommandozeile starten, stehen eine Reihe von Optionen zur Verfügung. Eine Liste erhalten Sie, indem Sie `qgis ---help` eingeben. Die Ausgabe zeigt folgende Informationen:

```
qgis --help
QGIS - 2.6.0-Brighton 'Brighton' (exported)
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
OPTION:
  [--snapshot filename]  emit snapshot of loaded datasets to given file
  [--width width]       width of snapshot to emit
  [--height height]     height of snapshot to emit
  [--lang language]     use language for interface text
  [--project projectfile] load the given QGIS project
  [--extent xmin,ymin,xmax,ymax] set initial map extent
  [--nologo]            hide splash screen
  [--noplugins]         don't restore plugins on startup
  [--nocustomization]   don't apply GUI customization
  [--customizationfile] use the given ini file as GUI customization
  [--optionspath path]  use the given QSettings path
  [--configpath path]   use the given path for all user configuration
  [--code path]         run the given python file on load
  [--defaultui]         start by resetting user ui settings to default
  [--help]              this text

FILE:
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
```

---

### **Tipp: Ein Beispiel mit der Kommandozeile**

Sie können einen oder mehrere Kartenlayer in der Kommandozeile angeben, wenn Sie QGIS starten. Z.B. wenn Sie sich in dem Ordner `qgis_sample_data` befinden, können Sie durch folgendes Kommando QGIS mit einem Vektor- und einen Rasterlayer starten: `qgis ./raster/landcover.img ./gml/lakes.gml`

---

#### **Kommandozeilenoption** `--snapshot`

Diese Option ermöglicht es, einen PNG-Snapshot des aktuellen Kartenfensters zu erstellen. Dies ist z.B. sinnvoll, wenn Sie zahlreiche Projekte angelegt haben und Snapshots von den Daten machen wollen.

QGIS erstellt ein PNG-Bild mit 800x600 Pixeln. Dies können Sie mit den Parametern `---width` und `---height` anpassen und dann hinter der Option `--snapshot` einen Dateinamen angeben.

#### **Kommandozeilenoption** `--lang`

Auf Basis der Systemsprache Ihres Rechners wird auch die Sprache der QGIS-Oberfläche eingestellt. Wenn Sie diese ändern möchten, können Sie einen Sprachcode festlegen. Z.B. startet `--lang=it` QGIS in italienischer Lokalisierung. Eine Liste der unterstützten Sprachen finden Sie mit dem entsprechenden Länderkürzel unter [http://hub.qgis.org/wiki/quantum-gis/GUI\\_Translation\\_Progress](http://hub.qgis.org/wiki/quantum-gis/GUI_Translation_Progress).

#### **Kommandozeilenoption** `--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.

#### **Kommandozeilenoption** `---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
```

#### **Kommandozeilenoption** `--nologo`

Diese Option verhindert das Anzeigen des Begrüßungsbildschirms beim Starten von QGIS.

**Kommandozeilenoption** `--noplugins`

Wenn Sie Probleme mit dem Starten von Erweiterungen haben können Sie das Laden beim Hochfahren von QGIS verhindern. Die Erweiterungen stehen danach immer noch über den QGIS-Erweiterungsmanager zu Verfügung.

**Kommandozeilenoption** `--customizationfile`

Wenn Sie diese Kommandozeilenoption verwenden können Sie eine GUI Anpassungsdatei definieren die dann beim Start verwendet wird.

**Kommandozeilenoption** `--nocustomization`

Wenn Sie diese Kommandozeilenoption verwenden wird die bestehende GUI Anpassung beim Start nicht angewendet.

**Kommandozeilenoption** `--optionspath`

Sie können Mehrfachkonfigurationen durchführen und entscheiden welche Sie verwenden wollen wenn Sie QGIS unter Verwendung dieser Option starten. Unter *Optionen* können Sie überprüfen wo das Betriebssystem die Einstellungen speichert. Derzeit gibt es noch keine Möglichkeit die Datei festzulegen, in die die Einstellungen gespeichert werden. Aus diesem Grund können Sie eine Kopie der Originaldatei machen und sie umbenennen. Die Option legt den Pfad zum Verzeichnis mit Einstellungen fest. Wenn Sie zum Beispiel die `/path/to/config/QGIS/QGIS2.ini` Einstellungen verwenden wollen verwenden Sie die Option:

```
--optionspath /path/to/config/
```

**Kommandozeilenoption** `--configpath`

Diese Option ähnelt der vorangegangenen, überschreibt jedoch den Standardpfad (`~/.qgis2`) für die Benutzerkonfiguration und zwingt **QSettings** dieses Verzeichnis zu verwenden. So kann der Benutzer z.B. eine QGIS-Installation mit allen Erweiterungen und Einstellungen auf einem USB-Stick transportieren.

**Kommandozeilenoption** `-code`



Diese Option kann verwendet werden um eine vorgegebene Python-Datei direkt nachdem QGIS gestartet hat auszuführen.


Zum Beispiel wenn Ihnen eine Python-Datei genannt `load_alaska.py` mit dem folgenden Inhalt vorliegt:


```
from qgis.utils import iface
raster_file = "/home/gisadmin/Documents/qgis_sample_data/raster/landcover.img"
layer_name = "Alaska"
iface.addRasterLayer(raster_file, layer_name)
```

Angenommen Sie befinden sich in dem Verzeichnis wo sich die Datei `load_alaska.py` befindet können Sie QGIS starten, die Rasterdatei `file:landcover.img` laden und dem Layer den Namen 'Alaska' geben in dem Sie den folgenden Befehl benutzen: `qgis --code load_alaska.py`

## 6.6 QGIS Projekte

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 *Optionen*). QGIS kann den aktuellen Status eines Projektes in einer Projektdatei speichern. Wählen Sie dazu in der Menüleiste Datei *Projekt* →  *Projekt speichern* oder *Projekt* →  *Projekt speichern als*.

Gespeicherte QGIS Projekte können anhand von *Projekt* →  *Öffnen...*, *Projekt* → *Neu aus Vorlage* oder *Projekt* → *Zuletzt verwendetes* → in eine QGIS Sitzung geladen werden.

Wenn Sie Ihre Sitzung löschen wollen und neu starten wählen Sie *Projekt* →  *Neu*. Jede dieser Menüoptionen wird Sie auffordern das bestehende Projekt zu speichern seit es geöffnet wurde oder zuletzt gespeichert wurde.

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 leider nicht. Um darauf hingewiesen zu werden, können Sie im Menü *Allgemein* im Menü *Einstellungen* → *Optionen* das Kontrollkästchen:

- *Bei Bedarf nachfragen ob geänderte Projekte und Datenquellen gespeichert werden sollen*
- *Warnung ausgeben, wenn QGIS-Projekt einer früheren Version geöffnet wird auswählen*

Immer wenn Sie in QGIS 2.2 ein Projekt speichern wird nun ein Backup von der Projektdatei gemacht.

## 6.7 Ausgabe

Es gibt viele Wege eine Ausgabe von Ihrer QGIS Sitzung zu erzeugen. Einen Weg haben Sie bereits in Abschnitt *QGIS Projekte* gezeigt. Hier finden Sie ein Beispiel Ausgabedateien auf andere Art und Weise zu erzeugen:

- Menüauswahl *Projekt* →  *Bild speichern als* öffnet einen Dateidialog indem Sie den Namen, Pfad und Bildtyp (PNG oder JPG Format) auswählen. Eine World-Datei mit der Erweiterung PNGW oder JPGW, die im gleichen Ordner abgelegt wird, georeferenziert das Bild.
- Menüoption *Projekt* → *DXF Export ...* öffnet einen Dialog in dem Sie den 'Darstellungsmodus', den 'Darstellungsmaßstab' und Vektorlayer, die Sie in DXF exportieren wollen, definieren können.
- Die Menüoption *Projekt* →  *Neue Druckzusammenstellung* öffnet einen Dialog in dem Sie das aktuelle Kartenfenster gestalten und drucken können (siehe Kapitel *Druckzusammenstellung*).

---

 QGIS GUI
 

---

Beim Öffnen von QGIS öffnet sich die grafische Benutzeroberfläche wie in der Abbildung zu sehen (die Nummern 1 bis 5 in gelben Kreisen werden unten erläutert).

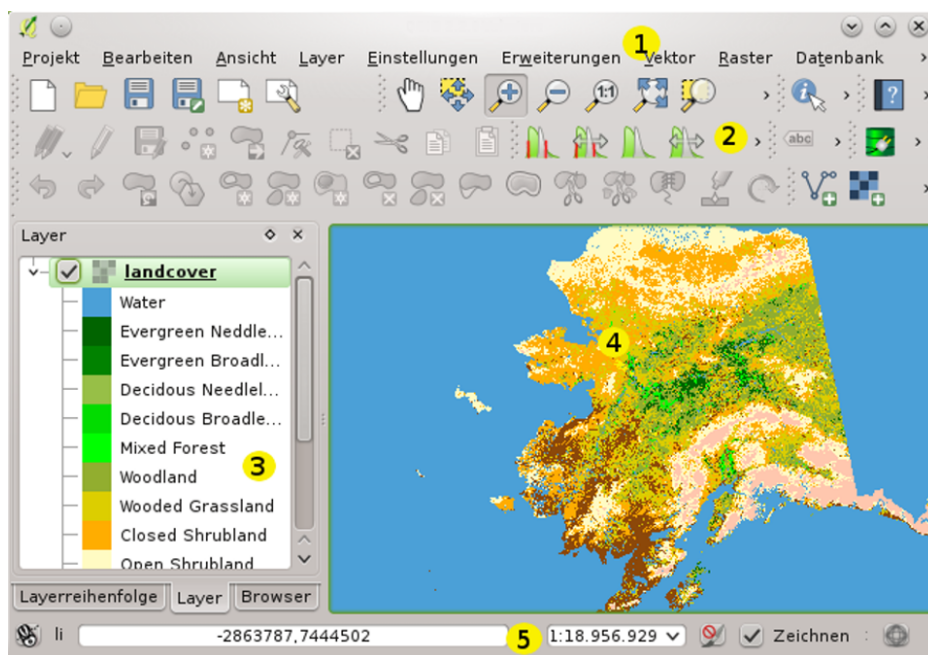



Abbildung 7.1: QGIS GUI mit Alaskabeispieldatensatz 

---

**Bemerkung:** Das Aussehen einzelner Bereiche (Titelleiste, etc.) kann in Abhängigkeit vom Betriebssystem und dem Fenstermanager abweichen.

---

Die QGIS GUI ist in fünf Bereiche unterteilt:

1. Menüleiste
2. Werkzeugleiste
3. Legende
4. Kartenfenster
5. Statusleiste









Diese fünf Komponenten der QGIS-Schnittstelle werden in den folgenden Abschnitten detaillierter beschrieben. Zwei weitere Abschnitte stellen die Tastenkürzel und das Hilfemenu vor.

## 7.1 Menüleiste

Die Menüleiste bietet Zugriff auf verschiedene Anwendungen mit Hilfe eines hierarchischen Menüs. Das Hauptmenü mit einer kurzen Beschreibung der enthaltenen Funktionen wird hier zusammen mit dem jeweiligen Icon, das auch in der Werkzeugleiste zu finden ist und dem Tastenkürzel aufgelistet. Die Tastenkürzel stellen dabei die Standardeinstellung dar und können im Menü *Einstellungen* → *Tastenkürzel festlegen* definiert werden.

Obwohl fast jede Anwendung der Menüleiste ein entsprechendes Icon in der Werkzeugleiste besitzt, ist die Anordnung unterschiedlich. Die Werkzeugleiste mit der entsprechenden Anwendung ist hinter jedem Menüeintrag als Kontrollkästchen angegeben. Weitere Informationen über die Werkzeuge in der Werkzeugleiste finden Sie in Abschnitt *Werkzeugleiste*.
























### 7.1.1 Projekt


Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 <i>Neu</i>	Ctrl+N	siehe <i>QGIS Projekte</i>	<i>Projekt</i>
 <i>Öffnen ...</i>	Strg+O	siehe <i>QGIS Projekte</i>	<i>Projekt</i>
<i>Neu aus Vorlage →</i>		siehe <i>QGIS Projekte</i>	<i>Projekt</i>
<i>Zuletzt verwendetes →</i>		siehe <i>QGIS Projekte</i>	
 <i>Speichern</i>	Strg+S	siehe <i>QGIS Projekte</i>	<i>Projekt</i>
 <i>Speichern als ...</i>	Strg+Shift+S	siehe <i>QGIS Projekte</i>	<i>Projekt</i>
 <i>Bild speichern als ...</i>		siehe <i>Ausgabe</i>	
<i>DXF-Export ...</i>		siehe <i>Ausgabe</i>	
 <i>Neue Druckzusammenstellung</i>	Strg+P	siehe <i>Druckzusammenstellung</i>	<i>Projekt</i>
 <i>Druckzusammenstellungen verwalten ...</i>		siehe <i>Druckzusammenstellung</i>	<i>Projekt</i>
<i>Druckzusammenstellungen →</i>		siehe <i>Druckzusammenstellung</i>	
 <i>QGIS beenden</i>	Strg+Q		








## 7.1.2 Bearbeiten
















Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 <i>Rückgängig</i>	Strg+Z	siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Wiederholen</i>	Strg+Shift+Z	siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Ausgewählte Objekte ausschneiden</i>	Strg+X	siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
 <i>Objekte kopieren</i>	Strg+C	siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
 <i>Objekte einfügen</i>	Strg+V	siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
<i>Objekte einfügen als →</i>		siehe <i>Mit Attributtabelle arbeiten</i>	
 <i>Objekt hinzufügen</i>	Strg+.	siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
 <i>Objekte verschieben</i>		siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
 <i>Ausgewähltes löschen</i>		siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
 <i>Objekt(e) drehen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Objekt vereinfachen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Ring hinzufügen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Teil hinzufügen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Ring füllen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Ring löschen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Teil löschen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Objekte überarbeiten</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Linie versetzen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Objekte trennen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Teile zerlegen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Gewählte Objekte verschmelzen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
 <i>Attribute gewählter Objekte vereinen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte Digitalisierung</i>
<b>24</b>  <i>Knotenwerkzeug</i>		siehe <i>Einen vorhandenen Layer editieren</i>	<b>Kapitel 7. QGIS GUI</b> <i>Digitalisierung</i>
 <i>Punktsymbole drehen</i>		siehe <i>Erweiterte Digitalisierung</i>	<i>Erweiterte</i>

Nach dem Sie auf das Icon  Bearbeitungsstatus umschalten gedrückt haben finden Sie das “Objekt hinzufügen” Icon im *Bearbeiten* Menü das abhängig vom Layertyp (Punkt, Linie oder Polygon) dargestellt wird.

### 7.1.3 Bearbeiten (extra)

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 Objekt hinzufügen		siehe <i>Einen vorhandenen Layer editieren</i>	Digitalisierung
 Objekt hinzufügen		siehe <i>Einen vorhandenen Layer editieren</i>	Digitalisierung
 Objekt hinzufügen		siehe <i>Einen vorhandenen Layer editieren</i>	Digitalisierung





### 7.1.4 Ansicht

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 Karte verschieben			Kartennavigation
 Karte zur Auswahl verschieben			Kartennavigation
 Hineinzoomen	Strg++		Kartennavigation
 Herauszoomen	Strg+-		Kartennavigation
Auswahl →		siehe <i>Objekte auswählen und Auswahl aufheben</i>	Attribute
 Objekte abfragen	Strg+Shift+I		Attribute
Messen →		siehe <i>sec_measure</i>	Attribute
 Volle Ausdehnung	Strg+Shift+F		Kartennavigation
 Auf den Layer zoomen			Kartennavigation
 Zur Auswahl zoomen	Strg+J		Kartennavigation
 Zoom zurück			Kartennavigation
 Zoom vor			Kartennavigation
 Auf tatsächliche Größe zoomen			Kartennavigation
Dekorationen →		siehe <i>Dekorationen</i>	
 Kartenhinweise			Attribute
 Neues Lesezeichen ...	Strg+B	siehe <i>Räumliche Lesezeichen</i>	Attribute
 Lesezeichen anzeigen	Strg+Shift+B	siehe <i>Räumliche Lesezeichen</i>	Attribute
 Aktualisieren	Strg+R		Kartennavigation


## 7.1.5 Layer

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
Neu →		siehe <i>Neue Vektorlayer erstellen</i>	<i>Layer verwalten</i>
Eingebettete Layer und Gruppen ...		siehe <i>Layer/Gruppen einbinden</i>	
Vektorlayer hinzufügen	Strg+Shift+V	siehe <i>Arbeiten mit Vektordaten</i>	<i>Layer verwalten</i>
Rasterlayer hinzufügen	Strg+Shift+R	siehe <i>Rasterlayer in QGIS laden</i>	<i>Layer verwalten</i>
PostGIS-Layer hinzufügen	Strg+Shift+D	siehe <i>PostGIS Layer laden</i>	<i>Layer verwalten</i>
SpatiaLite-Layer hinzufügen ...	Strg+Shift+L	siehe <i>SpatiaLite Layer laden</i>	<i>Layer verwalten</i>
Räumlichen MSSQL-Layer hinzufügen	Strg+Shift+M	siehe <i>Räumliche MSSQL-Layer</i>	<i>Layer verwalten</i>
Oracle-GeoRaster-Layer hinzufügen		siehe <i>Oracle-Spatial-GeoRaster Plugin</i>	<i>Layer verwalten</i>
SQL-Anywhere-Layer hinzufügen ...		siehe <i>SQL-Anywhere Plugin</i>	<i>Layer verwalten</i>
WMS/WMTS-Layer hinzufügen	Strg+Shift+W	siehe <i>WMS/WMTS Client</i>	<i>Layer verwalten</i>
WCS-Layer hinzufügen		siehe <i>WCS Client</i>	<i>Layer verwalten</i>
WFS-Layer hinzufügen ...		siehe <i>WFS und WFS-T Klient</i>	<i>Layer verwalten</i>
Textdatei als Layer importieren ...		siehe <i>Delimited Text Dateien</i>	<i>Layer verwalten</i>
Stil kopieren		siehe <i>Menü Stil</i>	
Stil einfügen		siehe <i>Menü Stil</i>	
Attributtabelle öffnen		siehe <i>Mit Attributtabelle arbeiten</i>	<i>Attribute</i>
Bearbeitungsstatus umschalten		siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
Layeränderungen speichern		siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
Aktuelle Änderungen → Speichern als ...		siehe <i>Einen vorhandenen Layer editieren</i>	<i>Digitalisierung</i>
Auswahl als Vektordatei speichern ...		Siehe <i>Mit Attributtabelle arbeiten</i>	
Layer löschen	Strg+D		
Layer kopieren			
KBS von Layer(n) setzen	Strg+Shift+C		
Layer-KBS dem Projekt zuweisen			
Eigenschaften ...			
Abfrage ...			
Beschriftung			
Zur Übersicht hinzufügen	Strg+Shift+O		<i>Layer verwalten</i>
Alle zur Übersicht hinzufügen			
Alle aus Übersicht entfernen			
Alle Layer anzeigen	Strg+Shift+U		<i>Layer verwalten</i>
Alle Layer ausblenden	Strg+Shift+H		<i>Layer verwalten</i>

### 7.1.6 Einstellungen






Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
<i>Bedienfelder</i> → <i>Werkzeugkästen</i> → <i>Auf Vollbildmodus umschalten</i>  <i>Projekteinstellungen ...</i>  <i>Benutzerkoordinaten- bezugssystem Stilmanager...</i>  <i>Tastenkürzel festlegen ...</i> <i>mActionOptions   Anpassungen</i> ...  <i>Optionen ...</i> <i>Fangoptionen ...</i>	F 11  Strg+Shift+P	siehe <i>Bedienfelder und Werkzeugkästen</i> siehe <i>Bedienfelder und Werkzeugkästen</i>  siehe <i>QGIS Projekte</i>  siehe <i>Eigenes Koordinatenbezugssystem definieren</i> siehe <i>Darstellung</i>  siehe <i>Anpassung</i>  siehe <i>Optionen</i>	

### 7.1.7 Erweiterungen

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 <i>Erweiterungen verwalten und installieren ...</i> <i>Python-Konsole</i>		siehe <i>Der Erweiterungen Dialog</i>	

Wenn Sie QGIS das erste Mal starten werden nicht alle Erweiterungen geladen.

### 7.1.8 Vektor

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
<i>OpenStreetMap</i> →  <i>Analyse-Werkzeuge</i> →  <i>Forschungswerkzeuge</i> →  <i>Geoverarbeitungswerkzeuge</i> →  <i>Geometrie-Werkzeuge</i> →  <i>Datenmanagement-Werkzeuge</i> →		siehe <i>OpenStreetMap Vektordateien laden</i>  siehe <i>fTools Plugin</i> siehe <i>fTools Plugin</i> siehe <i>fTools Plugin</i> siehe <i>fTools Plugin</i> siehe <i>fTools Plugin</i>	







Wenn Sie QGIS das erste Mal starten werden nicht alle Erweiterungen geladen.

### 7.1.9 Raster

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
<i>Rasterrechner ...</i>		siehe <i>Rasterrechner</i>	







Wenn Sie QGIS das erste Mal starten werden nicht alle Erweiterungen geladen.


## 7.1.10 Verarbeitung





Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 <i>Werkzeugkiste</i>		siehe <i>Die Werkzeugkiste</i>	
 <i>Grafische Modellierung</i>		siehe <i>Die Grafische Modellierung</i>	
 <i>Protokoll</i>		siehe <i>Das Protokoll</i>	
 <i>Optionen und Konfiguration</i>		siehe <i>Die Verarbeiten Umgebung konfigurieren</i>	
 <i>Ergebnisanzeige</i>		siehe <i>Konfiguration externer Anwendungen</i>	
 <i>Commander</i>	Strg+Alt+M	siehe <i>The QGIS Commander</i>	

Wenn Sie QGIS das erste Mal starten werden nicht alle Erweiterungen geladen.

## 7.1.11 Hilfe

Menüleiste	Tastenkürzel	Referenz	Werkzeugleiste
 <i>Hilfe-Übersicht</i>	F1		<i>Direkthilfe</i>
 <i>Direkthilfe</i> <i>API-Dokumentation</i> <i>Brauchen Sie professionelle Unterstützung?</i>	Shift+F1		<i>Direkthilfe</i>
 <i>QGIS-Homepage</i>	Strg+H		
 <i>QGIS-Version überprüfen</i>			
 <i>Über</i>			
 <i>QGIS-Sponsoren</i>			

Bitte beachten Sie dass unter Linux  die oben aufgeführten Menüleisteneinträge der Standard im KDE Window Manager sind. Unter GNOME hat das *Einstellungen* Menü unterschiedliche Inhalte und Ihre Einträge lassen sich hier finden:

 <i>Projekteinstellungen</i>	<i>Projekt</i>
 <i>Optionen</i>	<i>Bearbeiten</i>
 <i>Tastenkürzel festlegen</i>	<i>Bearbeiten</i>
<i>Stilmanager</i>	<i>Bearbeiten</i>
 <i>Benutzerkoordinatenbezugssystem</i>	<i>Bearbeiten</i>
<i>Bedienfelder →</i>	<i>Ansicht</i>
<i>Werkzeugkästen →</i>	<i>Ansicht</i>
<i>Volle Ausdehnung</i>	<i>Ansicht</i>
<i>Kachelmaßstab</i>	<i>Ansicht</i>
<i>GPS-Information</i>	<i>Ansicht</i>

## 7.2 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 einen freien Bereich der Werkzeugleiste fahren und auf den rechten Mausknopf drücken (lesen sie auch Abschnitt *Bedienfelder und Werkzeugkästen*).

### Tipp: Werkzeugleiste wiederherstellen

Wenn sie aus Versehen alle Untermenüs der Werkzeugleiste ausgeblendet haben, können Sie diese über das Menü *Einstellungen* → *Werkzeugkästen* → wieder zurückholen. Wenn eine Werkzeugleiste unter Windows verschwindet - was unter QGIS manchmal ein Problem zu sein scheint- müssen Sie `\HKEY_CURRENT_USER\Software\QuantumGIS\qgis\UI\state` aus der Registry löschen. Wenn Sie dann QGIS neu starten wird der Key wieder mit Standardeinstellungen geschrieben und die Werkzeugleisten sind wieder sichtbar

## 7.3 Legende


Der Legendenbereich führt alle Layer im Projekt auf. Das Kontrollkästchen in jedem Legendeneintrag kann dazu verwendet werden um den Layer anzuzeigen oder auszublenden. Die Legendenwerkzeugleiste im Legendenbereich ermöglicht es Ihnen **Eine Gruppe hinzufügen**, **Layersichtbarkeitseinstellungen** aller Layer oder Sichtbarkeitsvoreinstellungen vorzunehmen, die **Legende nach Karteninhalt filtern**, **Alle ausklappen** oder **Alle einklappen** und **Layer/Gruppe löschen**. Der Knopf  ermöglicht es Ihnen, **Voreinstellungen** von Views in der Legende hinzuzufügen. Das heißt dass Sie sich entscheiden können Layer mit bestimmten Kategorisierungen darzustellen und diese Views zu der Liste der **Voreinstellungen** hinzuzufügen. Um den View einer Voreinstellung hinzuzufügen klicken Sie einfach auf , wählen Sie *Voreinstellung hinzufügen ...* aus dem Drop-down-Menü und vergeben Sie einen Namen für die Voreinstellung. Danach sehen Sie eine Liste mit allen Voreinstellungen, die Sie mit durch Drücken des  Knopfes wieder aufrufen können.

Alle hinzugefügten Voreinstellungen liegen auch in der Druckzusammenstellung vor um es Ihnen zu ermöglichen eine Karte auf Basis Ihrer bestimmten Views zu erstellen (siehe *Hauptigenschaften*).

Die Z-Anordnung der Kartenlayer kann mit der 'drag and drop' Funktion der Maus festgelegt werden. Z-Anordnung bedeutet, dass ein weiter oben in der Legende angeordneter Layer über einem weiter unten angeordneten Layer im Kartenfenster angezeigt wird.


**Bemerkung:** Diese Einstellungen können mit dem 'Layerreihenfolge' Bedienfeld überschrieben werden.

Einzelne Layer können in der Legende zu Gruppen zusammengefasst werden. Sie können dies auf zwei Arten tun:

1. Drücken Sie das  Icon um eine neue Gruppe hinzuzufügen. Geben Sie einen Namen für die Gruppe ein und drücken Sie `Enter`. Klicken Sie jetzt auf einen bestehenden Layer und ziehen Sie ihn auf die Gruppe.
2. Wählen Sie einige Layer aus, machen Sie einen Rechtsklick in die Legende und wählen Sie *Gewählte gruppieren*. Die ausgewählten Layer werden automatisch in eine neue Gruppe gesetzt.

Um einen Layer aus einer Gruppe zu bringen, können Sie ihn herausschieben oder einen Rechtsklick darauf machen und *In oberste Ebene bringen* wählen. Gruppen können auch in andere Gruppen verschachtelt werden.

Das Kontrollkästchen für eine Gruppe zeigt oder verbirgt alle Layer einer Gruppe mit einem Klick.

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 *Digitalisieren und Editieren eines GRASS Vektorlayers* für weitere Informationen zum Editieren von GRASS Vektorlayern.

### Rechte-Maustaste-Menü für Rasterlayer

- *Auf die Layerausdehnung zoomen*
- *In der Übersicht anzeigen*
- *Auf besten Maßstab zoomen (100%)*
- *Auf aktuelle Ausdehnung strecken*
- *Entfernen*

- *Kopieren*
- *Maßstabsabhängige Sichtbarkeit des/der Layer setzen*
- *KBS für Layer setzen*
- *Layer-KBS dem Projekt zuweisen*
- *Speichern als ...*
- *Als Layerdefinitionsdatei speichern ...*
- *Eigenschaften ...*
- *Umbenennen*
- *Stil kopieren*

Zusätzlich abhängig von der Layerposition und -auswahl

- *In oberste Ebene bringen*
- *Gewählte gruppieren*

#### **Rechte-Maustaste-Menü für Vektorlayer**

- *Auf die Layerausdehnung zoomen*
- *In der Übersicht anzeigen*
- *Entfernen*
- *Kopieren*
- *Maßstabsabhängige Sichtbarkeit des/der Layer setzen*
- *KBS für Layer setzen*
- *Layer-KBS dem Projekt zuweisen*
- *Attributtabelle öffnen*
- *Bearbeitungsstatus umschalten (bei GRASS Layern nicht möglich)*
- *Speichern als ...*
- *Als Layerdefinitionsdatei speichern ...*
- *Filter ...*
- *Objektanzahl anzeigen*
- *Eigenschaften ...*
- *Umbenennen*
- *Stil kopieren*

Zusätzlich abhängig von der Layerposition und -auswahl

- *In oberste Ebene bringen*
- *Gewählte gruppieren*

#### **Rechte Maustaste-Menü für Layergruppen**

- *Zur Gruppe zoomen*
- *Entfernen*
- *KBS für Gruppe setzen*
- *Umbenennen*
- *Gruppe hinzufügen*



Es ist möglich mehr als einen Layer oder Gruppe zur gleichen Zeit auszuwählen indem man die `Strg` Taste gedrückt hält und die Layer mit der linken Maustaste auswählt. Sie können dann alle ausgewählten Layer gleichzeitig zu einer neuen Gruppe verschieben.

Sie können auch mehr als einen Layer oder Gruppe auf einmal löschen indem Sie mehrere Layer mit der `Strg` Taste auswählen und danach `Strg+D` drücken. Auf diese Art und Weise werden alle ausgewählten Layer und Gruppen aus der Layerliste entfernt.

### 7.3.1 Mit der legendenunabhängigen Layerreihenfolge arbeiten

Es gibt ein Bedienfeld das es Ihnen ermöglicht eine unabhängige Zeichenreihenfolge für die Legende zu bestimmen. Sie können es im Menü *Einstellungen* → *Bedienfelder* → *Layerreihenfolge* aktivieren. Diese Feature ermöglicht es Ihnen z.B. Ihre Layer der Wichtigkeit nach zu sortieren und diese gleichzeitig in der korrekten Reihenfolge darzustellen (siehe [figure\\_layer\\_order](#)). Das Aktivieren des  *Bestimmt Zeichenreihenfolge* Kontrollkästchens unterhalb der Layerliste bewirkt eine Rückkehr zum Standardverhalten.

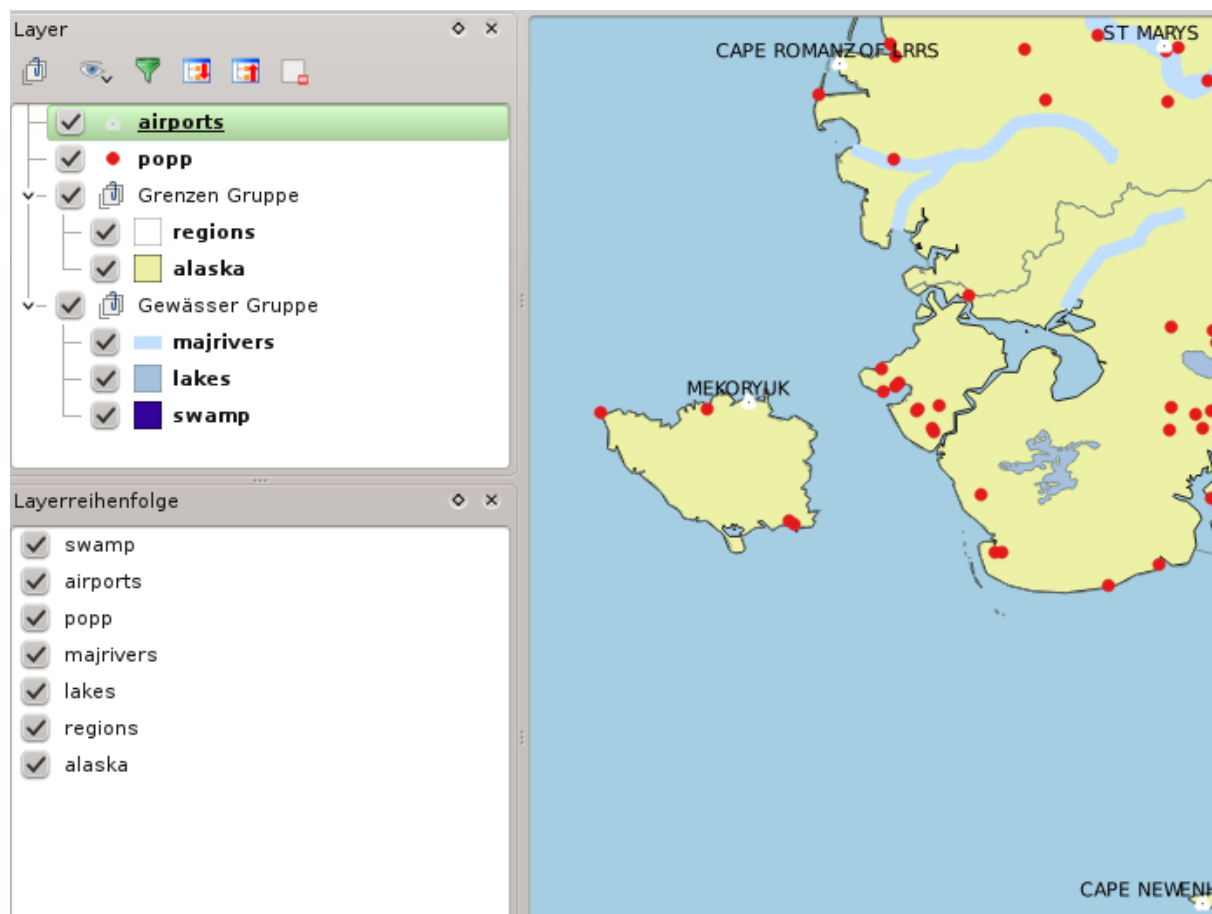



Abbildung 7.2: Eine legendenunabhängige Layerreihenfolge festlegen 

## 7.4 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 Legende sind miteinander verknüpft. Layer im Kartenfenster spiegeln Veränderungen in der Legende wider.

**Tipp: Mit dem Mausrad in der Karte zoomen**

Sie können das Mausrad benutzen, um im Kartenfenster in Layer hinein- bzw. hinauszuzoomen. Platzieren Sie dazu den Mauszeiger im Kartenfenster und drehen Sie das Mausrad nach vorne (Hinauszoomen) oder nach hinten zum Hineinzoomen. Der Mauszeiger bildet dabei das Zentrum. Sie können das Verhalten des Mauseisrades in der Menüleiste *Einstellungen* → *Optionen* unter dem Menü *Kartenwerkzeuge* einstellen.

---

**Tipp: Den Kartenausschnitt mit den Pfeiltasten und der Leertaste verschieben**

Sie können die Pfeiltasten verwenden, um den Layer zu verschieben. Platzieren Sie dazu den Mauszeiger im Kartenfenster und drücken Sie auf die rechte Pfeiltaste um nach Osten zu verschieben, die linke Pfeiltasten für Westen, die Hoch-Pfeiltasten für Norden und die Unten-Pfeiltasten für Süden. Sie können einen Layer auch mit der Space-Taste verschieben. Bewegen Sie dazu einfach die Maus während Sie die Space-Taste gedrückt halten.


---

## 7.5 Statusleiste

Die Statusleiste zeigt die aktuelle Position als Koordinate (z.B. in Meter oder Dezimalgrad) während der Mauszeiger über dem Kartenfenster verschoben wird. Links von der Koordinatenanzeige ist ein kleiner Knopf mit dem Sie auswählen können, ob die Koordinaten des Bezugssystems oder der Ausdehnung des Kartenfensters angezeigt werden sollen, während der Mauszeiger bewegt wird.

Neben der Koordinatenanzeige finden Sie die Maßstabsanzeige. Es zeigt den Maßstab des Kartenfensters. Wenn Sie herein- oder hinauszoomen zeigt QGIS ihnen den aktuellen Maßstab. Seit QGIS 1.8 gibt es eine Maßstabauswahl mit der Sie zwischen vordefinierten Maßstäben von 1:500 bis 1:1000000 wählen können.


Ein Fortschrittsbalken in der Statusleiste zeigt den Fortschritt des Zeichnens der einzelnen Layer im Kartenfenster. In einigen Fällen, etwa wenn statistische Informationen über Rasterlayer errechnet werden, wird der Fortschritt dieser Operation angezeigt.

Wenn ein neues externes Plugin oder ein Plugin Update im Internet verfügbar ist, wird diese Information auch in der Statusleiste angezeigt. Mit dem Kontrollkästchen auf der rechten Seite der Statusleiste kann eingestellt werden, ob Layer angezeigt werden sollen oder nicht (vgl. Abschnitt *Layeranzeige kontrollieren*). Das Icon  stoppt die Darstellung umgehend.

Rechts von den Renderingfunktionen finden Sie den EPSG Kode des aktuellen Projektes und ein Projektor Icon. Wenn man darauf klickt öffnet es die Projektionseigenschaften des aktuellen Projektes.

---

**Tipp: Die richtige Maßstabseinheit im Kartenfenster einstellen**

Wenn Sie QGIS starten, ist als Maßeinheit Grad voreingestellt und sagt QGIS, dass alle Koordinaten des Layers in Grad vorliegen. Um die korrekte Maßeinheit anzeigen zu lassen, können Sie es im Menü *Allgemein* im Menü *Einstellungen* → *Projekteinstellungen* einstellen oder sie wählen das passende Bezugskoordinatensystem (KBS) aus, indem Sie auf das Icon  KBS Status in der unteren rechten Ecke der Statusleiste klicken. Dann wird die Maßeinheit entsprechend der KBS eingestellt, z.B.: '+units=m'.

---

---

## Allgemeine Werkzeuge

---

### 8.1 Tastenkürzel

QGIS bietet Standard Tastenkürzel für zahlreiche Funktionen. Sie können im Kapitel *Menüleiste* nachgelesen werden. Zusätzlich ist es aber auch möglich, eigene Tastenkürzel zu definieren und bestehende zu ändern. Öffnen Sie dazu das Menü *Einstellungen* → *Tastenkürzel festlegen* ....



Abbildung 8.1: Tastenkürzel festlegen 🐧 (Gnome)

Die Konfiguration ist sehr einfach. Wählen Sie einfach eine Aktion aus und drücken dann auf [**Ändern**], [**Keins setzen**] oder [**Voreinstellung**]. Wenn Sie ihre Einstellung gefunden haben, kann diese als XML-Datei abgespeichert und somit auch für eine andere QGIS Installation genutzt werden.

### 8.2 Hilfe

Wenn Sie Hilfe zu bestimmten Themen brauchen erreichen Sie die Kontext-Hilfe über den [**Hilfe**] Knopf der Ihnen in den meisten Dialogen zur Verfügung steht. Bitte beachten Sie daß Erweiterungen von Drittanbietern zu

dementsprechend gewidmeten Internetseiten verweisen.

### 8.3 Layeranzeige kontrollieren

QGIS rendert standardmäßig alle sichtbaren Layer wenn das Kartenfenster aktualisiert werden muss. Die Abläufe, die eine 'Erneuerung' der Wiedergabe verursachen sind:

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

QGIS ermöglicht Ihnen den Darstellungsprozess in einer Reihe von Möglichkeiten zu steuern.

#### 8.3.1 Maßstabsabhängige Layeranzeige

Die maßstabsabhängige Wiedergabe erlaubt es, einen mini- und maximalen Maßstab, in dessen Rahmen ein Layer angezeigt wird, anzugeben. Um die maßstabsabhängige Wiedergabe einzustellen öffnen Sie den *Eigenschaften* Dialog eines Layers, indem Sie diesen in der Legende doppelt anklicken, im Menü *Allgemein* die entsprechenden Einstellungen vornehmen und das Kontrollkästchen  *Maßstabsabhängige Sichtbarkeit* aktivieren.

Sie können die Maßstabswerte bestimmen indem Sie erst auf den Level den Sie benutzen wollen zoomen und den Maßstabswert in der QGIS Statusleiste notieren.

#### 8.3.2 Layeranzeige kontrollieren

Die Kartendarstellung kann wie unten beschrieben auf verschiedene Art und Weise gesteuert werden.

##### Wiedergabe unterdrücken

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

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

Wenn Sie die  *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 erst einmal nicht angezeigt werden. Um diese Option zu nutzen deaktivieren Sie in der Menüleiste unter *Einstellungen* → *Optionen* im Menü *Darstellung* das Kontrollkästchen  *Normalerweise werden alle neuen Layer im Kartenfenster angezeigt*. Dadurch werden alle neu geladenen Layer erstmal nicht angezeigt.

## Zeichnen stoppen

Um die Wiedergabe einer Karte zu stoppen drücken Sie die ESC Taste. Dies stoppt die Erneuerung der Wiedergabe und die Karte wird nur zum Teil angezeigt. Es kann sein dass etwas Zeit zwischen dem Drücken der ESC Taste und dem Stoppen der Kartenwiedergabe vergeht.

---

**Bemerkung:** Es ist derzeit nicht möglich das Zeichnen zu stoppen - diese Funktion wurde mit der Version qt4 ausgeschaltet da es zu Problemen und Abstürzen der Benutzeroberfläche kam.

---

## 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* im Menü *Darstellung* die *Anzahl von Objekten, nach deren Zeichnung die Anzeige aktualisiert werden soll*. Der Wert 0 (Standard) verhindert die Aktualisierung während des Zeichnens. Bei zu niedrigem Wert leidet die Leistung, das ständig ein Update der Wiedergabe stattfindet. Zu Beginn ist oftmals ein Wert von 500 sinnvoll.

## Die Qualität der Wiedergabe beeinflussen

Um die Wiedergabequalität der Layer zu beeinflussen gibt es zwei Möglichkeiten. Öffnen Sie das Menü *Darstellung* im Menü *Einstellungen* → *Optionen* und aktivieren bzw. deaktivieren Sie folgende Kontrollkästchen.

- *Linien auf Kosten der Zeichengeschwindigkeit weniger gezackt zeichnen*
- *Problem mit falsch gefüllten Polygonen beheben*

## Darstellung beschleunigen

Es gibt zwei Einstellungen die es Ihnen ermöglichen die Darstellungsgeschwindigkeit zu erhöhen. Öffnen Sie den QGIS *Optionen* Dialog indem Sie *Einstellungen* → *Optionen* benutzen, gehen Sie zum *Darstellung* Menü und aktivieren oder deaktivieren Sie die folgenden Kontrollkästchen:


- *Hintergrundpuffer aktivieren*. Dies sorgt für eine bessere Grafikleistung wobei dann das Darstellen dann nicht mehr gestoppt und Objekte nicht mehr stufenweise dargestellt werden können. Wenn es deaktiviert ist können Sie die *Anzahl von Objekten nach deren Zeichnung die Anzeige aktualisiert werden soll*  festlegen, sonst ist diese Option nicht aktiv.
- *Wo möglich den Darstellungscache benutzen, um das Neuzeichnen zu beschleunigen*

## 8.4 Messen

Das Messen funktioniert mit Projektionen (z.B. UTM) und auch mit nicht projizierten Daten. Wenn die geladene Karte als Geografisches Koordinatensystem abgebildet wird (z.B. Latitude/Longitude) sind die Messergebnisse falsch. Sie müssen dann ein passendes KBS mit einer Projektion verwenden (siehe Abschnitt *Arbeiten mit Projektionen*). Alle Meßwerkzeuge verwenden auch die eingestellte Fangtoleranz der Digitalisierung. Dies ist zum Beispiel dann sinnvoll, wenn Sie entlang von Linien oder Polygongrenzen geladener Vektorlayer messen wollen.

Um ein Meßwerkzeug auszuwählen klicken Sie auf  und suchen das Werkzeug das Sie benutzen wollen aus.

## 8.4.1 Strecken, Flächen und Winkel messen

 **Linie messen:** QGIS kann reale Distanzen zwischen Punkten auf Basis eines vorgegebenen Ellipsoiden messen. Dazu wählen Sie im Menü *Einstellungen* → *Optionen* im Menü *Kartenwerkzeuge* einen Ellipsoiden aus. Es kann außerdem die Gummibandfarbe, die bevorzugte Maßeinheit (Meter oder Fuß) und das bevorzugte Winkelmaß (Grad, Bogenmaß, Gon) eingestellt werden. Das Werkzeug 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

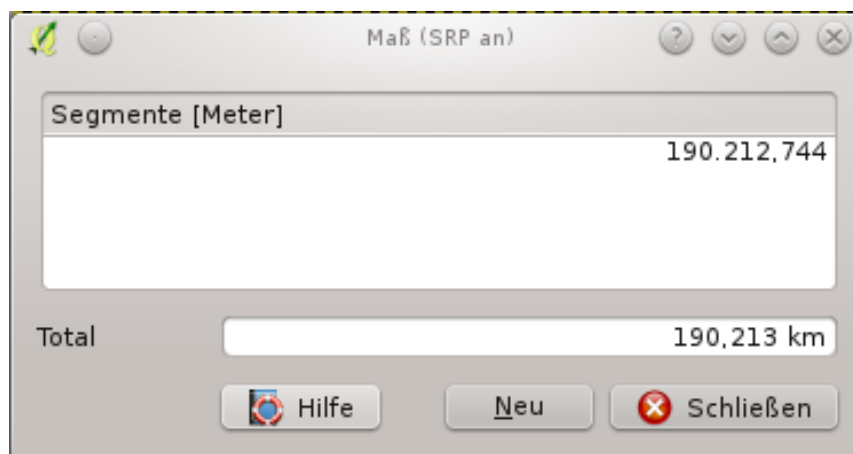



Abbildung 8.2: Entfernungen messen  (Gnome)




 **Fläche messen:** Auch Flächengrößen können gemessen werden. Das Popup zeigt dabei die akkumulierte Flächengröße. Zusätzlich schnappt das Meßwerkzeug auf den gerade ausgewählten Layer wenn für den entsprechenden Layer ein Fangmodus eingestellt wurde (siehe Abschnitt *Einstellen der Fangtoleranz und des Suchradius*). Wenn Sie also exakt entlang einer Linie oder Polygonumrandung messen möchten, setzen Sie erst eine Fangtoleranz und selektieren dann den Layer. Wenn Sie jetzt das Meßwerkzeug benutzen schnappt man mit jedem Mausklick (innerhalb der eingestellten Toleranz) auf den Layer.




Abbildung 8.3: Flächen messen  (Gnome)

 **Winkel messen:** Auch Winkel können mit dem Winkel messen- Werkzeug gemessen werden. Der Mauszeiger wird zum Kreuz. Drücken Sie die linke Maustaste und zeichnen Sie das erste Segment des Winkels, klicken nochmals auf die linke Maustaste und messen sie den Winkel, der im Dialogfenster angezeigt wird.

## 8.4.2 Objekte auswählen und Auswahl aufheben

Die Werkzeugleiste bietet mehrere Werkzeuge, um Objekte im Kartenfenster auszuwählen. Um ein oder mehrere Objekte auszuwählen, klicken Sie auf  und wählen Sie dann das entsprechende Werkzeug:

-  Einzelnes Objekt wählen

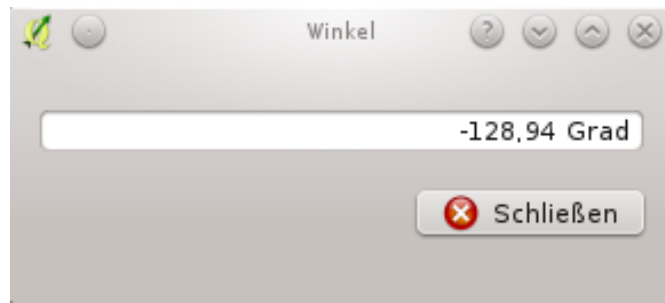








Abbildung 8.4: Winkel messen  (Gnome)


-  Objekte durch Rechteck wählen
-  Objekte durch Polygon wählen
-  Objekte freihändig wählen
-  Objekte durch Radius wählen

Um die Auswahl von Objekten wieder aufzuheben, klicken Sie auf den Knopf .

 Objekte mit einem Ausdruck wählen ermöglicht es dem Anwender ein Objekt anhand eines Ausdruckdialoges auszuwählen. Siehe auch die Beispiele in Kapitel [Ausdrücke](#).

Anwender können eine Objektauswahl in **Neuer Memory-Vektorlayer** oder **Neuer Vektorlayer** anhand von *Bearbeiten* → *Objekte einfügen als* → speichern indem Sie den gewünschten Modus auswählen.

## 8.5 Objekte abfragen



Das Objekte abfragen Werkzeug ermöglicht es Ihnen mit dem Kartenfenster zu interagieren und Informationen über Objekte über eine Pop-Up Fenster abzurufen. Um ein Objekt abzufragen verwenden Sie *Ansicht* → *Objekte abfragen*, drücken Sie `Strg + Umschalt + I` oder klicken Sie auf das  Objekte abfragen Icon in der Werkzeugleiste.

Wenn Sie auf mehrere Objekte klicken gibt der *Identifikationsergebnis* Dialog die Informationen über die ausgewählten Objekte in einer Liste wieder. Das erste Element ist die Nummer des Objekts in der Ergebnisliste, gefolgt vom Layernamen. Das erste Untermenü ist der Name eines Feldes mit seinem Wert. Schließlich werden alle Informationen über das Objekt abgebildet.

Dieses Fenster kann angepasst werden um benutzerdefinierte Felder anzuzeigen, standardmäßig gibt es drei Arten von Informationen wieder:

- **Aktionen:** Aktionen können zum Objekte abfragen Fenster hinzugefügt werden. Wenn Sie dann auf das Aktionen-Icon klicken wird eine Aktion ausgeführt. Standardmäßig wird nur eine Aktion hinzugefügt um das Objektformular zum Bearbeiten darzustellen.
- **Abgeleitet:** Diese Information wird von anderen Informationen berechnet oder abgeleitet. Sie können hier durch einen Klick X- und Y-Koordinate, Fläche in Karteneinheiten und Umfang in Karteneinheiten von Polygonen, Längen in Karteneinheiten von Linien und Objekt IDs herausfinden.
- **Datenattribute:** Dies ist die Liste von Attributfeldern aus den Daten.

Am unteren Rand des Fensters gibt es fünf Icons:

-  Baum ausklappen
-  Baum einklappen

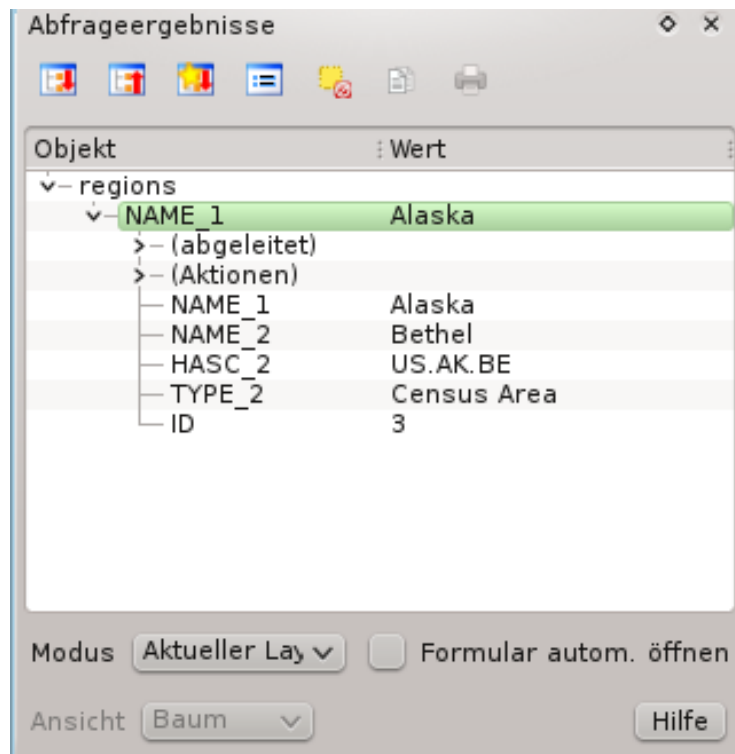






Abbildung 8.5: Objekte abfragen Dialog  (Gnome)

-  Neue Ergebnisse erscheinen automatisch ausgeklappt.
-  Gewählte Objekte in Zwischenablage kopieren.
-  Gewählte HTML-Antwort drucken

Andere Funktionen können im Kontextmenü des abgefrageten Objekts gefunden werden. Im Kontextmenü können Sie z.B.:


- Das Objektformular anzeigen
- Zum Objekt zoomen
- Objekt kopieren: Kopieren Sie alle Objektgeometrien und -attribute
- Auswahl des Objekts umkehren: fügt das ausgewählte Objekt der Auswahl hinzu
- Attributwert kopieren: Kopieren Sie nur den Wert des Attributes auf das Sie klicken
- Objektattribute kopieren: Kopieren Sie nur die Attribute
- Ergebnisse löschen: Löschen Sie Ergebnisse im Fenster
- Hervorhebungen löschen: Entfernen Sie in der Karte hervorgehobene Objekte
- Alle hervorheben
- Layer hervorheben
- Layer aktivieren: Wählen Sie einen Layer der aktiviert werden soll
- Layereigenschaften ... : Öffnen Sie das Layereigenschaften Fenster
- Alles ausklappen
- Alles zusammenfallen



## 8.6 Dekorationen

Die Dekorationen von QGIS umfassen das Gitter, den Urheberrechtshinweis, den Nordpfeil und die Maßstabseiste. Sie werden zum Dekorieren der Karte durch Hinzufügen von kartografischen Elementen benutzt.

### 8.6.1 Gitter

 Gitter ermöglicht es Ihnen ein Koordinatengitter und Koordinatenbeschriftungen der Karte hinzuzufügen.

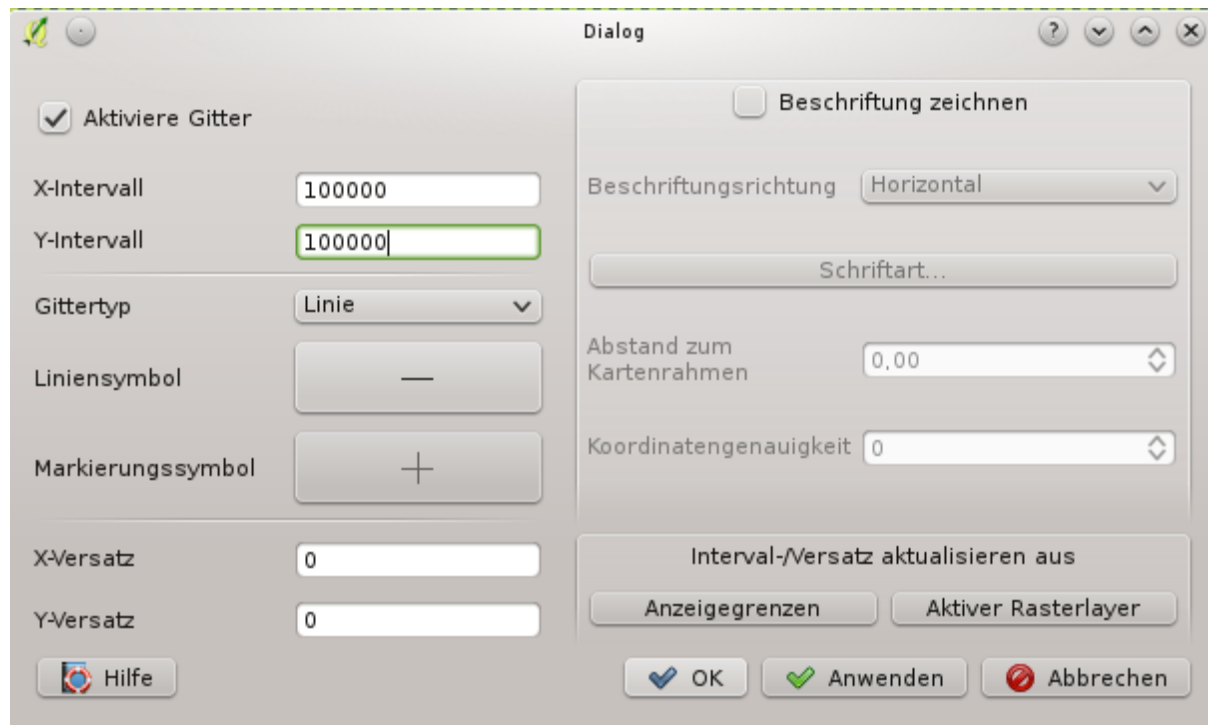



Abbildung 8.6: Der Gitter Dialog 

1. Wählen Sie das Menü *Ansicht* → *Dekorationen* → *Gitter*. Der Dialog öffnet sich (siehe [figure\\_decorations\\_1](#)).
2. Aktivieren Sie das Kontrollkästchen  *Aktiviere Gitter* und machen Sie Definitionen gemäß den Layern die Sie ins Kartenfenster geladen haben.
3. Aktivieren Sie das Kontrollkästchen  *Beschriftungen zeichnen* und machen Sie Beschriftungsdefinitionen gemäß den im Kartenfenster geladenen Layern.
4. Klicken Sie [**Anwenden**] um sicherzugehen dass es so aussieht wie sie es erwarten.
5. Klicken Sie [**OK**] um den Dialog zu schließen.

### 8.6.2 Urheberrechtshinweis

 Urheberrechtshinweis fügt einen Urheberrechtshinweis mit dem Text Ihrer Wahl in Ihr Kartenfenster ein.

1. Wählen Sie *Ansicht* → *Dekorationen* → *Urheberrechtshinweis*. Der Dialog öffnet sich (siehe [figure\\_decorations\\_2](#))
2. Geben Sie den Text den Sie auf der Karte platzieren wollen ein. Sie können wie im Beispiel gezeigt HTML benutzen.

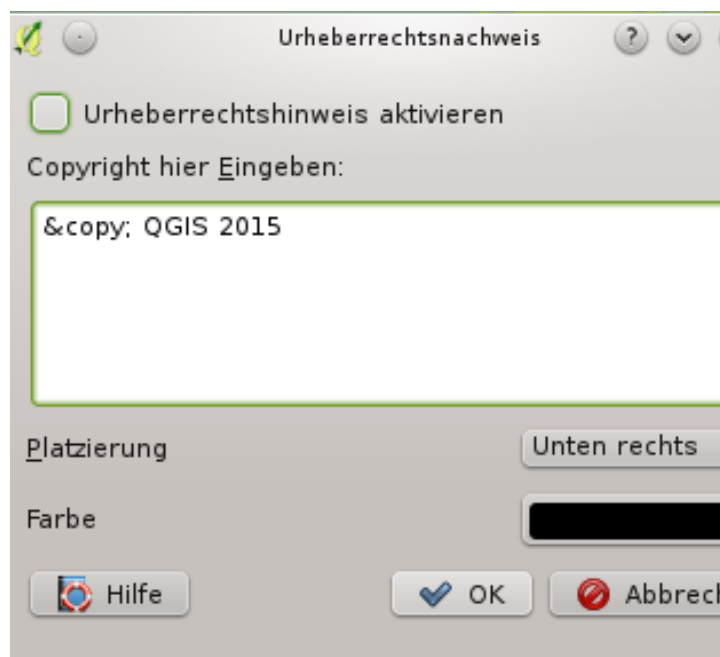





Abbildung 8.7: Der Urheberrechtshinweis-Dialog 

3. Wählen Sie die Platzierung der Beschriftung in der *Platzierung*  Kombobox aus.
4. Versichern Sie sich dass das  *Urheberrechtshinweis aktivieren* Kontrollkästchen aktiviert ist.
5. Klicken Sie [OK].

Im oben genannten Beispiel, das den Standard darstellt, platziert QGIS einen Urheberrechtshinweissymbol gefolgt vom Datum in der rechten unteren Ecke des Kartenfensters.

### 8.6.3 Nordpfeil



 *Nordpfeil* platziert einen einfachen Nordpfeil im Kartenfenster. 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, entscheidet das Programm, in welche Richtung der Nordpfeil idealerweise zeigen sollte. Zum Platzieren des Pfeils haben Sie vier Möglichkeiten entsprechend den vier Ecken des Kartenfensters.

### 8.6.4 Maßstab

 *Maßstab* fügt einen einfachen Maßstab in das Kartenfenster ein. Sie können Maßstabstil und Platzierung so wie die Beschriftung des Balkens steuern.

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.

Das Hinzufügen eines Maßstabbalkens geht folgendermaßen:

1. Klicken Sie auf *Ansicht* → *Dekorationen* → *Maßstab*. Der Dialog startet (siehe [figure\\_decorations\\_4](#)).
2. Wählen Sie die Platzierung aus der *Platzierung*  Kombobox aus.
3. Wählen Sie den Stil aus der *Maßstabstil*  Kombobox aus.

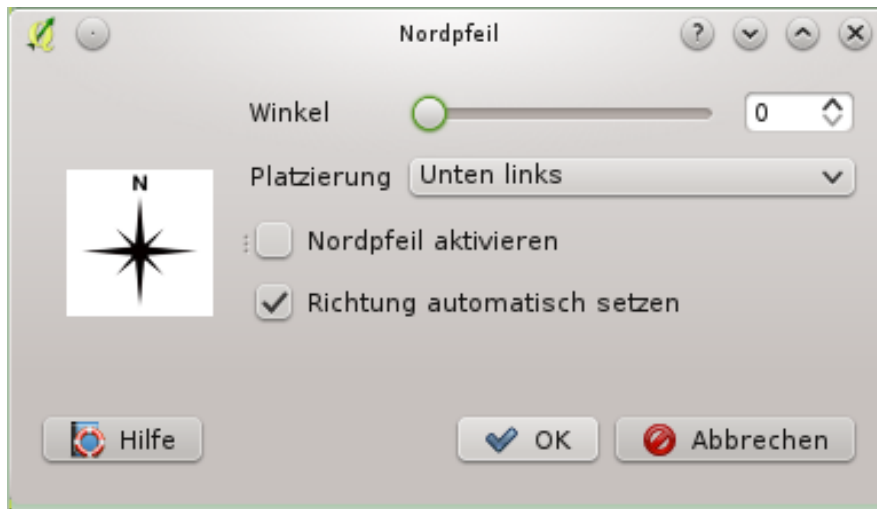



Abbildung 8.8: Der Nordpfeil Dialog 

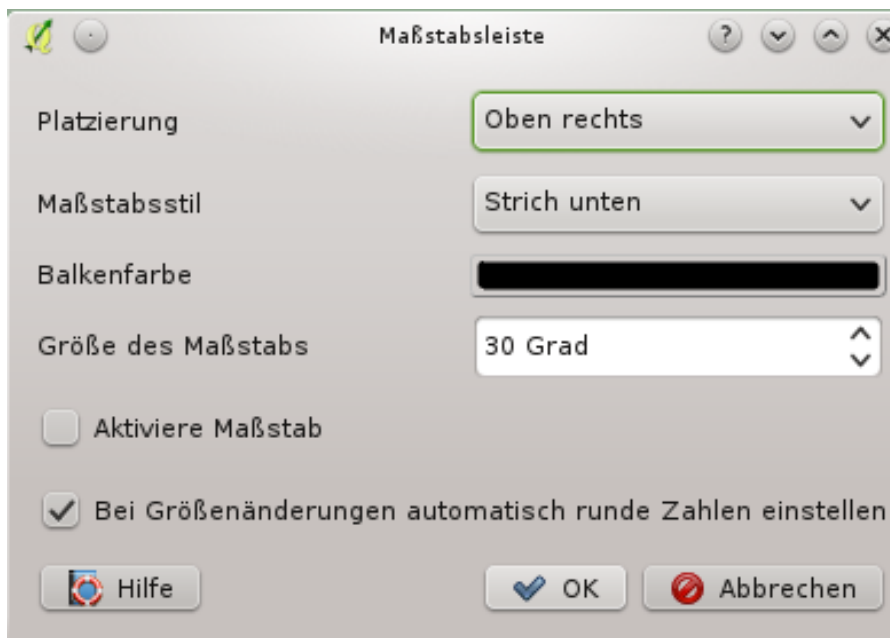



Abbildung 8.9: Der Maßstab Dialog 

4. Wählen Sie die Farbe für den Balken unter *Balkenfarbe* Border [ ] oder benutzen Sie die standardmäßig eingestellte schwarze Farbe.
5. Legen Sie die Größe des Balkens und seine Beschriftung unter *Größe des Maßstabs* 1,00 fest.
6. Versichern Sie sich dass das  *Aktiviere Maßstab* Kontrollkästchen aktiviert ist.
7. Gegebenenfalls können Sie  *Bei Größenänderungen automatisch runde Zahlen einstellen* aktivieren.
8. Klicken Sie [OK].

**Tipp: Einstellungen von Dekorationen**

Wenn Sie ein .qgs -Projekt speichern werden alle bezüglich Nordpfeil, Maßstab und Urheberrechtshinweis gemachten Änderungen im Projekt gespeichert und das nächste Mal wenn das Projekt geladen wird wiederhergestellt

## 8.7 Beschriftungstools

Das  *Beschriftungstext*-Werkzeug in der Attribut-Werkzeuleiste bietet die Möglichkeit formatierten Text in einer Sprechblase im QGIS Kartenfenster zu platzieren. Aktivieren Sie dazu einfach das Beschriftungstext-Werkzeug und klicken Sie in das Kartenfenster.

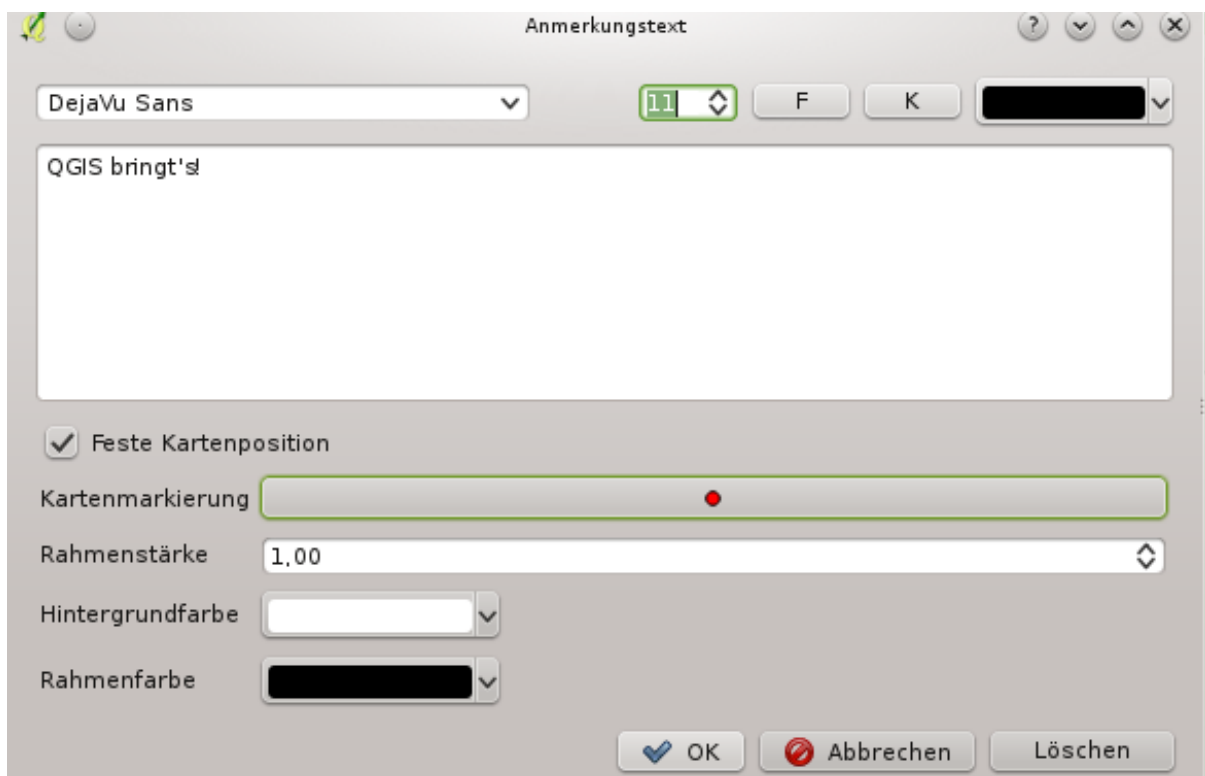




Abbildung 8.10: Anmerkungstext Dialog 


Ein Doppelklick auf das Element öffnet den Dialog mit verschiedenen Optionen. Im Texteditor können Sie den formatierten Text eingeben und einige Einstellungen vornehmen. Z.B. kann das Objekt an einer bestimmten Stelle positioniert werden, was dann durch ein Markersymbol angezeigt wird. Oder das es unabhängig von dem Objekt im Kartenfenster angezeigt wird. Der Text kann entweder durch Verschieben der Markierung oder der Sprechblase im Kartenfenster verschoben werden.

Das  **Beschriftung verschieben** Werkzeug ermöglicht es Ihnen die Beschriftung auf den Kartenfenster zu verschieben.


### 8.7.1 HTML-Beschriftung

Die  **HTML-Beschriftung** Werkzeuge in der Attribut-Werkzeugleiste ermöglichen es den Inhalt einer HTML-Datei in einer Sprechblase auf dem QGIS-Kartenfenster zu platzieren. Benutzen Sie das *HTML-Beschriftung* Werkzeug indem Sie ins Kartenfenster klicken und den Pfad zur HTML-Datei in den Dialog einfügen.

### 8.7.2 SVG-Anmerkung

Das  **SVG-Anmerkung** Werkzeug in der Attribut-Werkzeugleiste ermöglicht es ein SVG-Symbol in einer Sprechblase auf dem QGIS-Kartenfenster zu platzieren. Um das *SVG-Anmerkung* Werkzeug zu benutzen klicken Sie in das Kartenfenster und fügen Sie den Pfad zur SVG-Datei in den Dialog ein.

### 8.7.3 Beschriftungsformular

Zusätzlich können Sie eigene Beschriftungsformulare erstellen. Das  **Beschriftungsformular** Werkzeug ist nützlich um Attribute eines Vektorlayers in einem maßgeschneiderten QT-Designer Beschriftungsformular anzuzeigen (siehe `figure_custom_annotation`). Es ist ähnlich wie die Designer-Formulare für das Identifizierungswerkzeug, wird aber als Beschriftungstext angezeigt. Siehe auch das Video <https://www.youtube.com/watch?v=0pDBuSbQ02o> von Tim Sutton für weitere Informationen.

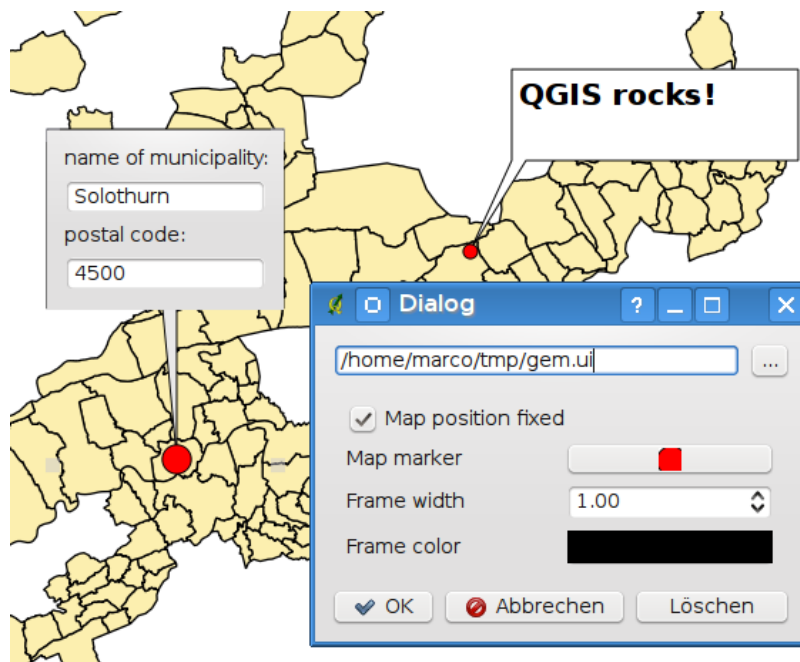


Abbildung 8.11: Benutzerfreundliches QT Designer Beschriftungsformular 

---

**Bemerkung:** Wenn Sie das Tastenkürzel `Ctrl+T` drücken während ein Beschriftungswerkzeug aktiv ist ändert sich deren Anzeigestatus zwischen an und aus.

---

## 8.8 Räumliche Lesezeichen

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

### 8.8.1 Ein Lesezeichen erstellen

Um ein Lesezeichen zu erstellen:

1. Zoomen oder verschieben Sie das Kartenfenster in einen Ausschnitt Ihrer Wahl.
2. Wählen Sie in der Menüleiste *Ansicht* → *Neues Lesezeichen* oder drücken Sie `kbd:Ctrl-B`.
3. Geben Sie eine Beschreibung für das Lesezeichen ein (bis zu 255 Zeichen).
4. Klicken Sie auf `Enter` um das Lesezeichen hinzuzufügen oder auf **[Löschen]** um das Lesezeichen zu entfernen.

Sie können mehrere Lesezeichen mit demselben Namen erstellen.

### 8.8.2 Arbeiten mit Lesezeichen

Um räumliche Lesezeichen zu verwenden oder zu verwalten wählen Sie die Menüoption *Ansicht* → *Lesezeichen anzeigen*. Mit dem *Räumliches Lesezeichen* Dialog können Sie zu einem räumlichen Lesezeichen zoomen oder ein Lesezeichen löschen. Sie können hier den Namen und die Koordinaten des Lesezeichens bearbeiten.

### 8.8.3 Auf ein Lesezeichen zoomen

Im Dialogfenster *Räumliches Lesezeichen* 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.

### 8.8.4 Ein Lesezeichen löschen


Um ein Lesezeichen aus dem *Räumliches Lesezeichen* Dialog zu löschen klicken Sie es an und klicken Sie dann **[Löschen]**. Bestätigen Sie Ihre Wahl indem Sie **[OK]** klicken oder brechen Sie das Löschen mit **[Abbrechen]** ab.

## 8.9 Layer/Gruppen einbinden

Wenn Sie Inhalte von anderen Projekten in Ihr Projekt einbinden wollen wählen Sie *Layer* → *Eingebettete Layer und Gruppen ...*

### 8.9.1 Layer einbetten

Mit dem folgenden Dialog können Sie Layer von anderen Projekten einbetten. Hier ein kleines Beispiel:


1. Drücken Sie  um nach einem anderen Projekt des Alaskadatensatzes zu suchen.
2. Wählen Sie die Projektdatei `grassland`. Sie können hier den Inhalt des Projektes einsehen (siehe [figure\\_embed\\_dialog](#)).
3. Klicken Sie `Strg` und klicken Sie auf die Layer `grassland` und `regions`. Klicken Sie **[OK]**. Die ausgewählten Layer sind jetzt in Legende und Kartenfenster eingebettet.

Währenddessen die Layer editierbar sind können Sie Ihre Eigenschaften wie Darstellung und Beschriftung nicht ändern.



Abbildung 8.12: Einzubettende Layer und Gruppen wählen 

### 8.9.2 Eingebettete Layer entfernen

Machen Sie einen Rechtsklick auf den eingebetteten Layer und wählen Sie  Entfernen.





---

## QGIS Konfiguration

---

Sie können QGIS auf vielfache Art und Weise im Menü *Einstellungen* → anpassen. Wählen Sie zwischen Bedienfeldern, Werkzeugkästen, Projekteinstellungen, Optionen und Anpassungen.

**Bemerkung:** QGIS befolgt die Desktopvorgaben zum Lokalisieren der Optionen und Projekteigenschaften. Im Zusammenhang mit dem von Ihnen verwendeten Betriebssystem können die oben Beschriebenen Dialoge entweder im Menü *Ansicht* (Bedienfelder und Werkzeugleisten) oder in *Projekt* für Optionen gefunden werden.

---

### 9.1 Bedienfelder und Werkzeugkästen

Im *Bedienfelder* → Menü können Sie QGIS Widgets an- und ausschalten. Das *Werkzeugkäste* → Menü bietet Ihnen die Möglichkeit Icongruppen in der QGIS Werkzeugleiste an- und auszuschalten (siehe [figure\\_panels\\_toolbars](#)).




---

#### Tipp: Die QGIS-Übersicht aktivieren

Sie können in QGIS eine Übersicht, die die Gesamtausdehnung der hinzugefügten Layer wiedergibt, benutzen. Sie kann unter dem Menü *Einstellungen* → *Bedienfelder* ausgewählt werden. Innerhalb des Fensters zeigt ein Rechteck die aktuelle Kartenausdehnung an. Dies ermöglicht es Ihnen schnell zu bestimmen welchen Bereich der Karte Sie gerade sehen. Beachten Sie dass Beschriftungen auch wenn die Layer explizit beschriftet worden sind in der Übersicht nicht dargestellt werden. Wenn Sie auf das Rechteck klicken und es in der Übersicht, die die aktuelle Kartenausdehnung zeigt, verschieben wird das Kartenfenster entsprechend geupdatet.




---

#### Tipp: Protokoll anzeigen lassen

Es ist möglich das QGIS Protokoll zu verfolgen. Sie können  *Protokoll* im Menü  *Einstellungen* → *Bedienfelder* oder  *Ansicht* → *Bedienfelder* aktivieren und die Meldungen, die in den verschiedenen Reitern während des Ladens und im Laufe von Operationen angezeigt werden, verfolgen.

---

### 9.2 Projekteigenschaften

Im Eigenschaftfenster für das Projekt unter  *Einstellungen* → *Projekteinstellungen* (kde) oder   *Projekt* → *Projekteigenschaften* (Gnome) können Sie die projektbezogenen Optionen einstellen. Diese beinhalten:

- Im Menü *Allgemein* den Projekttitle, die Auswahl- und Hintergrundfarbe, Layereinheiten, Genauigkeit und die Option die Pfade der Layer relativ zu speichern. Wenn die KBS-Transformation eingeschaltet ist können Sie einen Ellipsoiden für Abstandsberechnungen auswählen. Sie können die Karteneinheiten definieren (nur wenn die KBS-Transformation ausgeschaltet ist) sowie die Genauigkeit der benutzten Dezimalstellen. Auch können Sie eine Liste von Projektmaßstäben festlegen die die global voreingestellten Maßstäbe überschreibt.

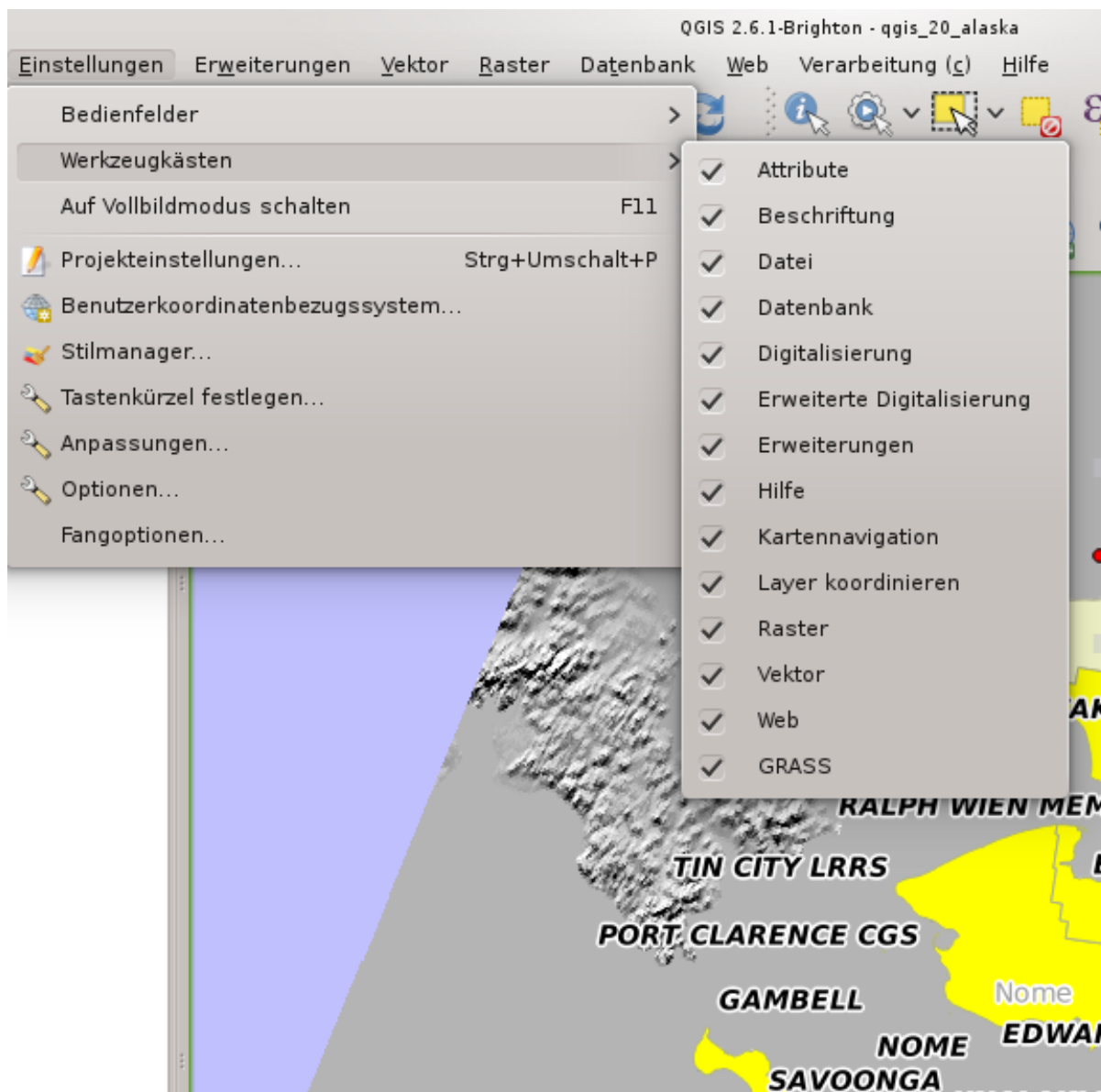



Abbildung 9.1: Das Bedienfelder und Werkzeugkästen Menü 

- Im *KBS* Menü können Sie das Koordinatenbezugssystem für dieses Projekt auswählen und Spontan-Reprojektion für das Darstellen von Vektor- und Rasterlayern mit verschiedenen KBS einschalten.
- Mit dem dritten *Abfragbare Layer* Menü legen Sie fest (oder verhindern) welche Layer auf das ‘Objekte abfragen’-Werkzeug reagieren (siehe der Abschnitt über Kartenwerkzeuge im Abschnitt *Optionen* über das Einschalten der Identifizierung von Mehrfachlayern.).
- Mit dem *Vorgabestile* Menü können Sie steuern wie neue Layer dargestellt werden wenn kein `.qml` Stil definiert wurde. Sie können hier auch den voreingestellten Transparenzlevel bestimmen und einstellen ob Symbolen Zufallsfarben zugewiesen werden sollen. Es gibt auch einen zusätzlichen Abschnitt in dem Sie bestimmte Farben für das laufende Projekt festlegen können. Sie können die hinzugefügten Farben im Drop-down-Menü des Farbdialogfensters, das Sie in jeder Darstellung zur Verfügung steht.
- Im Menü *OWS Server* können Sie Informationen über die QGIS Server WMS- und WFS-Eigenschaften, Ausdehnung und KBS-Beschränkungen definieren.
- Das Menü *Macros* wird verwendet um Python-Makros für Projekte zu bearbeiten. Zur Zeit sind nur drei Makros möglich: `openProject()`, `saveProject()` und `closeProject()`.

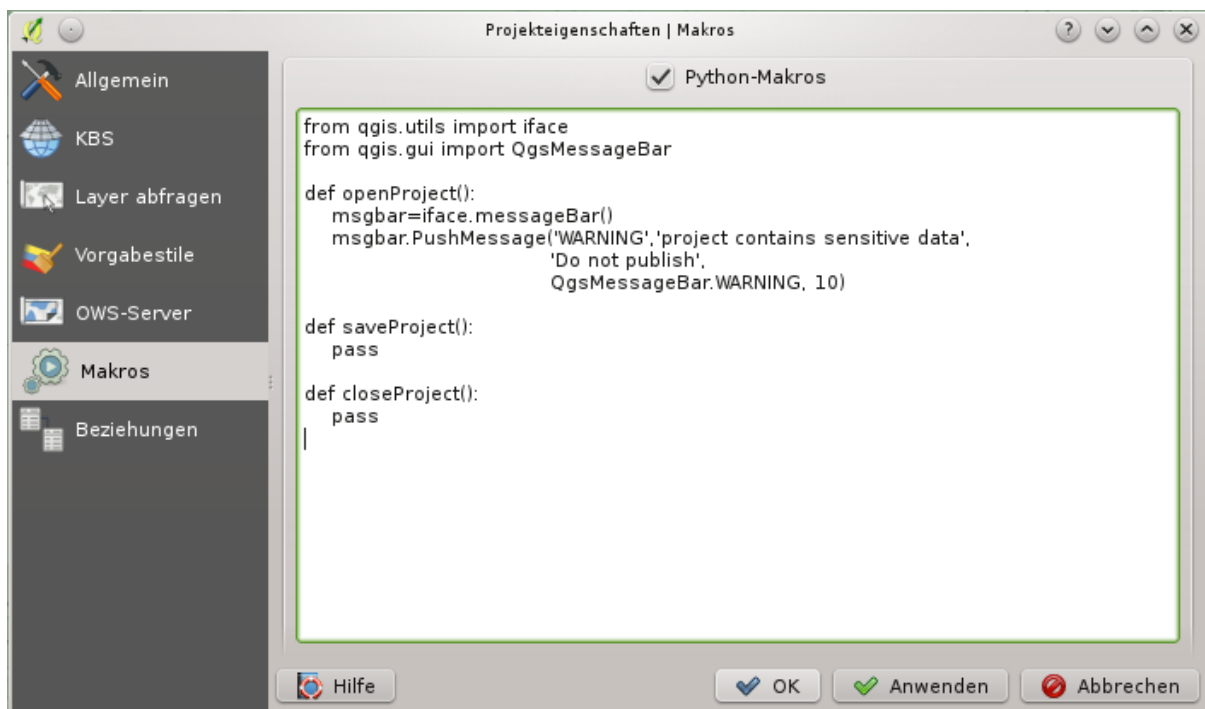


Abbildung 9.2: Makroeinstellungen in QGIS






- Das Menü *Beziehungen* wird verwendet um 1:n-Beziehungen zu definieren. Die Beziehungen werden im Dialog Projekteigenschaften definiert. Sind einmal Beziehungen für einen Layer erstellt erscheint ein neues Schnittstellenelement im Objektformular (z.B. wenn Sie ein Objekt abfragen und sein Formular öffnen) das die damit verknüpften Elemente auflistet. Dies ist eine leistungsfähige Methode um z.B. die Inspektionsgeschichte eines Pipelineabschnittes oder Straßenabschnittes auszudrücken. Sie können mehr über die Unterstützung von 1:n Beziehungen im Abschnitt *Ein-Zu-Mehrere-Beziehungen erstellen* finden.

## 9.3 Optionen




Einige grundlegende Optionen für QGIS können mit dem *Optionen* Dialog gewählt werden. Wählen Sie die Menüoption *Einstellungen* → *Optionen*. Die Menüs mit denen Sie Ihre Optionen optimieren können sind:

### 9.3.1 Menü Allgemein

#### Anwendung

- Wählen Sie *Stil* (QGIS-Neustart erforderlich)  und wählen Sie zwischen 'Oxygen', 'Windows', 'Motif', 'CDE', 'Plastique', 'GTK+' und 'Cleanlooks' .
- Definieren Sie das *Icon-Thema* . Derzeit ist nur 'default' möglich.
- Definieren Sie die *Icon-Größe* .
- Definieren Sie die *Schriftart*. Wählen Sie zwischen  *Qt-Voreinstellung* und einer benutzerdefinierten Schriftart.
- Verändern Sie *Dauer von zeitweiligen Meldungen oder Dialogen* .
- *Splashscreen beim Start nicht anzeigen*
- *Tipps beim Starten anzeigen*
- *Gruppenrahmentitel fett*
- *Gruppenrahmen im QGIS-Stil*
- *Farbauswahldialoge verwenden, die laufend aktualisieren*

#### Projektdateien

- *Projekt beim Starten öffnen*  (wählen Sie zwischen 'Neu', 'Zuletzt verwendetes' und 'Bestimmtes'). Wenn Sie 'Bestimmtes' wählen benutzen Sie den  Knopf um ein Projekt festzulegen.
- *Neues Projekt aus Vorgabeprojekt erstellen*. Sie haben die Möglichkeit auf *Aktuelles Projekt als Vorgabe speichern* oder *Vorgabe zurücksetzen* zu drücken. Sie können durch Ihre Dateien gehen und ein Verzeichnis definieren in dem Sie ihre benutzerdefinierten Projektvorlagen festlegen. Diese werden dann dem Menü *Projekt* → *Neu aus Vorlage* hinzugefügt wenn Sie erst  *Neues Projekt aus Vorgabeprojekt erstellen* aktivieren und dann ein Projekt in den Projektvorlagenordner speichern.
- *Bei Bedarf nachfragen, ob geänderte Projekt und Datenquellen gespeichert werden sollen*
- *Warnung ausgeben wenn QGIS-Projekt einer früheren Version geöffnet wird*
- *Makros aktivieren* . Diese Option wurde erstellt um Makros zu handhaben die geschrieben wurden um eine Aktion auf Projekt ereignisse auszuführen. Sie können zwischen 'Nie', 'Fragen', 'Nur in dieser Sitzung' und 'Immer (nicht empfohlen)' wählen.

### 9.3.2 Menü System

#### Umgebung

Systemumgebungsvariablen können nun eingesehen werden und viele von ihnen konfiguriert in der **Umgebung** Gruppe (siehe [figure\\_environment\\_variables](#)). Dies ist für Betriebssysteme wie Mac nützlich da dort eine GUI Anwendung nicht notwendigerweise die Shellumgebung des Benutzers übernimmt. Sie ist auch nützlich um Umgebungsvariablen für externe Werkzeuge die von der Verarbeitungs-Toolbox (z.B. SAGA, GRASS) kontrolliert werden zu setzen und zu visualisieren und um die Debug-Ausgabe für bestimmte Abschnitte des Quellcodes anzuschalten.

- *Benutzerdefinierte Umgebungsvariablen verwenden (Neustart erforderlich - Trennzeichen einschließen)*. Sie können Variablen [**Hinzufügen**] und [**Entfernen**]. Bereits definierte Umgebungsvariablen sind unter *Aktuelle Umgebungsvariablen* dargestellt und es ist möglich sie zu filtern indem Sie  *Nur QGIS-spezifische Variablen anzeigen* aktivieren.

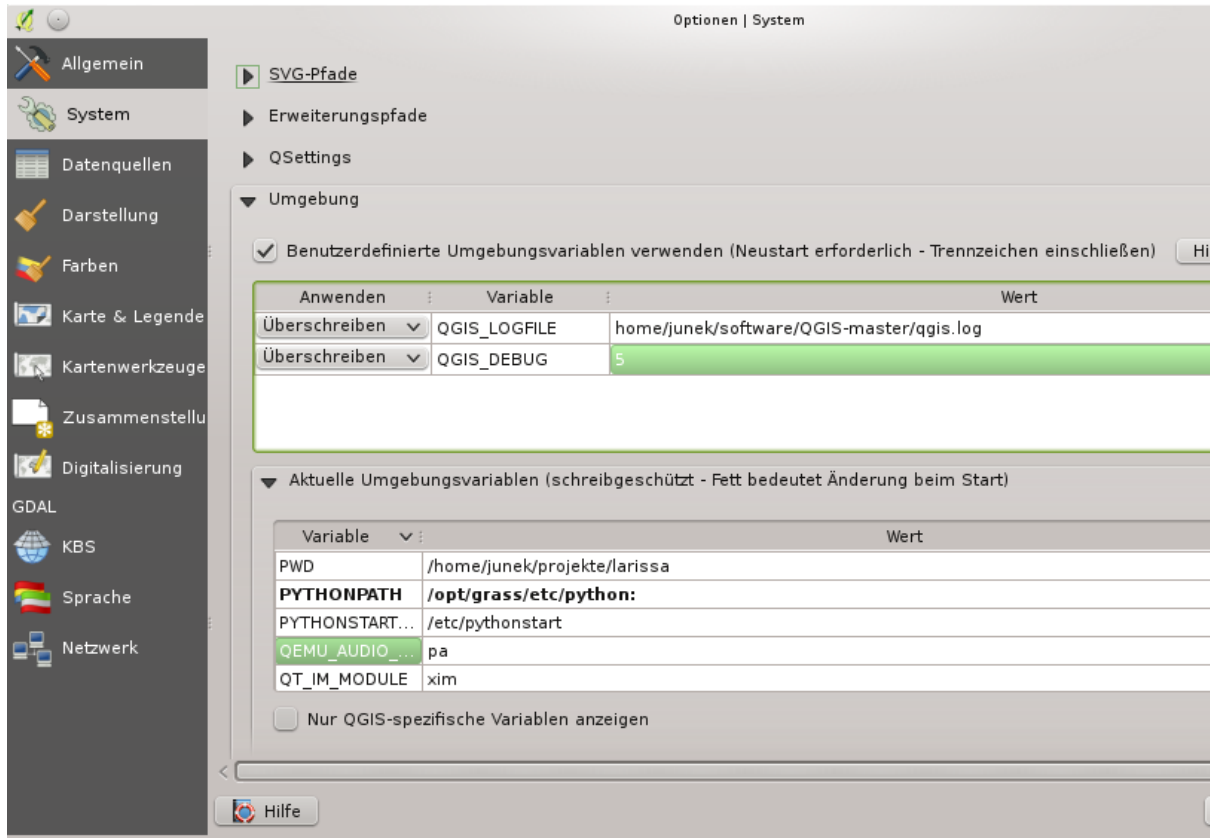


Abbildung 9.3: Systemumgebungsvariablen in QGIS

### Erweiterungspfade

[Hinzufügen] und [Entfernen] von *Nach zusätzlichen C++ Erweiterungsbibliotheken zu durchsuchende Pfad(e)*


## 9.3.3 Menü Datenquellen

### Objektattribute und -tabelle

- *Attributtabelle gedockt öffnen (QGIS-Neustart erforderlich)*
- *Geometrie in WKT aus Attributtabelle kopieren.* Wenn Sie *Ausgewählte Zeilen in die Zwischenablage kopieren* aus dem *Attributtabelle* Dialog verwenden, hat dies zum Ergebnis dass auch die Koordinaten von Punkten und Stützpunkten auch in die Zwischenablage kopiert werden.
- *Attributtabellenverhalten* . Es gibt drei Möglichkeiten: 'Alle Objekte anzeigen', 'Alle gewählten Objekte anzeigen' und 'Sichtbare Objekte der Karte anzeigen'
- *Attributtabellenzeilencache* . Dieser Zeilencache erlaubt es, die N zuletzt geladenen Attributzeilen im Cache zu speichern. Dadurch wird das Arbeiten mit der Attributtabelle schneller. Der Cache wird wieder gelöscht, wenn man die Attributtabelle schliesst.
- *Repräsentation für NULL-Werte.* Hier können Sie einen Wert für Datenfelder die einen NULL-Wert enthalten festlegen.

### Datenquellenbehandlung

- *Nach gültigen Element im Browser suchen* . Sie können zwischen 'Erweiterung prüfen' und 'Dateiinhalt prüfen' wählen.

- *Inhalt komprimierter Dateien (.zip) im Browser durchsuchen* . 'Nein', 'Grundsuche' und 'Vollsuche' sind möglich.
- *Beim Öffnen nach Rasterunterlayern fragen.* Einige Rasterformate unterstützen Unterlayer - sie werden Subdataset in GDAL genannt. Ein Beispiel sind netCDF-Dateien - wenn es viele netCDF-Variablen gibt interpretiert GDAL jede Variable als Subdataset. Die Option ermöglicht es Ihnen zu steuern wie mit Unterlayern umgegangen wird wenn eine Datei mit Unterlayern geöffnet wird. Sie haben die folgenden Wahlmöglichkeiten:
  - 'Immer': Immer fragen (ob es Unterlayer gibt)
  - 'Wenn nötig': Fragen ob Layer keine Bänder aber Unterlayer hat
  - 'Nie': Nie fragen, lädt dann nichts
  - 'Alle laden': Nie auffordern aber alle Unterlayer laden
- *Shapefile-Kodierungsangabe ignorieren.* Wenn eine Shapedatei Kodierungsinformationen enthält wird dieses von QGIS ignoriert.
- *PostGIS-Layer per Doppelklick hinzufügen und zur Auswahl den erweiterten Modus verwenden*
- *Oracle-Layer mit Doppelklick hinzufügen und zur Auswahl den erweiterten Modus verwenden*

### 9.3.4 Menü Darstellung

#### Zeichenverhalten

- *Normalerweise werden alle neuen Layer im Kartenfenster angezeigt*
- *Wo möglich den Darstellungscache benutzen, um das Neuzeichnen zu beschleunigen*
- *Layer mit vielen CPU-Kernen parallel zeichnen*
- *Max. zu benutzende Kerne*
- *Kartenaktualisierungsintervall*
- *Geometrievereinfachung für neue Layer voreinstellen*
- *Vereinfachungsschwelle*
- *Auf Datenlieferantenebene vereinfachen, wenn möglich*
- *Größter Maßstab bis zu dem der Layer vereinfacht werden soll*




#### Zeichenqualität


- *Linie auf Kosten der Zeichengeschwindigkeit weniger gezackt zeichnen*

#### Raster

- Mit *RGB Kanalauswahl* können Sie Nummer für den Roten Kanal, Grünen Kanal und Blauen Kanal festlegen.

#### Kontrasverbesserung

- *Einkanalgraustufen* . Für ein Einkanalgraustufenband kann eingestellt werden: 'Kein Stretch', 'Strecken auf MinMax', 'Strecken und Zuschneiden auf MinMax' und 'Zuschneiden auf 'MinMax'.
- *Multikanalfarbe (Byte/Kanal)* . Optionen sind 'Kein Strecken', 'Strecken auf MinMax', 'Strecken und Zuschneiden auf MinMax' und 'Zuschneiden auf MinMax'.
- *Multikanalfarbe (>Byte/Band)* . Optionen sind 'Kein Strecken', 'Strecken auf MinMax', 'Strecken und Zuschneiden auf MinMax' und 'Zuschneiden auf MinMax'.

- *Grenzen (Minimum/Maximum)* . Optionen sind 'Kumulativer Pixelanzahl-Schnitt', 'Minimum/Maximum', 'Mittlere +/- Standardabweichung'
- *Kommulative Pixelanzahl-Schnittgrenzen*
- *Standardabweichungsfaktor*

#### Fehlersuche

- *Kartenaktualisierung*

### 9.3.5 Menü Farben


Dieses Menü ermöglicht es Ihnen benutzerdefinierte Farben hinzuzufügen die Sie in jedem Farbdialogfenster der Darstellungen finden können. Sie sehen in dem Reiter eine Reihe von vordefinierten Farben: Sie können alle löschen oder bearbeiten. Darüberhinaus können Sie eine Farbe Ihrer Wahl hinzufügen und eine Kopieren und Einfügen Operation durchführen. Schließlich können Sie den Farbsatz als `gpl` Datei exportieren oder importieren.

### 9.3.6 Menü Karte und Legende

#### Voreingestelltes Kartenaussehen (Projekteigenschaften überschreiben)

- Definieren Sie eine *Farbe für Auswahlen* und eine *Hintergrundfarbe*

#### Layerlegende

- *Doppelklickaktion in Legende* . Sie können mit dem Doppelklick entweder 'Layereigenschaften öffnen' oder 'Attributtabelle öffnen'.
- Die folgenden *Legendenelementstile* sind möglich:
  - *Layernamen groß schreiben*
  - *Layernamen fett*
  - *Gruppennamen fett*
  - *Klassifikationsattributnamen anzeigen*
  - *Rastericons erzeugen (kann langsam sein)*
  - *Neue Layer zur gewählten oder aktuellen Gruppe hinzufügen*

### 9.3.7 Menü Kartenwerkzeuge

Dieses Menü bietet einige Optionen in Bezug auf das Verhalten des *Objekte abfragen* Werkzeugs.

- *Suchradius für die Objektidentifikation und zur Maptipanzeige* ist ein Toleranzfaktor ausgedrückt als Prozentsatz der Kartenbreite. Das heißt dass das Objekte abfragen Werkzeug so lange Ergebnisse anzeigt wie sie innerhalb dieser Toleranz klicken.
- Mit *Hervorhebungsfarbe* können Sie wählen mit welcher Farbe abgefragte Objekte hervorgehoben werden sollen.
- *Puffer*, ausgedrückt als Prozentsatz der Kartenbreite, legt eine Pufferdistanz, die durch die Umrisslinie der Hervorhebung dargestellt wird, fest.
- *Minimalbreite*, ausgedrückt als Prozentsatz der Kartenbreite, legt fest wie dick die Umrisslinie eines hervorgehobenen Objektes sein soll.

#### Messwerkzeug

- Definieren Sie *Gummibandfarbe* für das Meßwerkzeug



- Definieren Sie *Dezimalstellen*
- *Basiseinheit beibehalten*
- *Bevorzugte Maßeinheiten*  ('Meter', 'Fuß', 'Seemeilen' oder 'Grad')
- *Bevorzugtes Winkelmaß*  ('Grad', 'Bogenmaß' oder 'Gon')

#### Verschieben und Zoomen

- Definieren Sie *Mausradaktion* ('Zoom', 'Zoomen und mittig zentrieren', 'Zur Mausposition zoomen', 'Nichts')
- Definieren Sie *Zoomfaktor* für das Mausrad

#### Vordefinierte Maßstäbe

Hier finden Sie eine Liste von vordefinierten Maßstäben. Mit den [+] und [-]-Knöpfen können Sie Ihre individuellen Maßstäbe hinzufügen oder löschen.

### 9.3.8 Menü Zusammenstellung

#### Zusammenstellungsvoreinstellungen

Sie können hier die *Vorgabeschriftart* festlegen

#### Gitterdarstellung

- Definieren Sie den *Gitterstil*  ('Ausgefüllt', 'Punkte', 'Kreuze')
- Definieren Sie *Farbe ...*

#### Gittervoreinstellungen

- Legen Sie *Zwischenräume*  fest
- Legen Sie den *Gitterversatz*  für x und y fest
- Legen Sie die *Fangtoleranz*  fest

#### Führungsvorgaben

- Legen Sie die *Fangtoleranz*  fest

### 9.3.9 Menü Kartenwerkzeuge

#### Objekterzeugung

- *Eingabe der Attributwerte bei der Erstellung neuer Objekte unterdrücken*
- *Letzte Attributwerteingaben wiederverwenden*
- *Geometrien prüfen*. Das Bearbeiten komplexer Linien/Polygone mit vielen Stützpunkten kann zu einer erheblichen Verlangsamung der Darstellung führen. Das liegt an den Standard-Validierungsverfahren, die viel Zeit benötigen. Um die Darstellung zu beschleunigen ist es möglich die Geometrieüberprüfung von GEOS (von GEOS 3.3.an) zu wählen oder die Validierung ganz auszuschalten. Die GEOS Geometrieüberprüfung ist viel schneller, aber der Nachteil ist dass nur das erste Geometrieproblem gemeldet wird.

#### Gummiband

- Definieren Sie für das Gummiband die *Linienbreite* und *Linienfarbe*


#### Objektfang

- *Fangoptionsdialog in einem Dockfenster anzeigen (QGIS-Neustart erforderlich)*
- Legen Sie den *Voreingestellten Fangmodus*  ('Zum Stützpunkt', 'Zum Segment', 'Zum Stützpunkt und Segment', 'Aus') fest




- Definieren Sie die *Voreingestellte Fangtoleranz* in Karteneinheiten oder Pixeln
- Definieren Sie den *Suchradius für die Stützpunktbearbeitung* in Karteneinheiten oder Pixeln.

### Stützpunktmarken

- *Markierungen nur für gewählte Objekte anzeigen*
- Legen Sie für die Stützpunktmarken den *Markierungsstil*  ('Kreuz' (standard), 'Teiltransparenter Kreis' oder 'Keine') fest
- Definieren Sie die *Markierungsgröße* für die Stützpunktmarken

### Werkzeug zum Linien versetzen

Die nächsten 3 Optionen beziehen sich auf das  *Linie versetzen* Werkzeug in *Erweiterte Digitalisierung*. Durch die verschiedenen Einstellungen ist es möglich die Form des Linienversatzes zu beeinflussen. Diese Optionen sind von GEOS 3.3 an möglich.

- *Verbindungsstil*
- *Quadrantensegmente*
- *Eckengrenze*

## 9.3.10 Menü GDAL

GDAL ist eine Datenaustauschbibliothek für Rasterdateien. In diesem Menü können Sie *Erzeugungsoptionen bearbeiten* und *Pyramidenoptionen bearbeiten*. Definieren Sie welcher GDAL-Treiber für ein Rasterformat benutzt wird, da in einigen Fällen mehr als ein Treiber zur Verfügung gestellt wird.

### 9.3.11 Menü KBS

#### Vorgabe-KBS für neue Projekte

- *'Spontanreprojektion' nicht einschalten*
- *Spontanreprojektion (SRP) automatisch aktivieren wenn die Layer unterschiedliche KBS haben*
- *Spontanreprojektion voreinstellen*
- Wählen Sie ein KBS aus und *Neue Projekte immer in diesem KBS beginnen*

#### KBS für neue Layer

Der zweite Bereich ermöglicht es, Voreinstellungen vorzunehmen, wenn ein neuer Layer erzeugt wird oder ein Layer geladen wird, der keine KBS Information besitzt.

- *KBS abfragen*
- *KBS des Projekts benutzen*
- *Folgendes KBS benutzen*

#### Datumtransformationsvorgaben

- *Datumtransformation erfragen wenn keine Vorgabe definiert ist*
- Wenn Sie mit der 'Spontan' KBS Transformation gearbeitet haben können sie das Ergebnis der Transformation im Fenster unten sehen. Sie können Informationen über das 'Quell-KBS' und das 'Ziel-KBS' genauso wie über 'Quell-Datumstransformation' und 'Ziel-Datumstransformation' finden.

### 9.3.12 Menü Sprache

- *System-Locale überschreiben und Stattdessen folgende Spracheinstellungen benutzen*
- Informationen über die aktive System-Locale

### 9.3.13 Menü Netzwerk

#### Allgemein

- *WMS Suchadresse* - Standard ist `http://geopole.org/wms/search?search=%1&type=rss`
- Definieren Sie *Zeitüberschreitung für Netzwerkanfragen (ms)* - Standard ist 60000
- Definieren Sie *Verfallszeitraumvorgabe für WMS-C/WMTS Kacheln (Stunden)* - Standard ist 24
- Legen Sie *Maximale Wiederholungen bei Kachelabfragefehlern* fest.
- Definieren Sie *User-Agent*

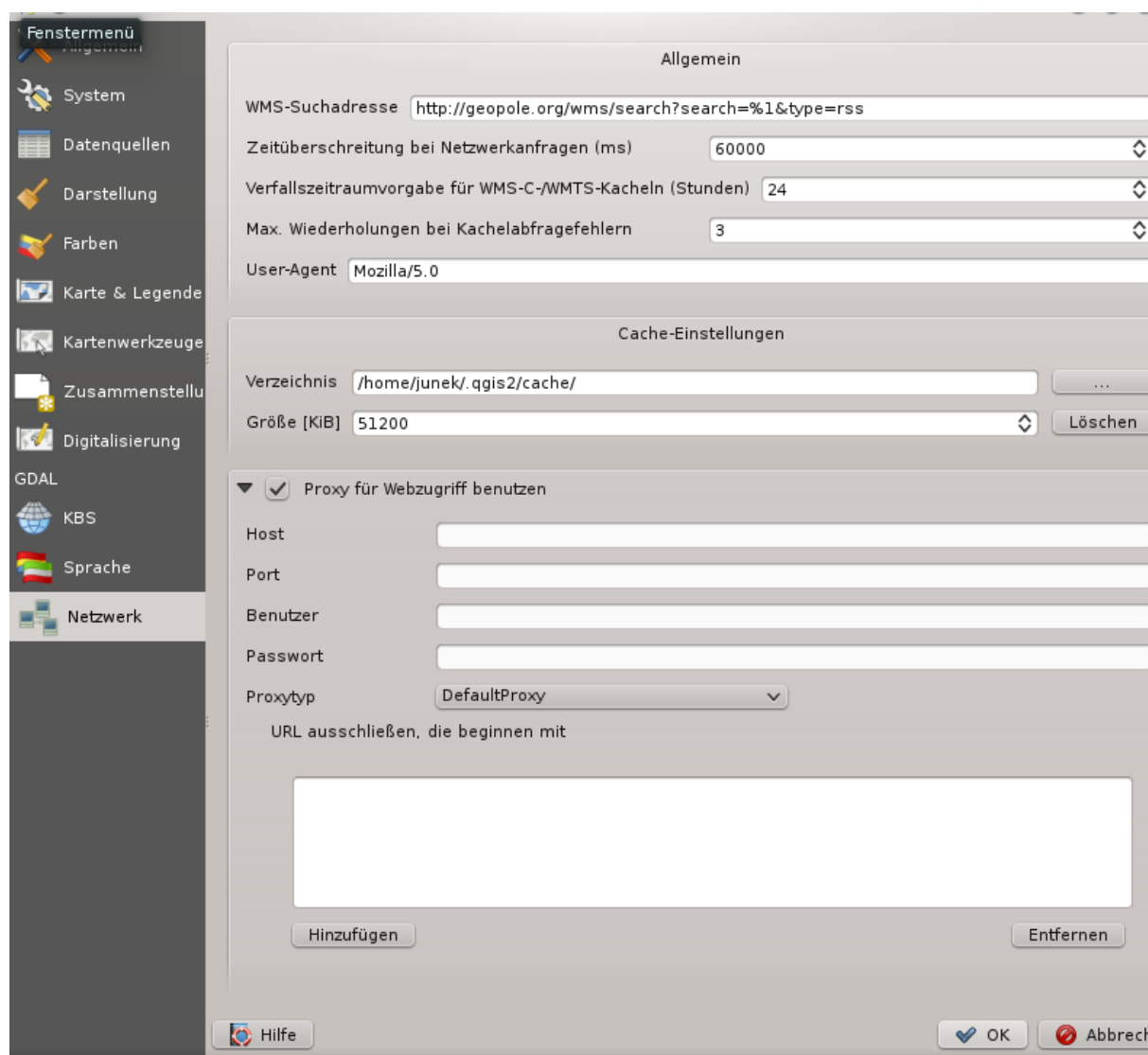



Abbildung 9.4: Proxy-Einstellungen in QGIS

#### Cache-Einstellungen

Definieren Sie das *Verzeichnis* und eine *Größe* für den Cache.

-  *Proxy für Web-Zugriff benutzen* und entsprechend ‘Host’, ‘Port’, ‘Benutzer’ und ‘Passwort’ definieren.
- Definieren Sie den *Proxytyp* → entsprechend ihrer Bedürfnisse.
  - *DefaultProxy* →: Proxy wird auf Grundlage des Proxy der Anwendung festgelegt
  - *Socks5Proxy* →: Proxy für jede Art von Verbindung. Unterstützt TCP, UDP, Bindung an einen Port (eingehende Verbindungen) und Authentifizierung.
  - *HttpProxy* →: Umgesetzt mit dem “CONNECT”-Befehl, unterstützt nur ausgehende TCP-Verbindungen und Authentifizierung.
  - *HttpCachingProxy*: Umgesetzt mit normalen HTTP Befehlen ist dies nur im Zusammenhang mit HTTP Befehlen sinnvoll einzusetzen.
  - *FtpCachingProxy*: Mit einem FTP-Proxy umgesetzt ist dies nur sinnvoll im Zusammenhang mit FTP-Anforderungen anzuwenden

Das Ausschließen von URLs kann in das Textfenster unter den Proxyeinstellungen eingetragen werden (siehe [Figure\\_Network\\_Tab](#)).

Wenn Sie weitere Informationen über die verschiedenen Proxyeinstellungen benötigen, schauen Sie bitte im Handbuch der QT-Bibliothek nach unter der <http://doc.trolltech.com/4.5/qnetworkproxy.html#ProxyType-enum>.




---

**Tipp: Proxy richtig anwenden**

Das Verwenden von Proxys kann manchmal schwierig sein. Es ist sinnvoll die oben angesprochenen Proxytypen auszuprobieren, um zu sehen, welcher in dem jeweiligen Fall am besten funktioniert.






---


Sie können die Optionen nach Ihren Bedürfnissen verändern. Bei einigen der Änderungen kann ein Neustart von QGIS erforderlich sein bevor sie wirksam werden.

-  Einstellungen sind gespeichert in einer Textdatei: `$HOME/.config/QGIS/QGIS2.conf`
-  Alle Einstellungen sind gespeichert in: `$HOME/Library/Preferences/org.qgis.qgis.plist`
-  Einstellungen sind gespeichert in der Registry unter: `HKEYCURRENT_USERSoftwareQGISqgis`

## 9.4 Anpassung

Mit dem Anpassungswerkzeug können fast jedes Element in der QGIS Benutzeroberfläche (de)aktivieren. Dies kann sehr nützlich sein wenn man viele Plugins, die sie nie benutzen und die den Bildschirm überfüllen, installiert hat.

Die QGIS-Anpassung ist in fünf Gruppen unterteilt. Mit  *Menus* können Sie Einträge aus der Menüleiste verstecken. In  *Panels* finden Sie die Dock-Fenster. Dock-Fenster sind Anwendungen die als frei schwebendes top-level Fenster oder eingebettet in das QGIS Hauptfenster a gedocktes Wided gestartet und genutzt werden können (siehe auch [Bedienfelder und Werkzeugkästen](#)). Mit  *Status Bar* können Features wie die Koordinateninformation deaktiviert werden. Mit  *Toolbars* können Sie Werkzeugleisteneinträge von QGIS und mit  *Widgeds* können Sie Dialoge genauso wie Ihre Knöpfe (de)aktivieren.

Mit  Umschalten um Bedienelemente der Hauptapplikation zu fangen können Sie auf Elemente in QGIS die sie verstecken wollen klicken und den entsprechenden Eintrag in Anpassung finden (siehe [figure\\_customization](#)). Sie können genauso ihre verschiedenen Setups für verschiedene Anwendungen speichern. Sie müssen QGIS neustarten bevor die Änderungen angewendet werden.

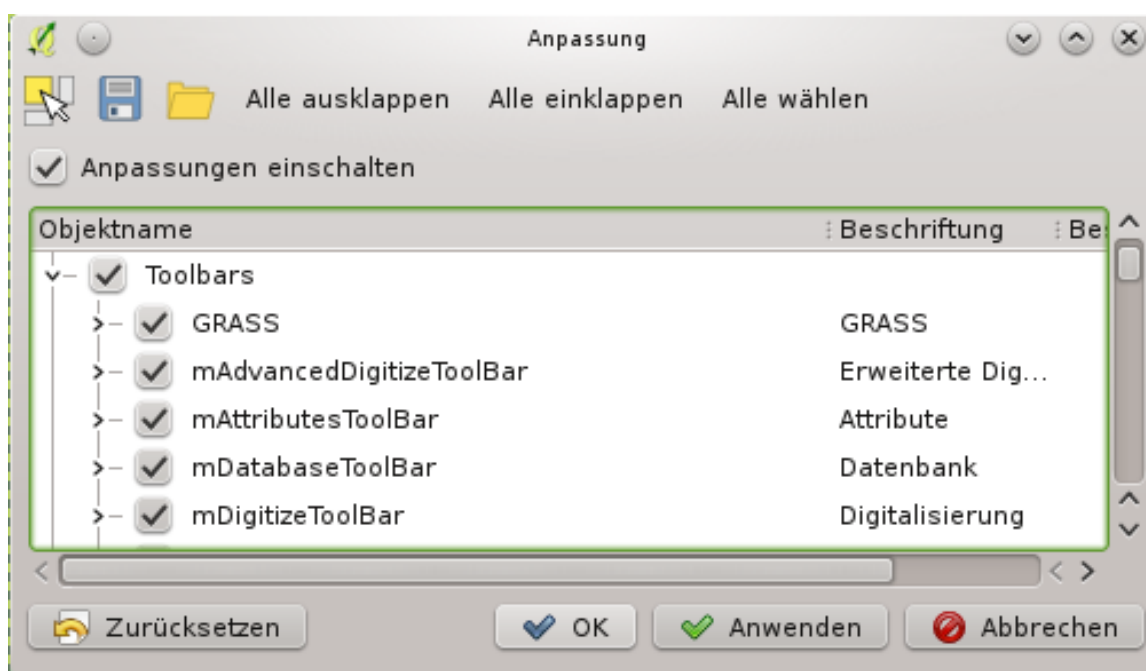



Abbildung 9.5: Der Anpassung Dialog 

---

## Arbeiten mit Projektionen

---

QGIS ermöglicht es, globale und projektbezogene KBS (Koordinatenbezugssysteme) für Layer ohne vordefinierte KBS zu definieren. Es können benutzerdefinierte Koordinatenbezugssysteme erstellt werden und für Raster- und Vektorlayer wird On-The-Fly (OTF) Projektion unterstützt, um Layer gemeinsam und lagegenau darzustellen, auch wenn sie unterschiedliche KBS besitzen.

### 10.1 Überblick zur Projektionsunterstützung

QGIS unterstützt etwa 2700 bekannte Koordinatenbezugssysteme (KBS). Diese 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. Selbstdefinierte KBS sind in einer Benutzerdatenbank abgelegt. Informationen zum Anlegen einer Benutzerdatenbank finden Sie im Abschnitt *Eigenes Koordinatenbezugssystem definieren*.


Die Koordinatenbezugssysteme in QGIS basieren auf EPSG Codes der European Petroleum Search Group (EPSG) und dem Institut Geographic National de France (IGNF) und entsprechen weitestgehend den spatial reference Tabellen der Software GDAL. Die EPSG IDs sind in einer SQLite-Datenbank abgelegt und werden benutzt, um KBS in QGIS zu spezifizieren.

Um OTF Projektion zu verwenden, müssen die Daten Informationen über ihr Koordinatenbezugssystem enthalten oder Sie müssen ein globales, layer- oder projektbezogenes KBS definieren. 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 eines KBS bei den Daten. Bei Shapes bedeutet dies, dass eine Datei mit der Endung .prj vorhanden sein muss, in der das KBS im Well Known Text (WKT) Format angegeben ist. Für ein Shape mit dem Namen `alaska.shp` gäbe es also eine entsprechende Projektionsdatei `alaska.prj`.

Sobald Sie ein neues KBS auswählen ändern sich die Layereinheiten automatisch im Menü *Allgemein* 'des `!mActionOptions!` :`guilabel: 'Projekteigenschaften` Dialogs unter *Projekt (Gnome, OS X)* oder `:guilabel: 'Einstellungen` (KDE, Windows).

### 10.2 Bestimmung einer globalen Projektion

QGIS startet ein neues Projekt mit dem Festlegen eines Standard-KBS. Vordefiniert ist EPSG:4326 - WGS84 (`proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`). Sie können diese Grundeinstellung über den **[Wählen ...]** Knopf, der dazu verwendet wird das Standard-Koordinatensystem für neue Projekte zu definieren, im ersten Abschnitt ändern, wie in `figure_projection_1` gezeigt. Diese Auswahl wird dann für die folgenden QGIS Sitzungen gespeichert.

Wenn Sie Layer, die kein KBS besitzen, verwenden, müssen Sie festlegen wie QGIS auf diese Layer reagiert. Dies kann global oder projektbezogen im *KBS* Menü unter *Einstellungen* →  *Optionen* erfolgen.

Die Optionen aus `figure_projection_1` sind:

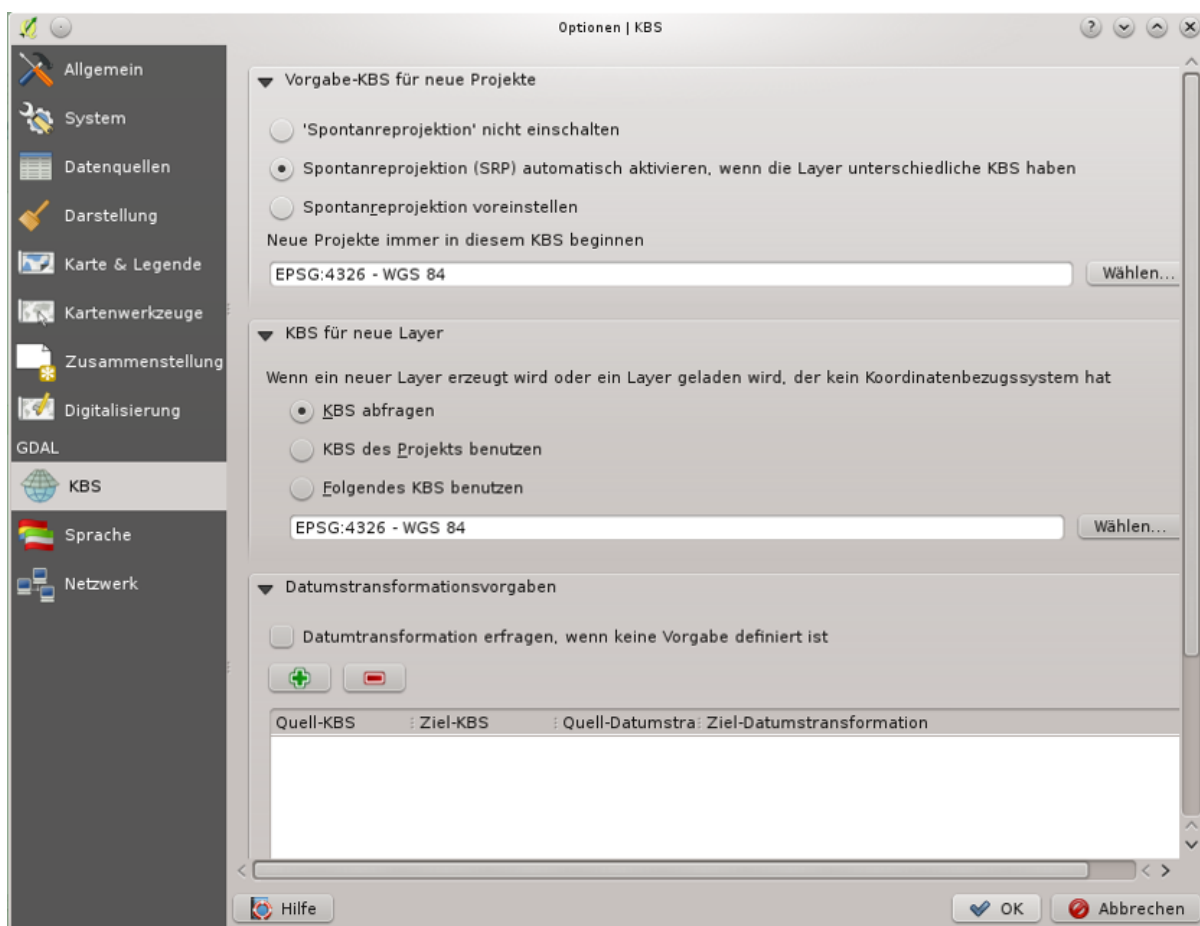



Abbildung 10.1: Menü KBS im QGIS Optionen Dialog 

- *KBS abfragen*
- *KBS des Projektes nutzen*
- *Folgendes KBS benutzen*

Wenn Sie ein Koordinatenbezugssystem für einen bestimmten Layer ohne KBS-Information definieren wollen können Sie das im Menü *Allgemein* des Raster- und Vektorlayereigenschaften-Dialogs durchführen (siehe [Menü Allgemein](#) für Rasterlayer und `:ref:vectorgeneralmenu'` für Vektorlayer). Wenn der Layer schon eine KBS-Information besitzt wird dies wie in [Vektor Layereigenschaften Dialog](#) dargestellt.


---

**Tipp: KBS im Kontextmenü des Layers**



Wenn Sie mit der rechten Maustaste auf den Layernamen im Legendenbereich klicken, öffnet sich das Kontextmenü des Layers (Kapitel [Legende](#)). Dort befinden sich zwei Möglichkeiten zur Einstellung des KBS *KBS für Layer setzen* öffnet den KBS Dialog (siehe Abbildung [figure\\_projection\\_2](#)). *Layer-KBS dem Projekt zuweisen* überschreibt das aktuelle Projekt-KBS mit dem KBS des Layers.


---

## 10.3 On-The-Fly (OTF) Projektion

QGIS unterstützt jetzt die On-The-Fly Reprojektion für Raster- und Vektorlayer. Diese Funktion ist aber nicht als Standard aktiviert. Um sie auszuwählen, öffnen Sie den Dialog  *Projekteinstellungen*, wechseln in das Menü *KBS* und klicken dort auf das Kontrollkästchen  *Spontan-KBS-Transformation aktivieren*.

Es gibt drei Möglichkeiten, um dies zu erreichen:

1. Wählen Sie  *Projekteigenschaften* aus dem *Projekt* (Gnome, OSX) oder *Einstellungen* (KDE, Windows) Menü.
2. Klicken Sie auf das  *KBS Status* Icon in der rechten unteren Ecke der Statusleiste.
3. Aktivieren Sie OTF als Standardeinstellung, indem Sie das Menü *KBS* im *Optionen* Dialog öffnen und  *'On-The-Fly'-Reprojektion voreinstellen* aktivieren wenn die Layer verschiedene KBS haben.

Wenn Sie bereits einen Layer geladen haben und nun die Unterstützung für Spontan-KBS-Projektion aktivieren wollen ist der beste Weg folgender: Öffnen Sie das Menü *KBS* im Menü *Projekteinstellungen ...*, wählen Sie das passende KBS für den Layer aus und aktivieren Sie dann das Kontrollkästchen  *'On-The-Fly'-KBS-Transformation aktivieren*. Das Icon  *KBS-Status* ist nun nicht länger ausgegraut, und alle daraufhin geladenen Layer werden On-The-Fly auf das ausgewählte KBS projiziert.

Das Menü *KBS* des *Projekteigenschaften*-Dialogs enthält 5 wichtige Optionen (siehe [Figure\\_projection\\_2](#)). Diese werden im Folgenden beschrieben.

1. **Spontan-KBS-Transformation aktivieren** – Dieses Kontrollkästchen wird dazu verwendet Spontan-KBS-Transformation ein- oder auszuschalten. Ist diese ausgeschaltet wird jeder Layer anhand der Koordinaten aus der Datenquelle gezeichnet und die unten beschriebenen Komponenten sind nicht aktiv. Ist diese eingeschaltet werden die Koordinaten in jedem Layer zu dem KBS, das für das Kartenfenster definiert wurde, projiziert.
2. **Filter** - wenn Sie den EPSG Code, die ID oder den Namen für ein Koordinatenbezugssystem kennen können Sie diese benutzen, um ihr Koordinatenbezugssystem zu finden. Geben Sie einfach einen EPSG Code, eine ID oder einen Namen ein.
3. **Kürzlich benutzte Koordinatenbezugssysteme** -Wenn Sie bestimmte Koordinatenbezugssysteme regelmäßig für ihre tägliche GIS Arbeit verwenden, werden diese für den 'schnellen' Zugriff unterhalb des Fensters mit den vorhandenen KBS angezeigt. Klicken Sie auf einen der Knöpfe, um das entsprechende KBS direkt auszuwählen.
4. **Koordinatenbezugssystem der Welt** – Dies ist eine Liste von allen KBS die von QGIS unterstützt werden, darunter Geographische, Projizierte und Benutzerdefinierte Koordinatenbezugssysteme. Um ein KBS zu

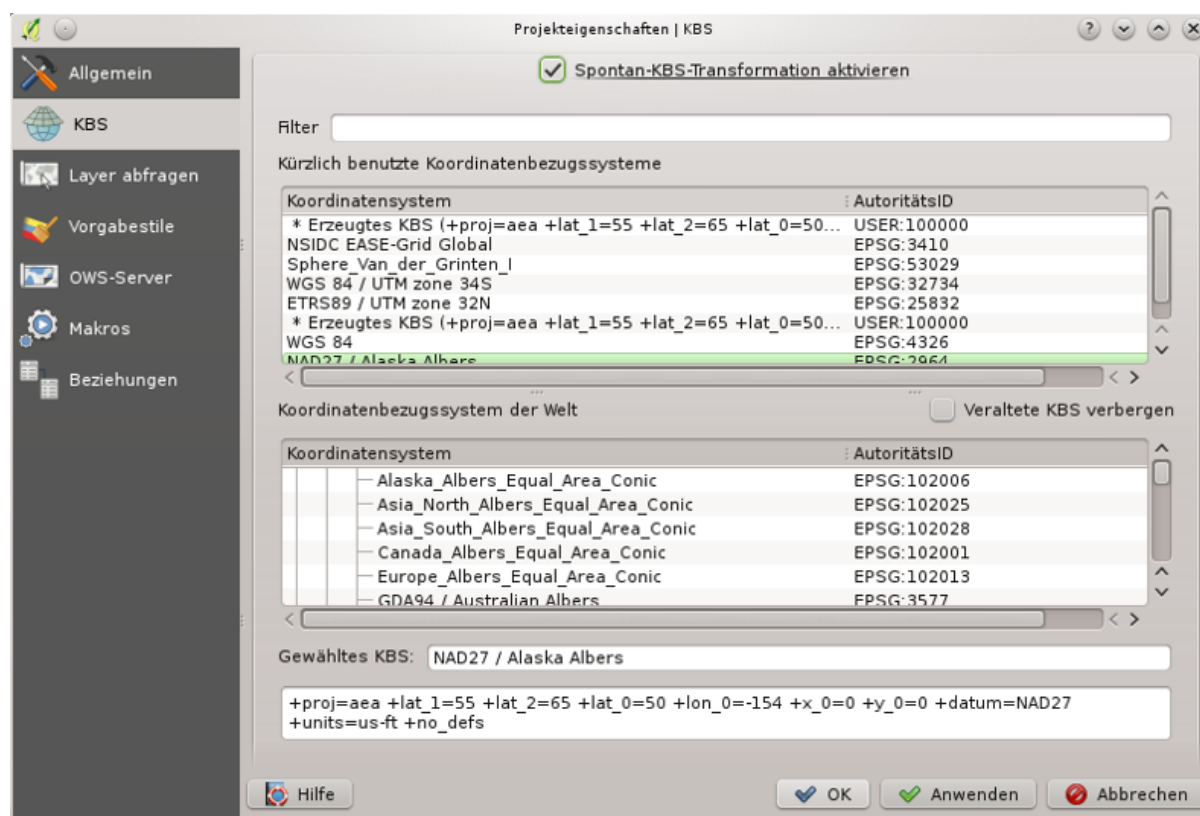



Abbildung 10.2: Projekteigenschaften Dialog 

definieren wählen Sie es aus der Liste indem Sie den entsprechenden Knoten aufklappen und das KBS auswählen. Das aktive KBS ist vorgewählt.


5. **Proj4Text** - dies ist ein Ausdruck der von der PROJ4-Bibliothek genutzt wird. Er dient nur zu Information und kann nicht verändert werden.

### Tipp: Dialog Projekteigenschaften

Wenn Sie den *Projekteigenschaften* Dialog von dem *Projekt* Menü öffnen müssen Sie auf das *CRS* Menü klicken um die KBS Einstellungen anzuzeigen.

Wenn man den Dialog anhand des  *KBS-Status* Icons öffnet wird automatisch das *KBS* Menü in den Vordergrund gebracht.

## 10.4 Eigenes Koordinatenbezugssystem definieren

Wenn QGIS nicht das Koordinatenbezugssystem das Sie brauchen zur Verfügung stellt können Sie ein Benutzerdefiniertes Koordinatensystem definieren. Um ein KBS zu definieren wählen Sie  *Benutzerkoordinatenbezugssystem ...* aus dem Menü *Einstellungen*. Benutzerkoordinatenbezugssysteme werden in Ihrer QGIS Benutzerdatenbank gespeichert. Zusätzlich zu Ihrem Benutzerkoordinatensystem enthält diese Datenbank auch Ihre Räumlichen Lesezeichen und andere Benutzerdaten.

Wenn man in QGIS eine eigene Projektion erstellen möchte bedarf es einem grundlegenden Verständnis im Umgang mit der PROJ.4-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>).






Abbildung 10.3: Der Benutzerkoordinatensystem Dialog 

Dieses Handbuch beschreibt die Anwendung von `proj.4` und die damit verbundenen Kommandozeilenprogramme. Die dort beschriebenen kartographischen Parameter sind identisch mit denen, die in QGIS verwendet werden.

Der Dialog *Definition eines Benutzerkoordinatensystems* braucht nur zwei Einträge, um eine eigene Projektion zu definieren:

1. Ein beschreibender Name
2. Die kartografischen Parameter im PROJ.4-Format


Um ein neues KBS zu erstellen klicken Sie  <sup>Neu</sup> Knopf und geben Sie einen aussagekräftigen Namen sowie die KBS-Parameter ein.

Denken Sie daran, dass die kartographischen Parameter mit einem `+proj=`-Block beginnen müssen, um den Beginn eines neuen KBS anzuzeigen.

Sie können das neue KBS testen, um zu sehen, ob bei einer Konvertierung von bekannten WGS84 Lat-Lon Koordinaten in ihre Projektion ein sinnvolles Ergebnis herauskommt. Dazu kopieren Sie ihre kartographischen Parameter in das Fenster *Parameter*, geben ein paar bekannte WGS84 Lat-Lon Koordinaten an und klicken dann auf den Knopf [**Berechnen**]. Vergleichen Sie die Ergebnisse mit den Werten im Kartenfenster.

## 10.5 Standard Datumtransformationen

Eine Spontan-Reprojektion hängt davon ab ob Daten in ein ‘Standard-KBS’ transformiert werden können, QGIS benutzt hierbei WGS84. Für einige KBS sind eine Reihe von Transformationen verfügbar. Sie können unter QGIS die benutzte Transformation definieren sonst benutzt es eine Standard-Transformation.

Im *KBS Menü* unter *Einstellungen* →  *Optionen* können Sie:

- `qgl` so einstellen dass es fragt wenn es eine Transformation definieren muss indem Sie  *Datumtransformation erfragen, wenn keine Vorgabe definiert ist* benutzen
- eine Liste von Benutzereinstellungen für Transformationen bearbeiten

QGIS fragt welche Transformation benutzt werden soll indem es eine Dialogbox, die den PROJ.4 Text der wiederum die Quell- und Ziel-Datumstransformation beschreibt, öffnet. Weitere Informationen sind zu finden indem man mit der Maus über eine Transformation geht. Benutzereinstellungen können gespeichert werden indem Sie  *Speichere Auswahl* auswählen.

## QGIS Browser

Der QGIS Browser ist ein Bedienfeld in QGIS mit dem Sie auf einfache Art und Weise in Ihrem Dateisystem navigieren können und Geodaten verwalten können. Sie können Zugang zu Ihren Vektordateien (z.B. ESRI Shape-dateien oder MapInfo Dateien), zu Ihren Datenbanken (z.B. PostGIS, Oracle, SpatiaLite oder räumliche MSSQL) und zu Ihren WMS/WFS Verbindungen herstellen. Sie können auch GRASS-Daten anschauen (siehe *GRASS GIS Integration* um die Daten ins QGIS zu importieren).

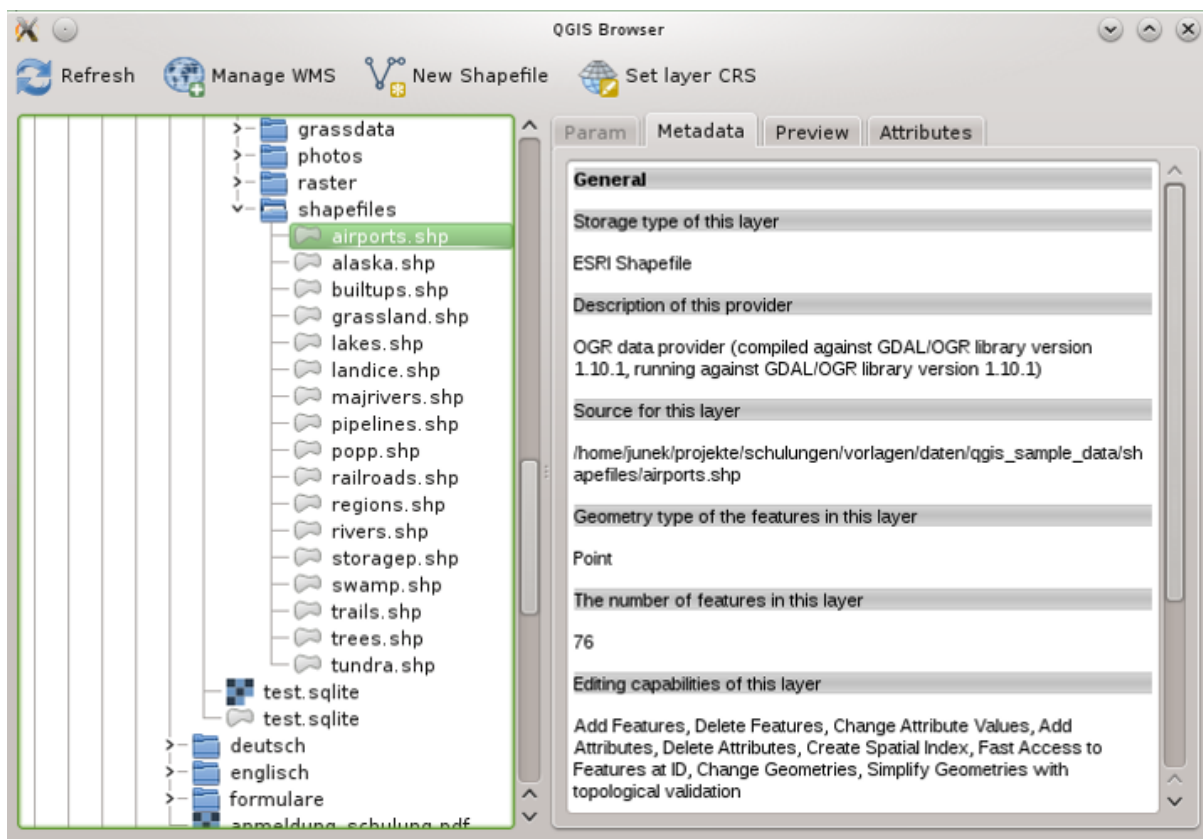







Abbildung 11.1: QGIS Browser als eigenständige Anwendung 

Verwenden Sie QGIS Browser um einen Preview von Ihren Daten zu machen. Mit der Drag- und Drop-Funktion können Sie auf einfache Art und Weise ihre Daten in das Kartenfenster und in die Legende bringen.


1. Aktivieren Sie den QGIS Browser: Machen Sie einen Rechtsklick in die Werkzeugleiste und aktivieren Sie  *Browser* oder wählen Sie es in *Einstellungen* → *Bedienfelder* aus.
2. Verschieben Sie das Bedienfeld in die Legende und lass Sie es los.
3. Klicken Sie auf den *Browser* Reiter

4. Durchsuchen Sie Ihr Dateisystem und wählen Sie die den `shapefile` Ordner aus dem `qgis_sample_data` Verzeichnis aus.
5. Drücken Sie den Umschalt Taste und wählen Sie die `airports.shp` und `alaska.shp` Dateien.
6. Klicken Sie die linke Maustaste und machen dann mit den Dateien ein Drag&Drop in die Kartenansicht.
7. Machen Sie einen Rechtsklick auf einen Layer und wählen Sie *Layer-KBS dem Projekt zuweisen*. Weitere Informationen sehen Sie unter *Arbeiten mit Projektionen*.
8. Klicken Sie auf  `Volle Ausdehnung` um die Layer sichtbar zu machen.

Es gibt einen zweiten Browser unter *Einstellungen* → *Bedienfelder*. Dies ist nützlich wenn Sie Dateien oder Layer zwischen verschiedenen Orten verschieben müssen.



1. Aktivieren Sie den QGIS Browser: Machen Sie einen Rechtsklick in die Werkzeugleiste und aktivieren Sie  *Browser (2)* oder wählen Sie es in *Einstellungen* → *Bedienfelder* aus.
2. Schieben Sie das Bedienfeld in die Legende.
3. Navigieren Sie zum *Browser (2)* Reiter und suchen Sie nach einer Shapedatei in Ihrem Dateisystem.
4. Wählen Sie eine Datei mit der linken Maustaste aus. Jetzt können Sie das  *Gewählte Layer hinzufügen* Icon verwenden um diese dem aktuellen Projekt hinzuzufügen.

QGIS sucht automatisch nach dem Koordinatenbezugssystem (KBS) und zoomt auf die Layerausdehnung wenn Sie in einem leeren QGIS Projekt arbeiten. Wenn es schon Dateien in Ihrem Projekt gibt wird die Datei einfach hinzugefügt und in dem Fall dass es die gleiche Ausdehnung und KBS hat wird sie dargestellt. Wenn die Datei ein anderes KBS und Layerausdehnung hat müssen Sie erst auf den Layer rechtsklicken und dann *Layer-KBS dem Projekt zuweisen* auswählen. Wählen Sie dann *Auf die Layerausdehnung zoomen*.

Die  *Dateifilter* Funktion funktioniert auf Verzeichnisebene. Suchen Sie den Ordner in dem Sie Dateien filtern wollen und geben Sie ein Suchwort oder Wildcard ein. Der Browser zeigt dann nur die zutreffenden Dateinamen an - andere Daten werden nicht dargestellt.

Es ist genauso möglich, den QGIS Browser als eigenständige Anwendung auszuführen.

### Den QGIS Browser starten

-  Tippen Sie `qbrowser` in eine Kommandozeile.
-  Starten Sie den QGIS Browser indem Sie das Startmenü oder die Desktop-Verknüpfung verwenden.
- **X** Sie können den QGIS Browser über dem Anwendungen Ordner erreichen.

In [figure\\_browser\\_standalone\\_metadata](#) können Sie die erweiterte Funktionalität des eigenständigen QGIS Browsers sehen. Der *Param* Reiter stellt die Details Ihrer verbindungs-basierten Datensätze dar, so PostGIS und räumliche MSSQL. Der *Metadata* Reiter enthält allgemeine Informationen über die Datei (siehe *Menü Metadaten*). Mit dem *Preview* Reiter können Sie einen Blick auf Ihre Dateien machen ohne Sie in Ihr QGIS Projekt zu importieren. Es ist sogar möglich eine Vorschau von den Attributen Ihrer Dateien im Reiter *Attributes* zu machen.

---

## Arbeiten mit Vektordaten

---

### 12.1 Unterstützte Datenformate

QGIS verwendet die OGR-bibliothek um Vektordatenformate zu lesen und zu schreiben, einschließlich ESRI Shapedateien, MapInfo und MicroStation Dateiformate, AutoCAD DXF, PostGIS, SpatiaLite, Oracle Spatial und MSSQL Spatial Datenbanken und viele mehr. GRASS Vektor und PostgreSQL Support wird durch native Datenprovider Plugins bereitgestellt. Die Vektordaten können auch im Lesemodus aus zip- und gzip-Archiven ins QGIS geladen werden. Zum Zeitpunkt der Erstellung dieses Dokumentes werden 69 Vektorformate von der OGR-Bibliothek unterstützt (siehe OGR-SOFTWARE-SUITE in *Literatur und Internetreferenzen*). Die vollständige Liste ist auf [http://www.gdal.org/ogr/ogr\\_formats.html](http://www.gdal.org/ogr/ogr_formats.html) zu finden.

---

**Bemerkung:** Einige der aufgelisteten Formate können auf ihrem Rechner aus unterschiedlichen Gründen nicht unterstützt werden. Einige brauchen z.B. kommerzielle Bibliotheken oder die GDAL Installation auf Ihrem Rechner wurde ohne die Unterstützung für das entsprechende Format erstellt. Nur Formate, die getestet wurden, können ausgewählt werden, wenn Sie eine Vektordatei in QGIS laden. Alle anderen werden angezeigt, wenn Sie \*.\* auswählen.

---

Das Arbeiten mit GRASS GIS Vektorlayern wird in Kapitel *GRASS GIS Integration* beschrieben.

In diesem Abschnitt wird beispielhaft beschrieben, wie man mit ESRI Shapedateien, PostGIS- und SpatiaLite-Layern, OpenStreetMap Vektordateien und Comma Separated data (CSV) 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.

#### 12.1.1 ESRI Shapes


Die ESRI Shapedatei ist das Standard Vektorformat in QGIS und wird durch die OGR Simple Feature Library (<http://www.gdal.org/ogr/>) bereitgestellt.

Ein Shape besteht derzeit aus mehreren Dateien. Die folgenden drei sind erforderlich:

1. `.shp` Datei (enthält die Geometrien)
2. `.dbf` Datei (enthält die Attribute im dBase-Format)
3. `.shx` Indexdatei

Darüber hinaus kann eine Datei mit `.prj` Endung existieren. Diese enthält die Projektionsinformationen des Shapes. Während es sehr nützlich ist eine Projektionsdatei zu verwenden ist dies nicht zwingend erforderlich. Ein Shape-Datensatz kann zusätzliche Dateien enthalten. Details dazu finden sich in der technischen Spezifikation von ESRI unter <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

## Shape Layer laden

Um eine Shapefile zu laden starten Sie QGIS und klicken Sie auf den  Vektorlayer hinzufügen Knopf in der Werkzeugleiste oder drücken einfach `Strg+Umschalt+V`. Dieses öffnet ein neues Fenster (siehe [figure\\_vector\\_1](#)).

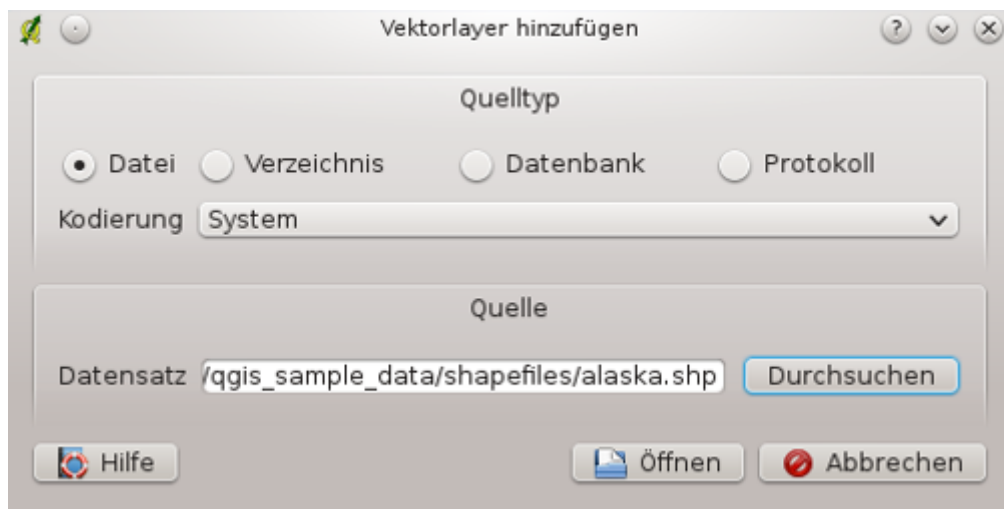




Abbildung 12.1: Vektorlayer hinzufügen Dialog 

Wählen Sie aus den möglichen Quelltypen  *Datei* und klicken Sie auf den Knopf [**Durchsuchen**]. Dadurch erscheint ein weiterer Dialog zum Öffnen (siehe [figure\\_vector\\_2](#)) mit dem Sie im Dateisystem navigieren können und Sie ein Shape oder eine andere unterstützte Datenquelle laden können. Die Auswahlbox *Filter*  ermöglicht es Ihnen einige OGR-unterstützte Dateiformate vorzuzwählen.

Außerdem kann auch der Kodierungstyp für die Shapefile eingestellt werden, falls dies notwendig ist.

Durch Auswahl eines Shapes und Anklicken des Knopfes [**Öffnen**] wird die Datei in QGIS geladen. In [Abbildung Figure\\_vector\\_3](#) sehen Sie das Ergebnis, nachdem die Beispieldatei `alaska.shp` ins QGIS geladen wurde.

---

### **Tipp: Farben von Vektorlayern**

Wenn Sie einen neuen Vektorlayer in QGIS laden, werden Farben zufällig zugewiesen. Wenn Sie mehrere neue Vektorlayer laden, werden jeweils unterschiedliche Farben zugewiesen.

Nach dem Laden können Sie mit den Navigationstools aus der Werkzeugleiste beliebig zoomen. Um den Stil eines Layers zu verändern öffnen Sie den *Layereigenschaften* Dialog in dem Sie auf den Layernamen doppelklicken oder indem Sie einen Rechtsklick auf den Namen in der Legende machen und *Eigenschaften* im Popumenu wählen. Vergleichen Sie Abschnitt [vector\\_style\\_tab](#) für weitere Informationen zum Editieren der Eigenschaften von Vektorlayern.

---

### **Tipp: Layer und Projekte von externen Datenträgern laden unter OS X**

Unter OS X werden externe Datenträger unter *Datei* → *Öffne Projekt* nicht gemeinsam mit den internen Festplatten angezeigt. Dies soll zukünftig behoben werden. Solange können Sie `/Volumes` in das Eingabefenster `'Dateiname'` eintragen und `Return` drücken. Danach können Sie auch zu den externen Datenträgern bzw. Rechnern in einem Netzwerk browsen.

---

## Die Darstellungsgeschwindigkeit von Shapefiles verbessern

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`.

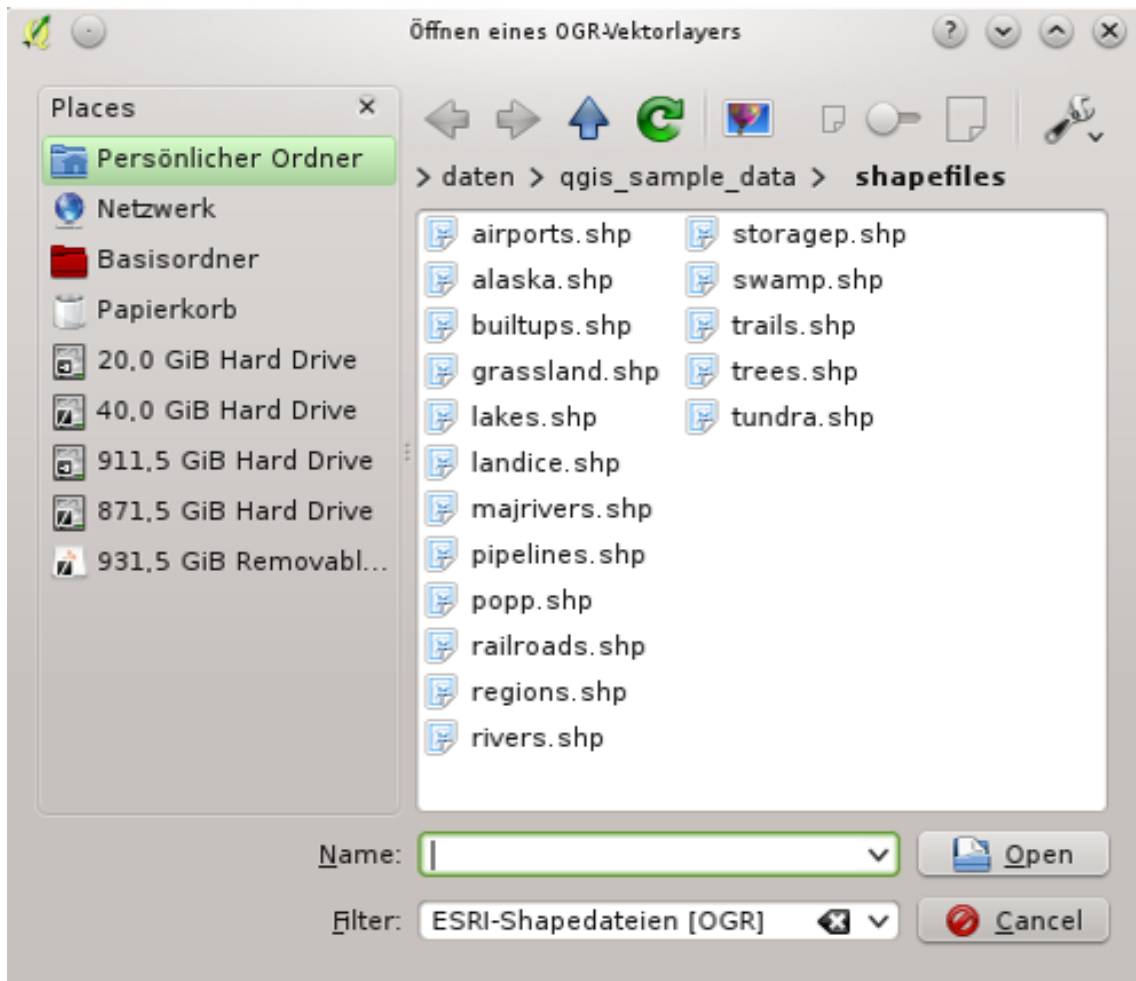


Abbildung 12.2: Öffnen eines OGR-Vektorlayers Dialog 

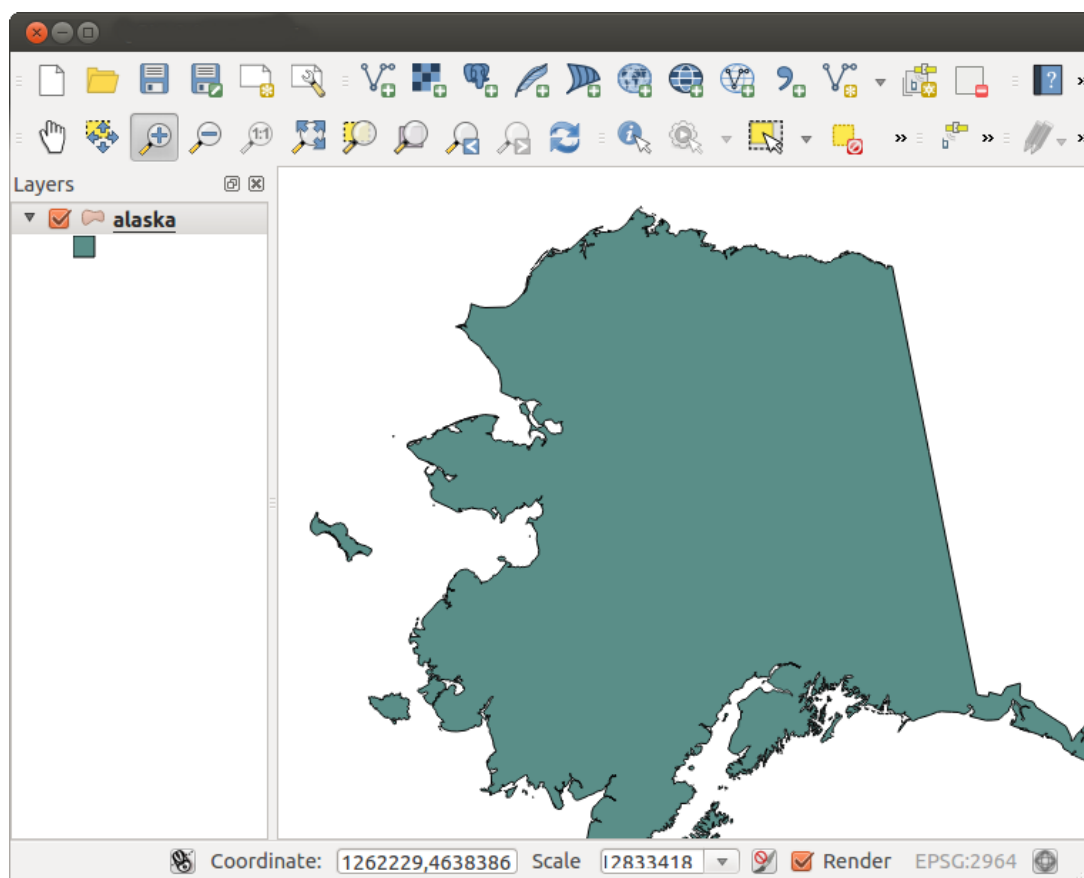



Abbildung 12.3: QGIS mit geladenem Shapefile aus dem Alaskabeispieldatensatz 🐧



Benutzen Sie folgende Schritte zum Erstellen eines räumlichen Index:



- Um eine Shapedatei zu laden klicken Sie auf den  Vektorlayer hinzufügen Knopf in der Werkzeugleiste oder drücken einfach `Strg+Umschalt+V`.
- Öffnen Sie den *Eigenschaften*-Dialog des Vektorlayers, indem Sie auf den Namen des Layers in der Legende doppelklicken oder mit der rechten Maustaste *Eigenschaften* auswählen.
- Im Menü *Allgemein* klicken Sie auf den **[Räumlichen Index erzeugen]** Knopf.

### Problem beim Laden eines Shapes mit .prj Datei





Wenn Sie eine Shapedatei mit `.prj`-Datei laden und QGIS ist nicht in der Lage, die Projektionsinformationen korrekt auszulesen, ist es notwendig das Koordinatenbezugssystem (KBS) manuell im *Allgemein* Menü des *Layereigenschaften* Dialog anhand des **[Festlegen ...]** Knopfs anzugeben. Hintergrund ist, dass `.prj` Dateien oftmals nicht die vollständigen Projektionsparameter enthalten, so wie QGIS sie benötigt und auch im *KBS* Dialog anzeigt.

Aus diesem Grund, wenn Sie ein neues Shapefile mit QGIS erstellen, werden derzeit zwei unterschiedliche Projektionsdateien angelegt. Eine `.prj` Datei, mit den unvollständigen Projektionsparametern, wie sie z.B. von ESRI Software gelesen und erstellt wird, und eine `.qpj` Datei, in der die vollständigen Projektionsparameter anhalten sind. Wenn Sie dann ein Shape in QGIS laden, und QGIS findet eine `.qpj` Datei, dann wird diese anstelle der `.prj` Datei benutzt.

### 12.1.2 MapInfo Layer laden

 Um eine MapInfo Datei zu laden, klicken Sie auf das Icon  Vektorlayer hinzufügen in der Werkzeugleiste oder drücken Sie die Taste `Strg+Umschalt+V`. Definieren Sie unter *Filter: Mapinfo File [OGR] (\*.mif \*.tab \*.MIF \*.TAB)* und wählen die gewünschte Datei aus.

### 12.1.3 ArcInfo Binary Coverage laden

 Um ein ArcInfo Coverage zu laden, klicken Sie auf das Icon  Vektorlayer hinzufügen in der Werkzeugleiste oder drücken Sie die Taste `Strg+Shift+V`. Wählen Sie aus den möglichen Quelltypen  *Verzeichnis*, wählen Sie *Arc/Info Binär-Coverage* . Wählen Sie nun den Ordner mit den ArcInfo Binary-Coverage-Dateien aus.

Auf die gleiche Art und Weise können auch andere Verzeichnis-basierte Vektorlayer geladen werden, wie etwa das U.K. National Transfer Format oder das raw TIGER Format des U.S. Census Bureau.

### 12.1.4 Delimited Text Dateien

Tabellendaten sind ein sehr beliebtes und weit verbreitetes Format wegen seiner Einfachheit und Lesbarkeit - die Daten können sogar in einem Texteditor eingesehen und bearbeitet werden. Eine Delimited Text Datei ist eine Attributtabelle bei der jede Spalte durch ein definiertes Zeichen und jede Zeile durch einen Zeilenumbruch getrennt wird. Die erste Zeile enthält normalerweise die Spaltennamen. Ein gängiger Typ von Delimited Text Datei ist eine CSV-Datei (Comma Separated Values), bei der jede Spalte durch ein Komma getrennt wird.

Solche Datensätze können auch Positionsinformationen in zwei Formaten enthalten:

- Als Punktkoordinaten in getrennten Spalten
- Als WKT-Daten die die Geometrien darstellen

QGIS macht es möglich eine Delimited Text Datei als Layer oder einfache Tabelle zu laden. Überprüfen Sie aber erst dass die Datei die folgenden Anforderungen erfüllt:

1. Die Datei muss eine begrenzte Kopfspalte mit Feldnamen besitzen. Dieses muss die erste Zeile in der Textdatei sein.

2. Die Kopfspalte muss Feld(er) mit Geometriedefinitionen enthalten. Diese Feld(er) können jeden beliebigen Namen haben.
3. Die X und Y Koordinaten (wenn die Geometrien anhand von Koordinaten definiert sind) müssen als Nummern angegeben werden. Das Koordinatensystem ist nicht wichtig.


Als Beispiel für eine Textdatei importieren wir die Datei `elevp.csv` aus dem QGIS Beispieldatensatz (siehe Kapitel *Beispieldaten*):

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

Einige weitere Anmerkungen zu Textdateien:

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

### Eine Delimited Text Datei laden

Klicken Sie danach auf das Icon  Textdatei als Layer importieren in der *Layer koordinieren* Werkzeugleiste, um den Dialog *Textdatei als Layer importieren* zu öffnen, wie in *figure\_delimited\_text\_1* zu sehen.

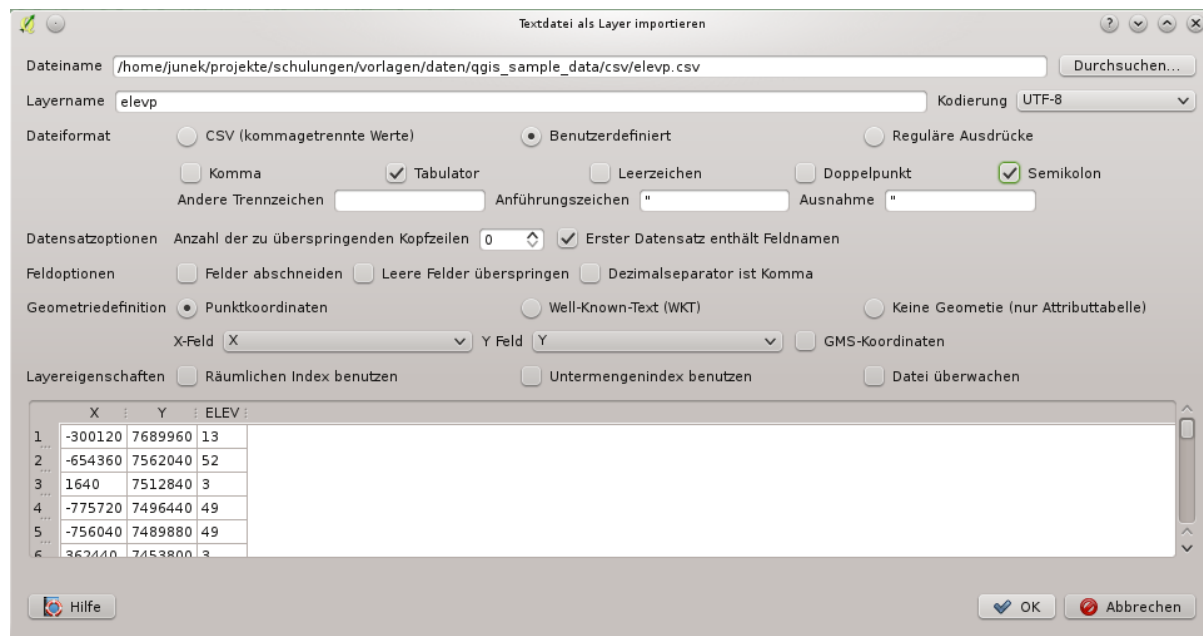


Abbildung 12.4: Textdatei als Layer importieren Dialog 

Als erstes wählen Sie eine Datei (z.B., `qgis_sample_data/csv/elevp.csv`) für den Import aus, indem Sie auf den **[Durchsuchen]** Knopf drücken. Sobald eine Datei ausgewählt wurde, versucht QGIS die Datei zu durchsuchen und auf Basis des zuletzt verwendeten Trennzeichens zu strukturieren. Damit QGIS die Datei richtig

durchsucht ist es wichtig das richtige Trennzeichen auszuwählen. Sie können ein Trennzeichen indem Sie  *Benutzerdefiniert* aktivieren angeben oder  *Reguläre Ausdrücke* aktivieren und Text in das *Ausdruck* Feld eingeben. Um z.B. einen Tabulator zu verwenden, geben Sie `\t` ein (dies ist ein regulärer Ausdruck für das Tab-Zeichen).

Ist die Datei erst einmal durchsucht können Sie die *Geometriedefinition* auf  *Punktkoordinaten* stellen und die X und Y Felder aus der Drop-Down-Liste auswählen. Sind die Koordinaten als Grad/Minuten/Sekunden definiert aktivieren Sie das Kontrollkästchen  *GMS-Koordinaten*

Als letztes geben Sie einen Layernamen ein (z.B., `elenvp`), wie in [figure\\_delimited\\_text\\_1](#) gezeigt. Um den Layer der Karte hinzuzufügen klicken Sie [OK]. Die Delimited Text Datei verhält sich jetzt wie jeder andere Kartenlayer in QGIS.

Es gibt auch eine Hilfeoption mit der Sie vorangehende und nachfolgende Leerzeichen von Feldern abschneiden können -  *Felder abschneiden*. Es ist ebenfalls möglich  *Leere Felder überspringen* anzuwenden. Falls nötig können Sie festlegen dass ein Komma der Dezimalseparator sein soll indem Sie  *Dezimalseparator ist Komma* aktivieren.

Wenn räumliche Informationen durch WKT repräsentiert sind aktivieren Sie  *Well Known Text (WKT)* und wählen Sie das Feld mit den WKT-Definitionen für Punkt-, Linien- oder Polygonobjekte aus. Wenn die Datei nicht-räumliche Daten enthält, aktivieren Sie  *Keine Geometrie (nur Attributtabelle)* und sie wird als ordinale Tabelle geladen.

Zusätzlich können Sie aktivieren:



- Räumlichen index benutzen* um die Performanz der Darstellung und räumlichen Selektion der Objekte zu verbessern.
- Untermengenindex benutzen*
- Datei überwachen* um Änderungen der Datei durch andere Anwendungen zu überwachen während QGIS läuft.

## 12.1.5 OpenStreet Daten

In den letzten Jahren hat das OpenStreetMap-Projekt an Popularität gewonnen, weil in vielen Ländern keine freien Geodaten, wie etwa digitale Straßenkarten zur Verfügung stehen. Ziel des OSM-Projekts ist es, eine frei editierbare Karte der Welt auf Basis von GPS-Daten, Luftaufnahmen oder einfach nur Ortskenntnis zu erschaffen. Um diese Idee zu unterstützen, wurde ein QGIS Plugin geschrieben, das die Arbeit mit OSM-Daten ermöglicht.

### OpenStreetMap Vektordateien laden




QGIS integriert OpenStreetMap-Import als Kernfunktionalität.

- Um sich mit dem OSM Server zu verbinden und Daten herunterzuladen öffnen Sie das Menü *Vektor* → *OpenStreetMap* → *Daten herunterladen*. Sie können diesen Schritt überspringen wenn Sie bereits eine `.osm` XML Datei über JOSM, Overpass API oder durch eine andere Quelle erhalten haben.
- Das Menü *Vektor* → *OpenStreetMap* → *Topologie aus XML importieren* konvertiert Ihre `.osm` Datei in eine SpatiaLite Datenbank und erstellt eine entsprechende Datenbankverbindung.
- Mit dem Menü *Vektor* → *OpenStreetMap* → *Topologie nach SpatiaLite exportieren* ermöglicht es Ihnen die Datenbankverbindung zu öffnen, den Datentyp Ihrer Wahl (Punkte, Linien oder Flächen) auszuwählen und Tags zum Importieren auszuwählen. Dies erstellt einen SpatiaLite-Geometrielayer den Sie Ihrem Projekt hinzufügen können indem Sie auf den  *SpatiaLite-Layer hinzufügen ...* Knopf in der Werkzeugleiste klicken oder die  *SpatiaLite-Layer hinzufügen ...* Option aus dem Menü *Layer* auswählen (siehe Abschnitt [SpatiaLite Layer laden](#)).

## 12.1.6 PostGIS Layer 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 Layern, die durch die OGR-Bibliothek geladen wurden.

### Erstellen einer PostGIS Anbindung

 Das erste mal wenn Sie eine PostGIS-Datenquelle verwenden müssen Sie eine Verbindung zur PostgreSQL-Datenbank die die Daten enthält erstellen. Beginnen Sie mit dem Klicken auf  PostGIS-Layer hinzufügen ... aus dem Layer Menü oder geben Sie `Strg+Umschalt+D` ein. Sie können genauso den Dialog *Vektorlayer hinzufügen* öffnen und  Datenbank auswählen. Um den Verbindungsmanager zu erreichen klicken Sie auf den **[Neu]** Knopf um den *Eine neue OGR-Datenbankverbindung* Dialog aufzurufen. Die für eine Verbindung erforderlichen Parameter sind:


- **Name:** Ein Name für die Verbindung. Kann derselbe wie für die **Datenbank** sein
- **Dienst:** Dienstparameter der alternativ zu Hostname/port verwendet werden kann (und eventuell Datenbank). Dies kann in `pg_service.conf` definiert werden.
- **Host:** Name des Datenbank-Hosts. Dies muss ein auflösbarer Name für den HOST sein, genau wie beim Benutzen von `telnet` oder `ping`. Wenn die Datenbank auf demselben Computer ist wie QGIS, tragen Sie hier einfach `'localhost'` ein.
- **Port:** Port Nummer der Datenbank auf dem Server. Standard ist 5432.
- **Datenbank:** Name der Datenbank.
- **SSL-Modus:** Wie die SSL-Verbindung mit dem Server ausgehandelt wird. Beachten Sie dass die PostGIS-Layerdarstellung erheblich verschnellert werden kann wenn man SSL im Verbindungseditor deaktiviert. Die folgenden Optionen sind möglich:
  - Abschalten: Nur versuchen eine unverschlüsselte SSL-Verbindung herzustellen.
  - Erlauben: Eine nicht-SSL-Verbindung versuchen. Wenn dies misslingt, eine SSL-Verbindung versuchen.
  - Bevorzugen: Eine SSL-Verbindung versuchen. Wenn dies misslingt eine nicht-SSL-Verbindung versuchen.
  - Verlangen: Versuche nur eine SSL-Verbindung.
- **Name:** Benutzername, um sich bei der Datenbank anzumelden.
- **Passwort:** Passwort das zusammen mit *Benutzername* verwendet wird um sich bei der Datenbank anzumelden.

Bei Bedarf können Sie die folgenden Kontrollkästchen aktivieren:


- *Benutzernamen speichern*
- *Passwort speichern*
- *Nur in geometry\_columns nachschauen*
- *Geometrietyp von Spalten ohne Einschränkung nicht feststellen (GEOMETRY)*
- *Nur im 'public' Schema nachschauen*
- *Auch geometrieloze Tabellen anzeigen*
- *Geschätzte Tabellenmetadaten nutzen*

Wenn alle Parameter eingetragen sind, kann die Verbindung getestet werden, indem Sie auf den Knopf **[Verbindung testen]** drücken.

## Laden eines Layers aus der PostGIS Datenbank

 Sobald Sie eine oder mehrere Verbindungen definiert haben können Sie Layer aus der PostgreSQL-Datenbank laden. Natürlich erfordert dieses dass man Daten in PostgreSQL vorhält. Lesen Sie in Kapitel *Layer nach PostgreSQL/PostGIS importieren* über eine Diskussion die das Importieren von Daten in die Datenbank beschreibt.

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

- Wenn der *PostGIS-Layer hinzufügen* Dialog nicht schon geöffnet ist öffnet das Klicken von  *PostGIS-Layer hinzufügen ...* oder das Eingeben von `Strg+Umschalt+D` den Dialog.
- Wählen Sie eine Verbindung aus dem Drop-Down Menü und klicken auf **[Verbinden]**.
- Wählen Sie  *Auch geometrieloze Tabelle anzeigen* an oder ab.
- Nutzen Sie gegebenenfalls die  *Suchoptionen* um zu definieren welche Objekte aus dem Layer geladen werden sollen oder benutzen Sie den **[Erstelle Abfrage]** Knopf um den *Abfrageerstellung* Dialog zu starten.
- Suchen Sie einen Layer den Sie laden möchten.
- Wählen Sie es aus indem Sie darauf klicken. Sie können mehrere Layer gleichzeitig indem Sie die `Umschalt`-Taste gedrückt halten auswählen. Siehe Abschnitt *Abfrageeditor* für Informationen wie man die PostgreSQL-Abfrageerstellung für das weitere Definieren des Layers benutzt.
- Klicken Sie auf den Knopf **[Hinzufügen]** um den Layer zu laden.

---

### Tipp: PostGIS-Layer

Normalerweise ist ein PostGIS-Layer über einen Eintrag in der `geometry_columns` Tabelle definiert. Seit Version 0.9.0 ist QGIS in der Lage, Layer 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.

---

## Einige Details zu PostGIS-Layern

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 Layern 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 den Layer laden zu können.

Unter QGIS müssen PostgreSQL-Layer eine Spalte beinhalten die als eindeutigen Schlüssel für den Layer benutzt werden kann. Für Tabellen heißt dies normalerweise dass die Tabelle einen Primärschlüssel oder eine Spalte mit einer eindeutigen Beschränkung braucht. In QGIS muss diese Spalte vom Typ `int4` (ein Integer mit 4 Byte Größe) sein. Alternativ kann die `ctid`-Spalte als Primärschlüssel verwendet werden. Wenn eine Tabelle keines dieser Elemente aufweist wird die `oid`-Spalte anstelle dessen verwendet. Die Performanz wird verbessert wenn die Spalte indexiert ist (beachten Sie das Primärschlüssel in PostgreSQL automatisch indexiert werden).


Wenn der PostgreSQL-Layer ein View ist bestehende dieselben Anforderungen, nur dass Views keine Primärschlüssel oder Spalten mit eindeutigen Einschränkungen besitzen. Sie müssen ein Primärschlüsselfeld (dies muss vom Typ Integer sein) im QGIS-Dialog definieren bevor Sie den View laden können. Wenn im View keine geeignete Spalte existiert lädt QGIS den Layer nicht. Tritt dies auf ist die Lösung den View zu verändern, so dass dieser dann eine geeignete Spalte beinhaltet (eine Spalte vom Typ Integer und entweder mit Primärschlüssel oder eindeutiger Beschränkung, vorzugsweise indexiert).

QGIS bietet ein Kontrollkästchen **Abfrage nach Id** das standardmäßig aktiviert ist. Diese Option wählt die ids ohne die Attribute aus was in den meisten Fällen schneller ist. Es kann sinnvoll sein diese Option zurückzusetzen wenn Sie umfangreiche Views verwenden.

### 12.1.7 Layer nach PostgreSQL/PostGIS importieren

Daten können anhand von mehreren Werkzeugen in PostgreSQL/PostGIS importiert werden, einschließlich des SPIT Plugins und den Kommandozeilenwerkzeugen shp2pgsql und ogr2ogr.

#### DB-Manager

QGIS bietet ein Kernplugin das  DB Manager heisst. Es kann dafür verwendet werden um Shapedateien und andere Datenformate zu laden und beinhaltet Unterstützung für Schemas. Siehe Kapitel *DB Manager Plugin* für weitere Informationen.

#### shp2pgsql

PostGIS beinhaltet ein Programm mit dem Namen **shp2pgsql** das dazu benutzt werden kann Shpdateien in eine PostGIS-fähige Datenbank zu importieren. Um z.B. eine Shpedatei `lakes.shp` in eine PostgreSQL-Datenbank mit dem Namen `gis_data` zu importieren verwenden Sie den folgenden Befehl:

```
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 *Arbeiten mit Projektionen*.

---

#### Tipp: Layer aus PostGIS exportieren

Wie das Importprogramm **shp2pgsql** gibt es ebenfalls ein Werkzeug um PostGIS Datensätze als Shapedateien zu exportieren: **pgsql2shp**. Dies ist innerhalb der PostGIS-Installation enthalten.

---


#### ogr2ogr

Neben **shp2pgsql** und dem **DB Manager** gibt es noch ein anderes Programm um Geodaten PostGIS zuzuführen: **ogr2ogr**. Dies ist Teil Ihrer 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
```

Dies wird die Shpedatei `alaska.shp` in die PostGIS-Datenbank `postgis` mit dem Benutzer `postgres` und dem Passwort `*topsecret` auf dem Host-Server `myhost.de` importieren.

Beachten Sie dass OGR mit PostgreSQL gebaut werden muss um PostGIS-Unterstützung zu erhalten. Sie können dies überprüfen indem Sie eingeben (in )

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

Wenn Sie es vorziehen dem PostgreSQL **COPY**-Befehl anstelle der Standard **INSERT INTO** Methode zu arbeiten können Sie die folgende Umgebungsvariable (zumindest unter  und **X** zugänglich) benutzen

```
export PG_USE_COPY=YES
```

**ogr2ogr** erstellt keine räumlichen Indizes wie **shp2pgsl** das tut. Sie müssen diese hinterher als zusätzlichen Schritt manuell unter Zuhilfenahme des normalen SQL-Befehls **CREATE INDEX** erstellen (wie im nächsten Abschnitt *Geschwindigkeit optimieren* beschrieben).



## 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 PostGIS räumlicher Index erstellt wird. PostGIS unterstützt das Erstellen eines GiST (Generalized Search Tree) Index, um den Zugriff auf die Ebenen zu beschleunigen (die GiST Index Informationen wurden aus der PostGIS Dokumentation übernommen unter: <http://postgis.refractions.net>).

Dies ist die Syntax für das Erstellen eines GiST-Index:

```
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 *Literatur und Internetreferenzen*).

Im Folgenden sehen Sie ein Beispiel, um einen GiST-Index zu erstellen:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, 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$
```

### 12.1.8 Vektorlayer, die den Längengrad 180° überschreiten

Viele GIS Applikationen stellen einen Vektorlayer, der über den Längengrad 180 hinausgeht nicht zusammenhängend dar ([http://postgis.refractions.net/documentation/manual-2.0/ST\\_Shift\\_Longitude.html](http://postgis.refractions.net/documentation/manual-2.0/ST_Shift_Longitude.html)). So wird in QGIS der Layer geteilt und man sieht im Kartenfenster zwei, weit voneinander entfernte Teile, die eigentlich zusammengehören. In Abbildung [Figure\\_vector\\_4](#) sollte z.B. der kleine Punkt in der linken Ecke des Kartenfensters (Chatham Inseln) rechts neben Neuseeland angezeigt werden.



Abbildung 12.5: Karte in lat/lon die den 180° Längengrad schneidet 

Eine Möglichkeit, dies zu umgehen, bietet PostGIS und die Funktion **ST\_Umschalt\_Longitude** ([http://postgis.refractions.net/documentation/manual-1.4/ST\\_Umschalt\\_Longitude.html](http://postgis.refractions.net/documentation/manual-1.4/ST_Umschalt_Longitude.html)). Die Funktion liest alle Objekte der Karte ein und wenn der Längengrad <0° ist, werden 360° hinzugezählt. Das Ergebnis ist eine 0-360° Karte, die als Mittelpunkt den Längengrad 180° verwendet.

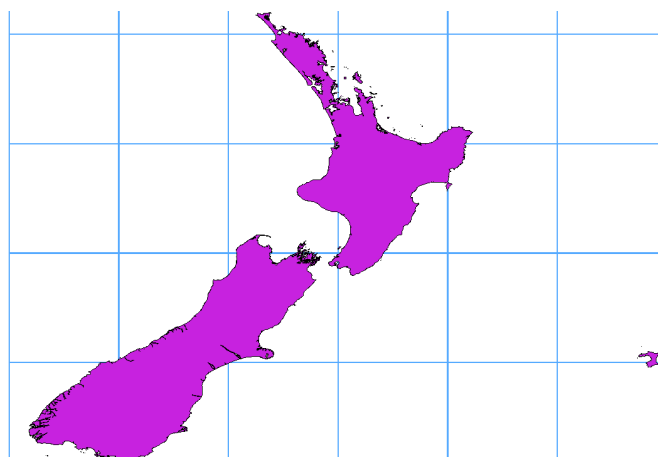





Abbildung 12.6: Das Überqueren von 180° Länge unter Verwendung der `ST_Shift_Longitude` Funktion

## Beispielanwendung

- Importieren Sie Daten in PostGIS (*Layer nach PostgreSQL/PostGIS importieren*) in dem Sie z.B. das DB Manager Plugin benutzen.
- Öffnen Sie das PostGIS Kommandozeilenfenster und geben Sie folgendes Kommando ein (in diesem Beispiel steht der Name "TABELLE" für den tatsächlichen Namen der PostGIS Tabelle): `gis_data=# update TABELLE set the_geom=ST_Shift_Longitude(the_geom);`
- Wenn alles gut gelaufen ist sollten Sie jetzt eine Bestätigung über die Anzahl der Objekte die geupdated worden sind erhalten. Dann sollten Sie die Karte laden und den Unterschied sehen können (*Figure\_vector\_5*).

### 12.1.9 SpatiaLite Layer laden

 Wenn Sie das erste Mal einen Layer aus einer SpatiaLite Datenbank laden möchten, klicken Sie auf das Icon  SpatiaLite-Layer hinzufügen in der Werkzeugleiste oder indem Sie  *SpatiaLite Layer hinzufügen ...* im Menü *Layer* auswählen oder indem Sie die Taste `Strg+Shift+L` drücken. Dies öffnet einen Dialog, über den Sie entweder eine Verbindung zu einer bereits in QGIS definierten SpatiaLite Datenbank auswählen oder eine neue Verbindung erstellen können. Um eine neue Verbindung zu erstellen, klicken Sie auf den Knopf **[Neu]** und verwenden dann den Dateibrowser, um eine entsprechende SpatiaLite Datenbank auszuwählen. Dabei handelt es sich um eine Datei mit der Endung `.sqlite`.

Wenn Sie einen Vektorlayer im SpatiaLite-Format abspeichern wollen, wählen Sie den Layer in der Legende aus, benutzen Sie das Kontextmenü der rechten Maustaste und klicken Sie auf *Speichern als ....* Geben Sie den Namen der Ausgabe an, wählen Sie `sqlite` als Format aus und das `KBS`. Danach fügen Sie noch `SPATIALITE=YES` in das Fenster 'Datenquelle'. Damit sagen Sie OGR, dass eine SpatiaLite-Datenbank erstellt werden soll. Siehe [http://www.gdal.org/ogr/drv\\_sqlite.html](http://www.gdal.org/ogr/drv_sqlite.html).

QGIS unterstützt auch das Bearbeiten von Views in SpatiaLite-Datenbanken.

### Einen neuen SpatiaLite Layer erzeugen

Wenn Sie einen neuen SpatiaLite Layer erzeugen wollen, finden Sie in Kapitel *Einen neuen SpatiaLite Layer erstellen* eine Anleitung.

---




#### **Tipp: SpatiaLite Datenmanagement Plugin**

Für das Managen von SpatiaLite-Daten können Sie mehrere Python Plugins verwenden: QSpatialite oder DB Manager (Kernplugin, wird empfohlen). Sie können mit dem Plugin Installer gedownloaded und installiert werden.

---






## 12.1.10 Räumliche MSSQL-Layer

 QGIS verfügt auch über native MS SQL 2008 Unterstützung. Wenn Sie das erste mal Räumliche MSSQL Daten laden fangen Sie damit an indem Sie auf den  Add MSSQL Spatial Layer Knopf in der Werkzeugleiste klicken oder indem Sie die  Räumlichen MSSQL-Layer hinzufügen... Option aus dem Layer Menü wählen, oder indem Sie `Strg+Umschalt+M` eingeben.

## 12.1.11 Oracle Spatial Layer

Die räumlichen Objekte in Oracle Spatial helfen Nutzern Geographische und Location Daten als native Typen innerhalb einer Oracledatenbank zu verwalten. QGIS bietet jetzt Unterstützung für solche Ebenen.

### Erstellen einer PostGIS Anbindung


 Wenn Sie das erste mal eine Oracle Spatial Datenquelle verwenden müssen Sie eine Verbindung zu der Datenbank die die Daten enthält erstellen. Beginnen Sie indem Sie auf den  Oracle-GeoRaster-Layer hinzufügen Knopf in der Werkzeugleiste drücken, die  Oracle-GeoRaster-Layer hinzufügen Option aus dem Layer Menü wählen oder indem Sie `Strg+Umschalt+O` eingeben. Um den Verbindungsmanager zu erreichen klicken Sie auf den [Neu] Knopf um den *Oracle-Verbindung herstellen* Dialog aufzurufen. Die für die Verbindung erforderlichen Parameter sind:

- **Name:** Ein Name für diese Verbindung. Er kann der gleiche wie *Datenbankinstanz* sein
- **Datenbank:** SID oder SERVICE\_NAME der Oracleinstanz.
- **Host:** Name des Datenbank-Hosts. Dies muss ein auflösbarer Name für den HOST sein, genau wie beim Benutzen von telnet oder ping. Wenn die Datenbank auf demselben Computer ist wie QGIS, tragen Sie hier einfach 'localhost' ein.
- **Port:** Port Nummer der Oracle Spatial Datenbank auf dem Server. Standard ist 1521.
- **Benutzername:** Benutzername der verwendet wird um sich in die Datenbank einzuloggen.
- **Passwort:** Passwort das zusammen mit *Benutzername* verwendet wird um sich bei der Datenbank anzumelden.

Wahlweise können Sie die folgenden Kontrollkästchen aktivieren:

- *Benutzernamen speichern* zeigt an ob der Datenbankbenutzername in der Verbindungskonfiguration gespeichert werden soll.
- *Passwort speichern* zeigt an ob das Datenbankpasswort in den Verbindungseinstellungen gespeichert werden soll.
- *Nur in Metadatentabelle geometry\_columns nachsehen.* Schränkt die dargestellten Tabellen auf jene die sich im all\_sdo\_geom\_metadata View befinden ein. Dies kann die Initialanzeige von räumlichen Tabellen verschnellern.
- *Nur nach Tabellen des Benutzers suchen.* Beim Suchen nach räumlichen Tabellen die Suche auf Tabellen die dem Benutzer gehören einschränken.
- *Auch geometrieloze Tabelle anzeigen* zeigt dass Tabellen ohne Geometrie auch standardmäßig aufgeführt werden.
- *Verwende geschätzte Tabellenstatistiken für die Layermetadaten.* Wenn der Layer aufgesetzt wird werden verschiedene Metadaten für die Oracletabelle benötigt. Dies beinhaltet Informationen wie die Tabellenspaltenanzahl, Geometrietyp und räumliche Ausdehnung der Daten in der Geometriespalte. Wenn die Tabelle eine große Anzahl von Spalten enthält kann das Festlegen dieser Daten zeitaufwändig sein. Indem

Sie diese Optionen aktivieren werden die folgenden schnellen Tabellenmetadatenoptionen durchgeführt: die Spaltenanzahl wird durch `all_tables.num_rows` festgelegt. Die Tabellenausdehnung wird immer mit den `SDO_TUNE.EXTENTS_OF` Funktionen festgelegt, auch wenn ein Layerfilter angewendet wird. Die Tabellengeometrie wird von den ersten 100 nicht-null Geometriespalten der Tabelle festgelegt.



-  *Nur bestehende Geometrietypen.* Führt nur die bestehenden Geometrietypen auf fragt nicht nach anderen.

Wenn alle Parameter eingetragen sind, kann die Verbindung getestet werden, indem Sie auf den Knopf **[Verbindung testen]** drücken.


---

### **Tipp: QGIS Benutzereinstellungen und Sicherheit**




Je nach Arbeitsumgebung birgt das Speichern von Passwörtern in Ihren QGIS-Einstellungen Risiken in sich. Ihre Einstellungen für QGIS werden basierend auf ihrem Betriebssystem unverschlüsselt in der Systemkonfiguration und den Projektdateien gespeichert! Ihre benutzerdefinierten Einstellungen für QGIS werden abhängig vom Betriebssystem gespeichert.

-  Die Einstellungen werden in Ihrem Homeverzeichnis in `~/ .qgis2` gespeichert.
  -  Die Einstellungen werden in der Registry gespeichert.
- 

### **Einen Oracle Spatial Layer laden**

 Haben Sie erst eine oder mehrere Verbindungen definiert können Sie Layer von der Oracle-Datenbank laden. Natürlich erfordert dies dass Sie schon Daten in Oracle vorhalten.

Um einen Layer von Oracle Spatial zu laden, führen Sie die folgenden Schritte durch:

- Wenn der *Oracle-Spatial-GeoRaster wählen* Dialog noch nicht offen ist, klicken Sie auf den  Oracle-GeoRaster-Layer Werkzeugknopf.
  - Wählen Sie eine Verbindung aus dem Drop-Down Menü und klicken auf **[Verbinden]**.
  - Wählen Sie  *Auch geometrieloze Tabelle anzeigen* an oder ab.
  - Verwenden Sie wahlweise  *Suchoptionen* um zu definieren welche Objekte aus dem Layer geladen werden sollen oder verwenden Sie den **[Abfrage erstellen]** Knopf um den Dialog *Abfrageerstellung* zu starten.
  - Suchen Sie einen Layer den Sie laden möchten.
  - Wählen Sie ihn aus indem Sie darauf klicken. Sie können mehrere Layer auf einmal wählen indem Sie die Umschalt Taste beim Klicken gedrückt halten. Siehe Kapitel *Abfrageeditor* über Informationen wie man den Oracle Query Builder benutzt um den Layer weiter zu definieren.
  - Klicken Sie auf den Knopf **[Hinzufügen]** um den Layer zu laden.
- 

### **Tipp: Oracle Spatial Layer**

Normalerweise wird ein Oracle Spatial Layer durch einen Eintrag in der `USER_SDO_METADATA` Tabelle definiert.

---


## 12.2 Die Symbolbibliothek

### 12.2.1 Darstellung

Die Symbolbibliothek ist der Ort wo Anwender allgemeine Symbole, die in mehreren QGIS Projekten verwendet werden können, erstellen können. Sie ermöglicht Anwendern Symbole zu exportieren und zu importieren, Symbole zu gruppieren und hinzuzufügen, Symbole zu bearbeiten und zu entfernen. Sie können sie unter *Einstellungen* → *Style Manager* oder vom **Stil** Reiter der Vektorlayereigenschaften aus öffnen.


#### Symbole teilen und importieren

Anwender können Symbole in zwei gängige Formate exportieren und importieren: qml (QGIS Format) und SLD (OGC Standard). Beachten Sie dass das SLD Format von QGIS nicht vollständig unterstützt wird.


 Freigeben gibt eine Drop-down-Liste wieder mit der der Anwender Symbole importieren oder exportieren kann.

#### Gruppen und Schlaue Gruppen

Gruppen sind Kategorien von Symbolen und Schlaue Gruppen sind dynamische Gruppen.






Um eine Gruppe zu erstellen machen Sie einen Rechtsklick auf eine bestehende Gruppe oder auf das **Gruppen** Hauptfenster links in der Bibliothek. Sie können ebenfalls eine Gruppe auswählen und auf den  Element hinzufügen Knopf klicken.

Um ein Symbol einer Gruppe hinzuzufügen können Sie entweder einen Rechtsklick auf ein Symbol machen und dann *Gruppierung anwenden* und den vorher hinzugefügten Gruppennamen auswählen. Es gibt einen zweiten

Weg mehrere Symbole einer Gruppe hinzuzufügen: wählen Sie einfach eine Gruppe aus und klicken Sie  und wählen Sie **Symbole gruppieren**. Für alle Symbole wird ein Kontrollkästchen angezeigt, das es Ihnen ermöglicht ein Symbol den ausgewählten Gruppen hinzuzufügen. Wenn Sie fertig sind können Sie auf den gleichen Knopf klicken und **Gruppierung beenden** auswählen.

Das Erstellen von **Schlaue Symbolen** ist dem Erstellen einer Gruppe ähnlich, Sie müssen stattdessen nur **\*Schlaue Gruppen** auswählen. Das Dialogfenster ermöglicht es dem Anwender den Ausdruck zum Auswählen von Symbolen auswählen damit sie in der Schlaue Gruppe erscheinen (hat die Markierung, ist ein Mitglied der Gruppe, hat einen Teil von Namensübereinstimmung, etc.)

#### Symbole hinzufügen, bearbeiten und entfernen

Mit dem *Stilmanager* aus dem **[Symbol]**  Menü können Sie Ihre Symbole verwalten. Sie können  Element hinzufügen,  Element bearbeiten,  Element löschen und  Element freigeben. 'Markierung' Symbole, 'Linie' Symbole, 'Füllung' Muster und 'Farbverläufe' können verwendet werden um die Symbole zu erstellen. Die Symbole werden dann zu 'Alle Symbole', 'Gruppen' oder 'Schlaue Gruppen' zugewiesen.

Für jede Art von Symbol finden Sie immer die gleiche Dialogstruktur:

- in der linken oberen Seite eine Symboldarstellung
- unter der Symboldarstellung einen Symbolbaum der die Symbollayer zeigt
- rechts können Sie einige Parameter einstellen (Einheit, Transparenz, Farbe, Größe und Drehung)
- unter diesen Parametern finden Sie Symbole aus der Symbolbibliothek

Der Symbolbaum ermöglicht das Hinzufügen, Löschen oder Sperren eines einfachen neuen Symbols. Sie können den Symbollayer nach oben oder unten verschieben.

Detailliertere Einstellungen können vorgenommen werden indem man auf den zweiten Level im *Symbollayer* Dialog klickt. Sie können *Symbollayer* definieren, die danach kombiniert werden. Ein Symbol kann aus mehreren *Symbollayer* bestehen. Die Einstellungen werden später in diesem Kapitel gezeigt.

---

**Tipp:** Beachten Sie dass wenn Sie einmal die Größe in den unteren Levels des *Symbollayer* Dialogs eingestellt haben, die Größe des ganzen Symbols mit dem *Größe* Menü im ersten Level wieder geändert werden kann. Die Größe der unteren Levels ändern sich entsprechend während das Größenverhältnis beibehalten wird.

---

### 12.2.2 Markierungssymbole

Markierungssymbole besitzen mehrere Symbollayertypen:

- Ellipsenmarkierung
- Schriftmarkierung
- Einfache Markierung (voreingestellt)
- SVG-Markierung
- Vektorfeldmarkierung

Die folgenden Einstellungen sind möglich:

- *Symbollayertyp*:: Sie habe die Option Ellipsenmarkierungen, Schriftmarkierungen, Einfache Markierungen, SVG-Markierungen und Vektorfeldmarkierungen zu verwenden.
- *Farben*
- *Größe*
- *Umrandungsstil*
- *Umrandungsstärke*
- *Drehung*
- *X-, Y-Versatz*: Sie können das Symbol in X- oder Y-Richtung verschieben.
- *Ankerpunkt*
- *Datendefinierte Eigenschaften ...*

### 12.2.3 Liniensymbole

Linienmarkierungssymbole haben nur zwei Symbollayertypen:

- Markierungslinie
- Einfache Linie (voreingestellt)

Der voreingestellte Symbollayertyp zeichnet eine einfache Linie wohingegen der andere einen regelmäßigen Markierungspunkt auf der Linie darstellt. Sie können verschiedene Markierungsplatzierungen wie Stützpunkt, Intervall oder Mittelpunkt wählen. Die Markierungslinie kann einen Versatz entlang der Linie oder einen Linienversatz haben. Schließlich ermöglicht Ihnen *Markierung rotieren* die Orientierung des Symbols zu verändern.

Die folgenden Einstellungen sind möglich:

- *Farbe*
- *Stiftbreite*
- *Versatz*
- *Stiftstil*
- *Verbindungsstil*

- *Endstil*
- *Benutzerdefinierte Strichlierung verwenden*
- *Strichlierungsmustereinheit*
- *Datendefinierte Eigenschaften ...*

## 12.2.4 Füllungssymbole

Füllungsmarkierungssymbole haben ebenfalls mehrere Symbollayertypen:

- Zentrierte Füllung
- Gradientenfüllung
- Linienmusterfüllung
- Punktmusterfüllung
- SVG-Füllung
- Shapeburstfüllung
- Einfache Füllung (voreingestellt)
- Rand: Markierungslinie (das Gleiche wie Linienmarkierung)
- Rand: Einfach Linie (das Gleiche wie Linienmarkierung)

Die folgenden Einstellungen sind möglich:

- *Farben* für den Rahmen und die Füllung
- *Füllstil*
- *Umrandungsstil*
- *Umrandungsstärke*
- *X-, Y-Versatz*
- *Datendefinierte Eigenschaften ...*

Wenn Sie die Farb-Combobox verwenden, können Sie ein Drag-und-Drop von einem Farbknopf zu einem anderen Farbknopf machen, eine Farbe kopieren und einfügen, eine Farbe von irgendwo aufnehmen oder eine Farbe von der Palette oder Recent oder Standard auswählen. Die Combobox erlaubt es, das Objekt mit Transparenz zu füllen. Sie können auch einfach auf den Knopf klicken um den Palette-Dialog zu öffnen. Beachten Sie dass Sie eine Farbe von einer externen Software wie GIMP importieren können.

Der 'Gradientenfüllung' *Symbollayertyp* ermöglicht Ihnen zwischen einer  *Zweifarbige* und einer  *Farbverlauf* Einstellung zu wählen. Sie können den  *Objektzentroiden* als *Referenzpunkt* verwenden. Alle 'Gradientenfüllung' *Symbollayertyp* Füllungen sind auch über das *Symbol* Menü der kategorisierten und abgestuften Darstellung und über das *Regeleigenschaften* Menü der regelbasierten Darstellung erreichbar. Eine andere Möglichkeit ist eine 'Shapeburstfüllung' zu wählen, die eine gepufferte Gradientenfüllung darstellen, wobei ein Gradient von der Grenze eines Polygons in Richtung der Polygonmitte gezeichnet wird. Die konfigurierbaren Parameter beinhalten den Abstand von der Grenze zum Schatten, die Verwendung eines Farbverlaufs oder eines einfachen Zweifarbengradienten, Verwischstärke und Versätze.

Es ist möglich nur Polygongrenzen in das Polygon zu zeichnen. Verwenden Sie dazu 'Rand: Einfache Linie' mit gewähltem Kontrollkästchen  *Linie nur im Polygon zeichnen*.

## 12.2.5 Farbverlauf

Sie können einen benutzerdefinierten Farbverlauf indem Sie *Neuer Farbverlauf ...* aus dem *Farbverlauf* Dropdown-Menü auswählen. Ein Dialog wird Sie auffordern einen Verlaufstyp zu wählen: Gradiente, Zufällig, Farbrauer oder cpt-city. Die ersten drei haben Optionen für die Anzahl von Schritten oder Mehrfachstops im Farbverlauf. Sie können die  *Invertieren* Option beim Klassifizieren der Daten mit einem Farbverlauf verwenden. Siehe [figure\\_symbology\\_3](#) als Beispiel eines benutzerdefinierten Farbverlaufs und [figure\\_symbology\\_3a](#) für den cpt-city Dialog.

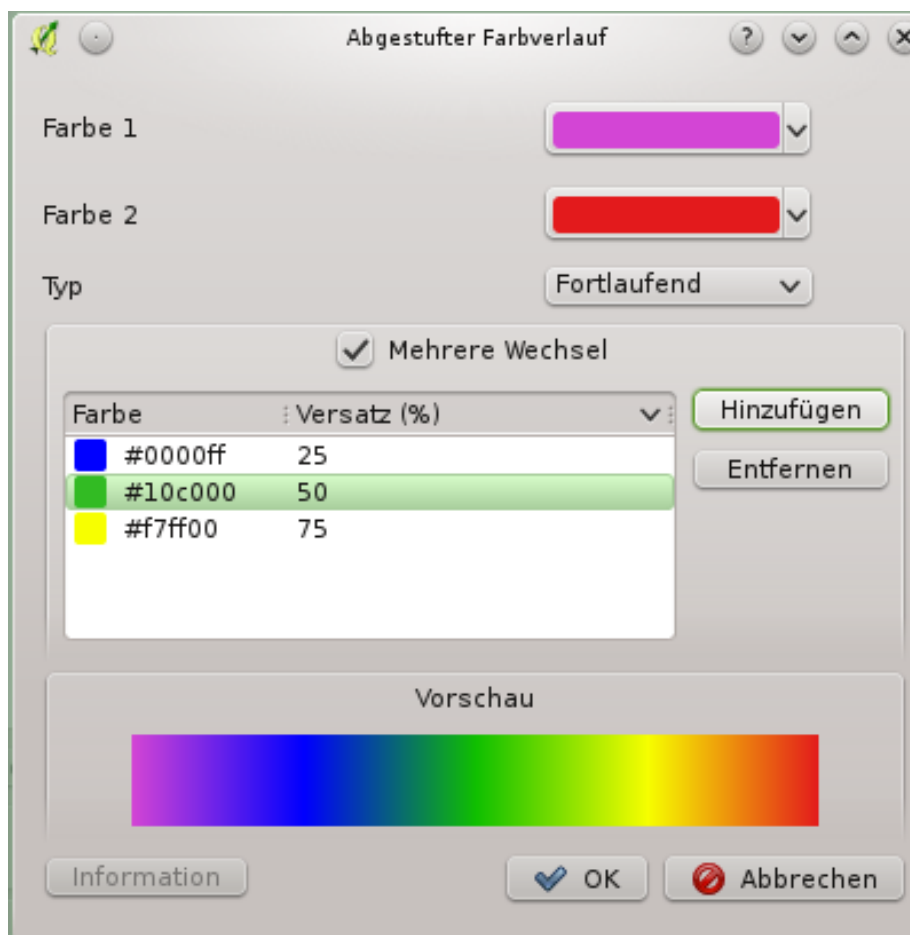



Abbildung 12.7: Beispiel eines benutzerdefinierten Farbverlaufs mit Mehrfachstops 

Die cpt-city Option öffnet einen neuen Dialog mit hunderten von sofort einsatzfähigen Themen.

Ausdrücke für das Beschriften verwenden 

## 12.3 Vektorlayereigenschaften

Der *Layereigenschaften*-Dialog stellt Informationen über den Layer, Darstellungseinstellungen und Beschriftungsoptionen bereit. Wenn ein Vektorlayer aus einer PostgreSQL/PostGIS Datenbank geladen wurde, können über den Dialog *Layereigenschaften* auch SQL-Abfragen mit dem *Objektuntermenge*-Dialog im Menü *Allgemein* angewendet werden. Um den *Layereigenschaften*-Dialog zu erreichen doppelklicken Sie einen Layer in der Legende oder machen Sie einen Rechtsklick auf den Layer und wählen Sie *Eigenschaften* aus dem Pop-upmenü.

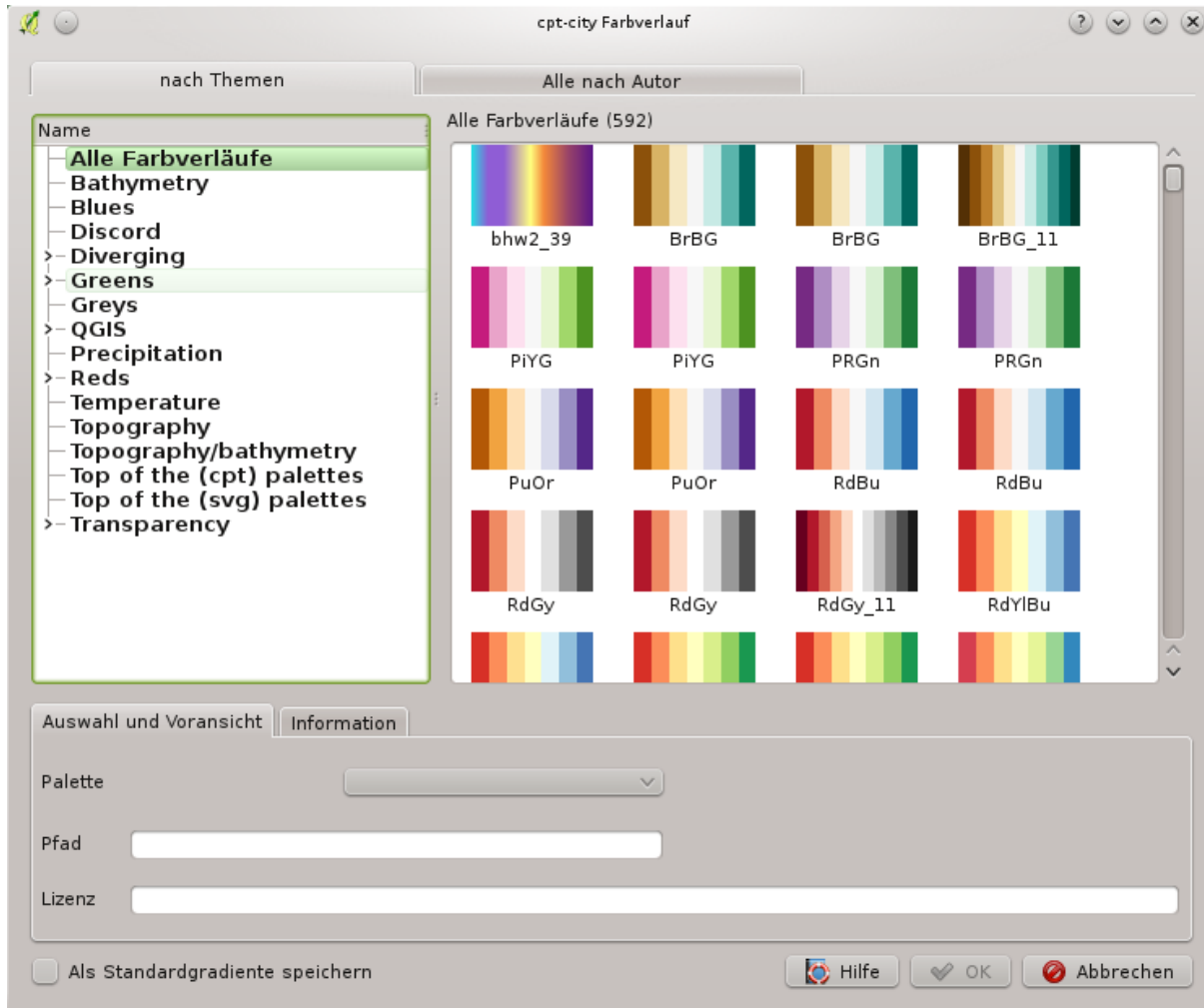


Abbildung 12.8: cpt-city Dialog mit hunderten von Farbverläufen 🐧

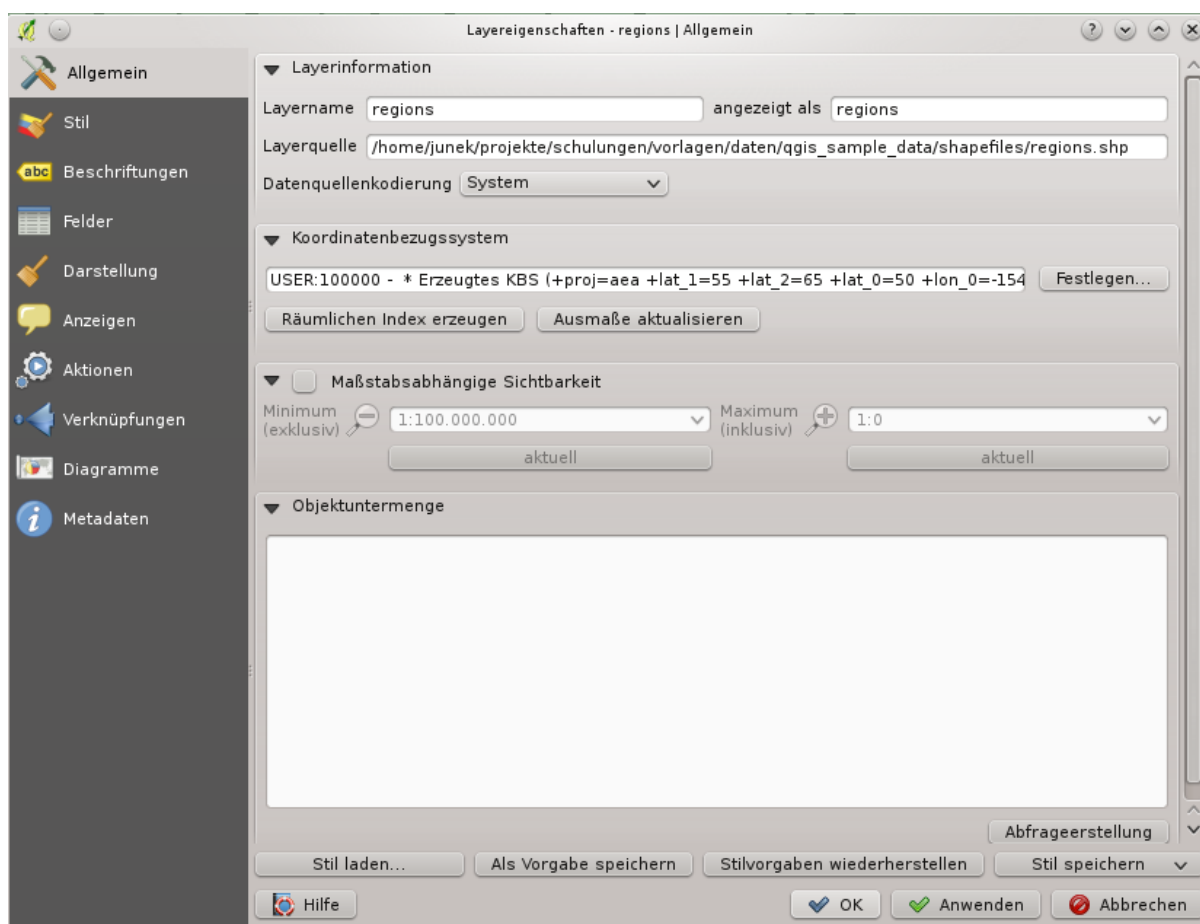


Abbildung 12.9: Vektor Layereigenschaften Dialog 





### 12.3.1 Menü Stil

Das Menü Stil stellt Ihnen ein umfassendes Werkzeug zum Darstellen und symbolisieren Ihrer Vektordaten zur Verfügung. Sie können *Layerdarstellung* → Werkzeuge, die für alle Vektordaten gleich sind, genauso wie spezielle Symbolisierungstools, die für die verschiedenen Arten von Vektordaten konzipiert wurden, verwenden.

#### Darstellungen

Der Renderer ist dafür verantwortlich ein Objekt zusammen mit dem richtigen Symbol zu zeichnen. Es gibt vier Arten von Renderern: Einzelsymbol, Kategorisiert, Abgestuft, Regelbasierend und Punktverdrängung. Es gibt keinen kontinuierliche Farbe Renderer da es in der Tat einfach ein spezieller Fall des Abgestuft Renderers ist. Die Kategorisiert und Abgestuft Renderer können erstellt werden indem ein Symbol und ein Farbverlauf festgelegt werden - Sie werden die Farben für Symbole angemessen einsetzen. Für Punktlayer ist ein Punktverdrängung Renderer erhältlich. Für jeden Datentyp (Punkte, Linien und Polygone) sind Symbollayertypen erhältlich. Abhängig vom ausgesuchten Renderer gibt es im Menü *Stil* verschiedene zusätzliche Bereiche. Im rechten unteren Teil des Symbologie Dialogs gibt es einen **[Symbol]** Knopf der den Zugang zum Stilmanager ermöglicht (siehe Abschnitt **vector\_style\_manager**). Mit dem Stilmanager können Sie bestehende Symbole bearbeiten und entfernen als auch neue Symbole hinzufügen.

Nachdem alle nötigen Veränderungen vorgenommen wurden kann das Symbol zur Liste der aktuellen Stilsymbole hinzugefügt werden (indem Sie **[Symbol]**  *Speichern* verwenden) und dann kann es auf einfache Weise später benutzt werden. Darüberhinaus können Sie den **[Stil speichern ...]**  Knopf um das Symbol als QGIS-Layerstildatei (.qml) oder SLD-Datei (.sld) zu speichern benutzen. SLDs können von jedem Darstellungstyp - Einzelsymbol, Kategorisiert, Abgestuft oder Regelbasierend - exportiert werden, aber wenn Sie ein SLD importieren wird entweder nur Einzelsymbol oder oder Regelbasierend erstellt. Das heisst dass Kategorisierte oder Abgestufte Stile nach Regelbasierend konvertiert werden. Wenn Sie diese Darstellungsarten beibehalten wollen müssen Sie zum QML-Format greifen. Auf der anderen Seite kann es sehr nützlich sein diese einfache Art, Stile nach Regelbasierend zu konvertieren, anzuwenden.

Wenn Sie den Darstellungstyp beim Einstellen des Stils eines Vektorlayers ändern werden die Einstellungen für das Symbol beibehalten. Beachten Sie dass dieses Vorgehen nur für eine Änderung funktioniert. Wenn Sie den Darstellungstyp wiederholt ändern gehen die Einstellungen für das Symbol verloren.

Wenn die Datenquelle eines Layers eine Datenbank ist (PostGIS oder Spatialite zum Beispiel) können Sie Ihren Layerstil innerhalb der Tabelle der Datenbank speichern. Klicken Sie einfach auf die *Stil speichern* Combobox und wählen Sie das **In Datenbank speichern** Element aus und füllen Sie dann den Dialog zum Definieren eines Stilnamens, hinzufügen einer Beschreibung und einer ui-Datei sowie ob der Stil ein voreingestellter Stil ist. Wenn Sie einen Layer von der Datenbank laden und bereits ein Stil für diesen Layer existiert lädt QGIS den Layer und seinen Stil. Sie können der Datenbank mehrere Stile hinzufügen. Übrigens wird nur einer der voreingestellte Stil sein.

---

#### **Tipp: Auswahl und Ändern von Mehrfachsymbolen**

Mit der Symbologie können Sie Mehrfachsymbole auswählen und Rechtsklicken um Farbe, Transparenz, Größe oder Breite der ausgewählten Einträge zu ändern.

---

#### **Einzelsymbol Darstellung**

Der Einzelsymbol Renderer wird verwendet um alle Objekte des Layers mit einem einfachen benutzerdefinierten Symbol darzustellen. Die Eigenschaften, die im Menü *Stil* angepasst werden können, hängen teilweise von dem Typ des Layers ab, wobei alle Typen die folgende Dialogstruktur befolgen. Im oberen linken Teil des Menüs gibt es einen Preview von den aktuellen Symbolen die dargestellt werden sollen. Im rechten Teil des Menüs gibt es eine Liste von Symbolen die bereits für den aktuellen Stil definiert wurden, welche durch Auswählen aus der Liste benutzt werden können. Das aktuelle Symbol kann mit dem Menü auf der rechten Seite verändert werden. Wenn Sie auf die erste Ebene im *Symbollayer* Dialog auf der linken Seite klicken ist es möglich grundlegende Parameter wie *Größe, Transparenz, Farbe* und `:guilabel:'Drehung'` zu definieren. Hier werden die Ebenen zusammengeführt.

#### **Kategorisierte Darstellung**

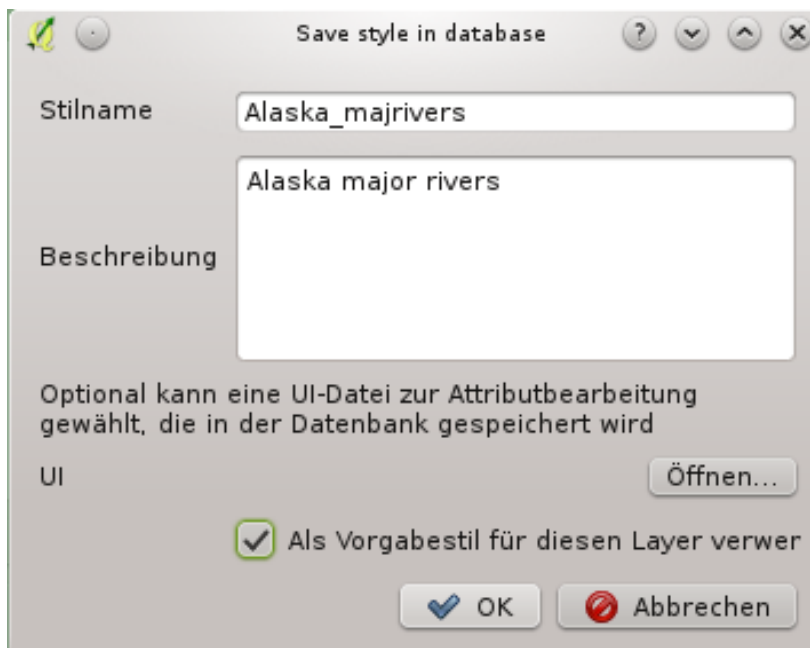


Abbildung 12.10: Stil in Datenbank speichern Dialog 

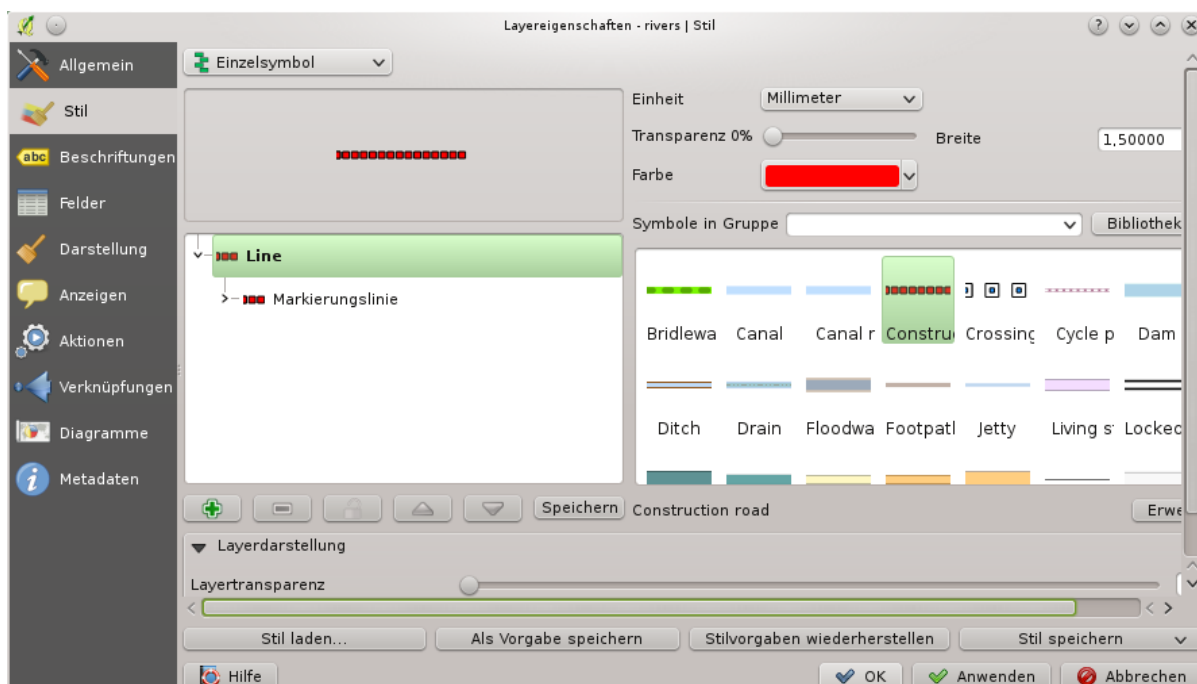



Abbildung 12.11: Linieneigenschaften Einzel-symbol 

Der Kategorisiert Renderer wird verwendet um alle Objekte eines Layers darzustellen indem man ein einfaches benutzerdefiniertes Symbol, dessen Farbe den Wert des ausgewählten Attributs des Objekts wiedergibt. Im *Stil* Menü können Sie folgendes auswählen:

- Das Attribut (indem Sie die Spalten Listbox oder die *Spaltenausdruck einstellen* Funktion benutzen, siehe *Ausdrücke*)
- Das Symbol (über die Auswahl Symbol)
- Die Farben (indem Sie die Farbverlauf Listbox verwenden)

Klicken Sie dann auf den **Klassifizieren** Knopf um Klassen aus den unterschiedlichen Werten der Attributspalte zu erstellen. Jede Klasse kann ausgeschaltet werden indem Sie das Kontrollkästchen links vom Klassennamen deaktivieren.

Sie können das Symbol, den Wert oder die Beschriftung verändern, klicken Sie einfach auf das Element, das Sie ändern wollen.

Ein Rechtsklick zeigt ein Kontextmenü für **Kopieren/Einfügen, Farbe ändern, Transparenz ändern, Ausgabereinheit ändern, Breite ändern**.

Mit dem **[Erweitert]** Knopf in der rechten unteren Ecke des Dialogs können Sie die Felder, die Drehungs- und Größenskalierungsinformationen enthalten, einstellen. Der Einfachheit halber werden in der Mitte des Menüs die Werte aller aktuell ausgewählten Attribute zusammen aufgeführt, inklusive der Symbole die dargestellt werden sollen.

Das Beispiel in *figure\_symbology\_2* zeigt den Dialog Kategorisierte Darstellung für den Vektorlayer *rivers* des QGIS -Beispieldatensatzes.

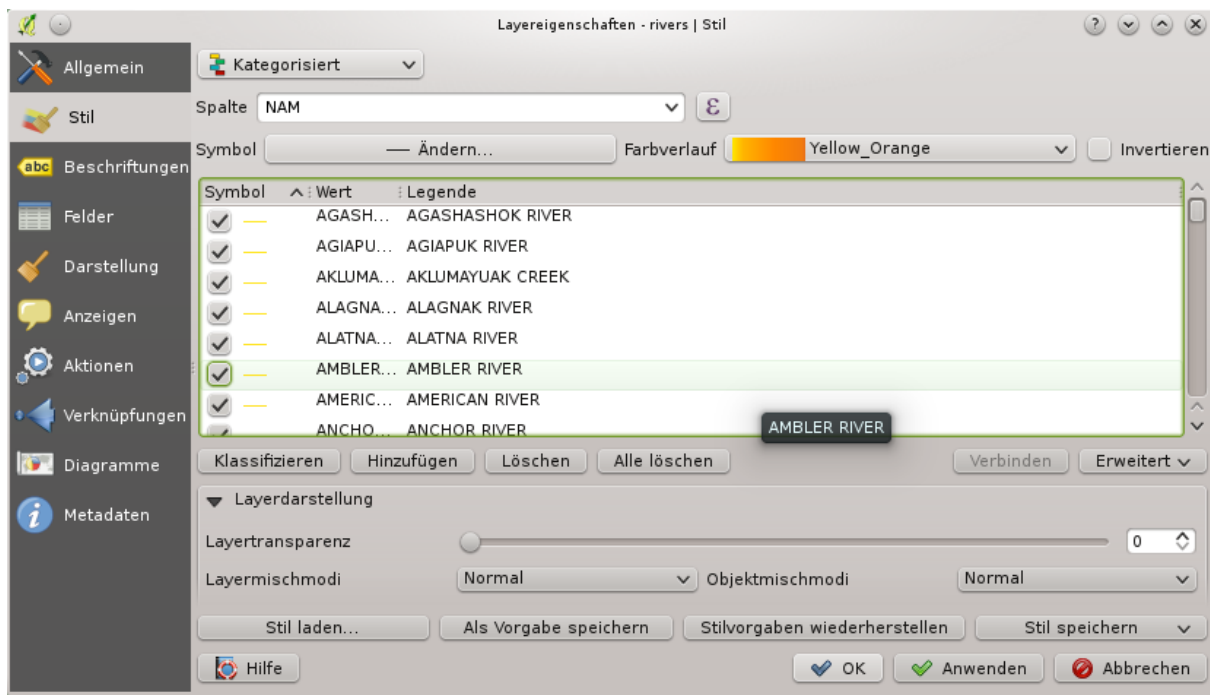


Abbildung 12.12: Kategorisierte Symbolisierungsoptionen 

### Abgestufte Darstellung

Der Abgestuft Renderer wird verwendet alle Objekte eines Layers darzustellen indem ein einfaches benutzerdefiniertes Symbol dessen Farbe die Zuweisung eines Objektattributes zu einer Klasse reflektiert.

Wie beim Kategorisiert Renderer können Sie mit dem Abgestuft Renderer die Drehung und Größenskalierung von angegebenen Spalten definieren.

Genauso können Sie -analog zum Kategorisierten Renderer - im Menü *Stil* auswählen:

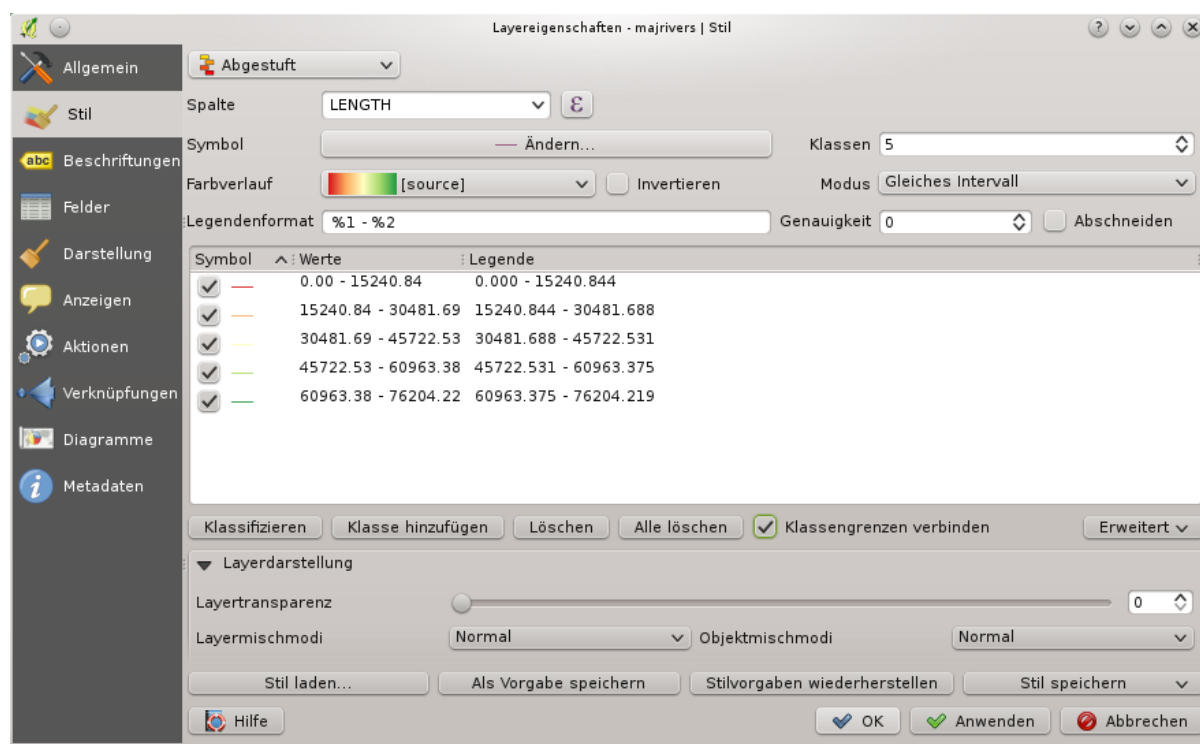


Abbildung 12.13: Abgestufte Symbolisierungsoptionen 

- Das Attribut (indem Sie die Spalten Listbox oder die  $\epsilon$ ... *Spaltenausdruck einstellen* Funktion benutzen, siehe das *Ausdrücke* Kapitel)
- Das Symbol (über die Auswahl Symbol)
- Die Farben (indem man die Farbverlaufsliste verwendet)

Zusätzlich können Sie die Anzahl der Klassen und auch den Modus für das Klassifizieren von Objekten innerhalb der Klassen (indem man die Modus-Liste verwendet). Die möglichen Modi sind:

- Gleiches Intervall: jede Klasse hat die gleich Größe (z.B. Werte von 0 bis 16 und 4 Klassen, jede Klasse hat eine Größe von 4);
- Quantil: jede Klasse beinhaltet die gleiche Anzahl von Elementen (nach der Idee eines Boxplots);
- Natürliche Unterbrechungen (Jenks): die Varianz innerhalb jeder Klasse ist minimal währenddessen die Varianz zwischen Klassen maximal ist;
- Standardabweichung: Klassen werden abhängig von der Standardabweichung der Werte erstellt;
- Schöne Unterbrechungen: das gleiche wie Natürliche Unterbrechungen nur dass die extremen Werte jeder Klasse Ganzzahlen sind.

Das Listenfeld im mittleren Teil des *Stil* Menüs führt die Klassen zusammen mit ihren Bereichen, Beschriftungen und Symbolen die dargestellt werden auf.

Klicken Sie auf den **Klassifizieren** Knopf um Klassen anhand des ausgewählten Modus zu erstellen. Jede Klasse kann anhand des Deaktivierens des Kontrollkästchens links neben dem Klassennamen ausgeschaltet werden.

Sie können das Symbol, den Wert oder die Beschriftung verändern, klicken Sie einfach auf das Element, das Sie ändern wollen.

Ein Rechtsklick zeigt ein Kontextmenü für **Kopieren/Einfügen**, **Farbe ändern**, **Transparenz ändern**, **Ausgabereinheit ändern**, **Breite ändern**.

Das Beispiel [figure\\_symbology\\_4](#) zeigt den Abgestufte Darstellung Dialog für den Vektorlayer rivers des QGIS-Beispieldatensatzes.

**Tipp: Thematische Karten anhand von Ausdrücken erstellen**

Kategorisierte und Abgestufte thematische Karten können jetzt anhand des Ergebnisses eines Ausdrucks erstellt werden. Im Eigenschaftendialog für Vektorlayer wurde die Attributauswahl um eine *Set column expression* Funktion ergänzt. Jetzt brauchen Sie nicht länger das Klassifikationsattribut in eine neue Spalte in Ihrer Attribut-tabelle schreiben wenn Sie wollen, dass das Klassifikationsattribut eine Zusammenstellung von mehreren Feldern wie z.B. eine Formel irgendeiner Art ist.

**Regelbasierende Darstellung**

Der regelbasierte Renderer wird verwendet um alle Objekte eines Layers anhand eines regelbasierten Symbols dessen Farbe die Zuordnung eines ausgewählten Objektattributs zu einer Klasse wiedergibt, darzustellen. Die Regeln basieren auf SQL-Anweisungen. Mit dem Dialog können Sie anhand von Filtern oder Maßstäben gruppieren und Sie können entscheiden ob Sie die Zeichenreihenfolge benutzen wollen oder nur die erste zutreffende Regel benutzen wollen.

Das Beispiel *figure\_symbology\_5* zeigt den Dialog der regelbasierenden Darstellung für den Layer rivers des QGIS Beispieldatensatzes.

Um eine Regel zu erstellen aktivieren Sie eine bestehende Zeile mit einem Doppelklick oder klicken Sie auf '+' und klicken Sie auf eine neue Regel. Im Dialog *Regeleigenschaften* können Sie eine Beschriftung für die Regel definieren. Klicken Sie den *...* Knopf um den Ausdruckseditor zu öffnen. Klicken Sie auf *Felder und Werte* in der **Funktionsliste** um sich alle Attribute der aktuellen Attributtabelle anzeigen zu lassen. Um ein Attribut zum **Ausdruck** Feld hinzuzufügen machen Sie einen Doppelklick auf den Namen in der *Felder und Werte*-Liste. *Allgemein können Sie die verschiedenen Felder, Werte und Funktionen verwenden um den Berechnungsausdruck zu konstruieren oder Sie können ihn einfach in das Feld eingeben (see :ref:'vector\_expressions')*. Sie können eine neue Regel mit Kopieren und Einfügen anhand der rechten Maustaste in eine bestehende Regel erstellen. Sie können auch die 'ELSE'-Regel, die ausgeführt wird wenn keine der anderen Regeln zutrifft, verwenden. Seit QGIS 2.6 erscheint die Beschriftung für die Regeln in einem Pseudobaum in der Kartenlegende. Machen Sie einfach einen Doppelklick auf die Regeln in der Kartenlegende und ein Stil Menü der Layereigenschaften erscheint, das die Regel, die der Hintergrund für das Symbol im Pseudobaum ist, zeigt.

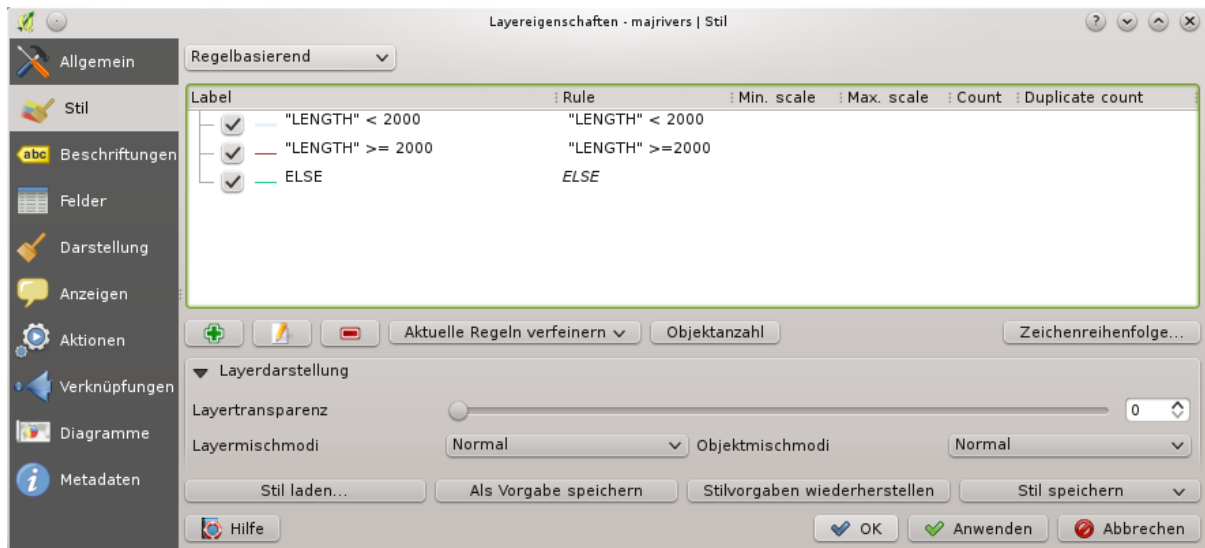


Abbildung 12.14: Regelbasierte Symbolisierungsoptionen

**Punkt-Verschiebung**

Für Punktlayer gibt es eine Darstellungsart, mit der es möglich ist, sämtliche Punkte eines Layers auch dann darzustellen, wenn sie sich teilweise an derselben Stelle befinden. Die Punkte werden dabei um ein Zentrumssymbol herum auf einem Versatzkreis angeordnet und dargestellt.

**Tipp: Symbologie exportierten**

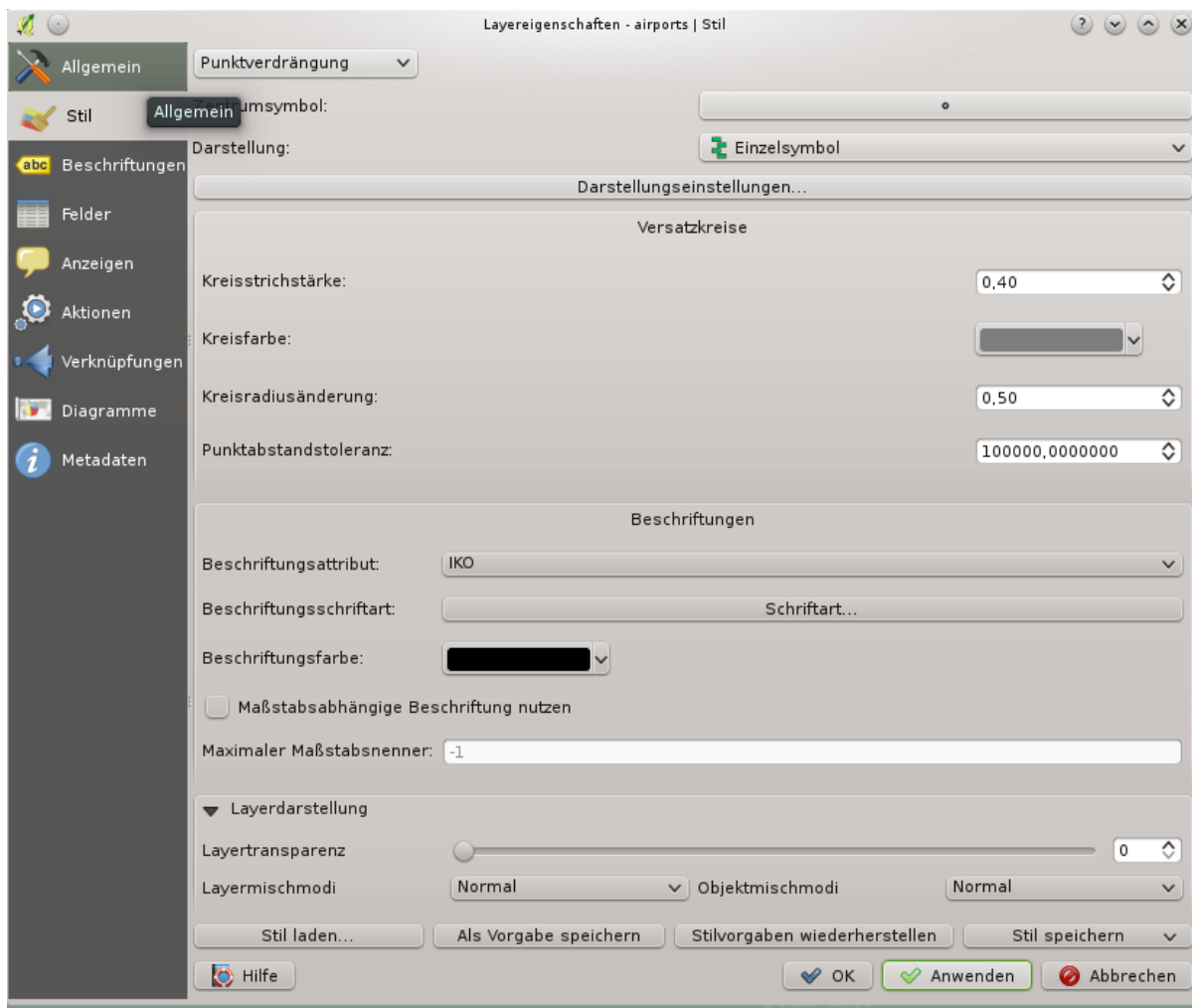


Abbildung 12.15: Dialog Punktverdrängung 🐧

Sie haben die Option die Vektorsymbologie von QGIS nach Google \*.kml, \*.dxf und MapInfo \*.tab Dateien zu exportieren. Öffnen Sie einfach das Rechte-Maustasten-Menü des Layers und klicken Sie auf *Speichern als ...* um den Namen der Ausgabedatei und ihr Format festzulegen. Verwenden Sie im Dialog das *:guilabel:Darstellungsexport* Menü um die Symbologie entweder als *Objektdarstellung* → oder *Symbollayerdarstellung* → zu speichern. Wenn Sie Symbollayer verwendet haben wird empfohlen die zweite Einstellung zu benutzen.

### Umgekehrte Polygone

Die Umgekehrte Polygon Darstellung ermöglicht es dem Anwender die Außenseite der Polygone des Layers zu füllen. Wie vorher können Sie Subrenderer auswählen. Diese Subrenderer sind das gleiche wie bei dem Hauptdarstellungen.

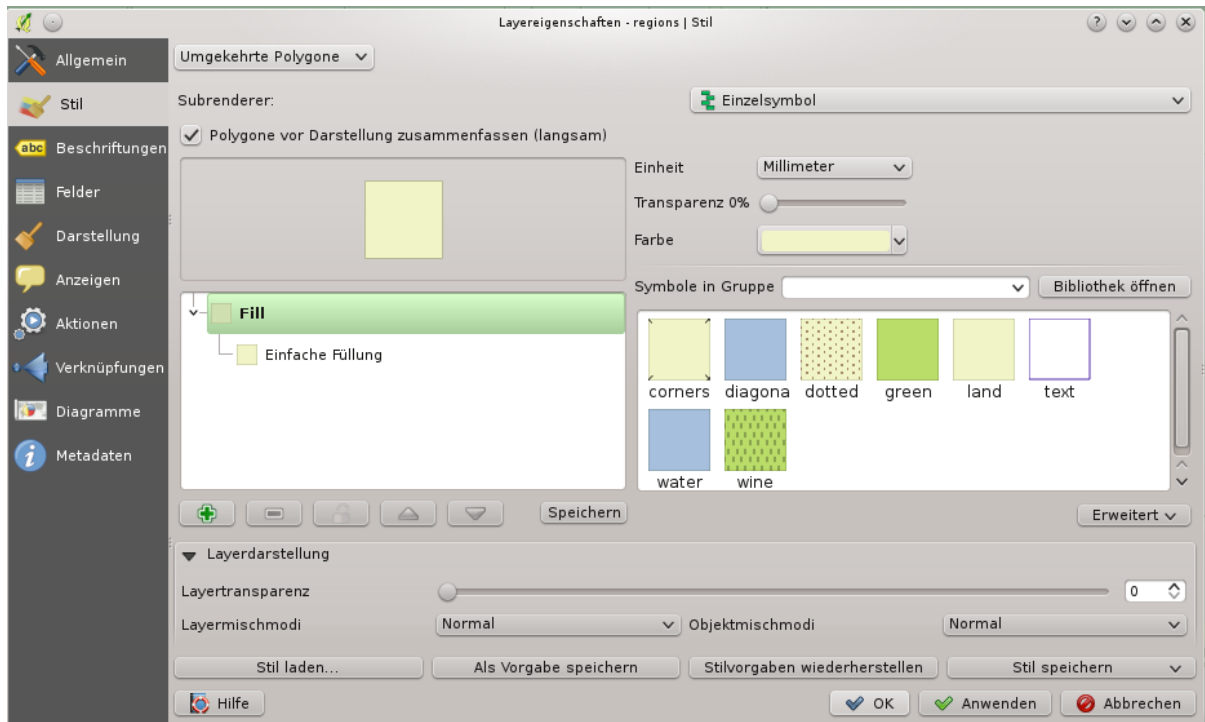










Abbildung 12.16: Umgekehrte Polygone Dialog 

### Farbwahl

Unabhängig vom Stiltyp, der verwendet wird, zeigt der *Farbe wählen* Dialog e san wenn Sie klicken um eine Farbe zu wählen - entweder Füllungs- oder Rahmenfarbe. Der Dialog besitzt 4 verschiedene Reiter, die es Ihnen ermöglichen Farben anhand des  Farbverlaufs,  Farbkreises, der  Farbproben oder der  Farbwahl auszuwählen.

Egal welche Methode Sie verwenden, die ausgewählte Farbe wird immer durch Farbglegler für HSV- (Hue, Saturation, Value) und RGB- (Rot, Blau, Grün) Werte beschrieben. Es gibt auch einen *Deckkraft*-Regler um den Transparenzgrad einzustellen. Im linken unteren Teil des Dialogs können Sie einen Vergleich zwischen *Aktuell* und *Alt* Farbe, die Sie zur Zeit auswählen und im rechten unteren Teil haben Sie die Option eine Farbe hinzuzufügen, aus der Sie gerade einen Farbkopf gemacht haben.

Mit dem  Farbverlauf oder mit dem  Farbkreis können Sie alle möglichen Farbkombinationen suchen. Dennoch gibt es auch noch andere Möglichkeiten. Indem Sie *Farbproben*  verwenden können Sie aus einer vorausgewählten Liste wählen. Diese ausgewählte Liste wird mit einer drei Methoden *Recent colors*, *Standard colors* oder *Project colors* gefüllt.

Eine andere Option ist die  Farbwahl zu benutzen, die es Ihnen ermöglicht eine Farbprobe unter Ihrem Mauszeiger in jedem Teil von QGIS oder sogar von einer anderen Anwendung zu nehmen, indem Sie die Leertaste drücken.



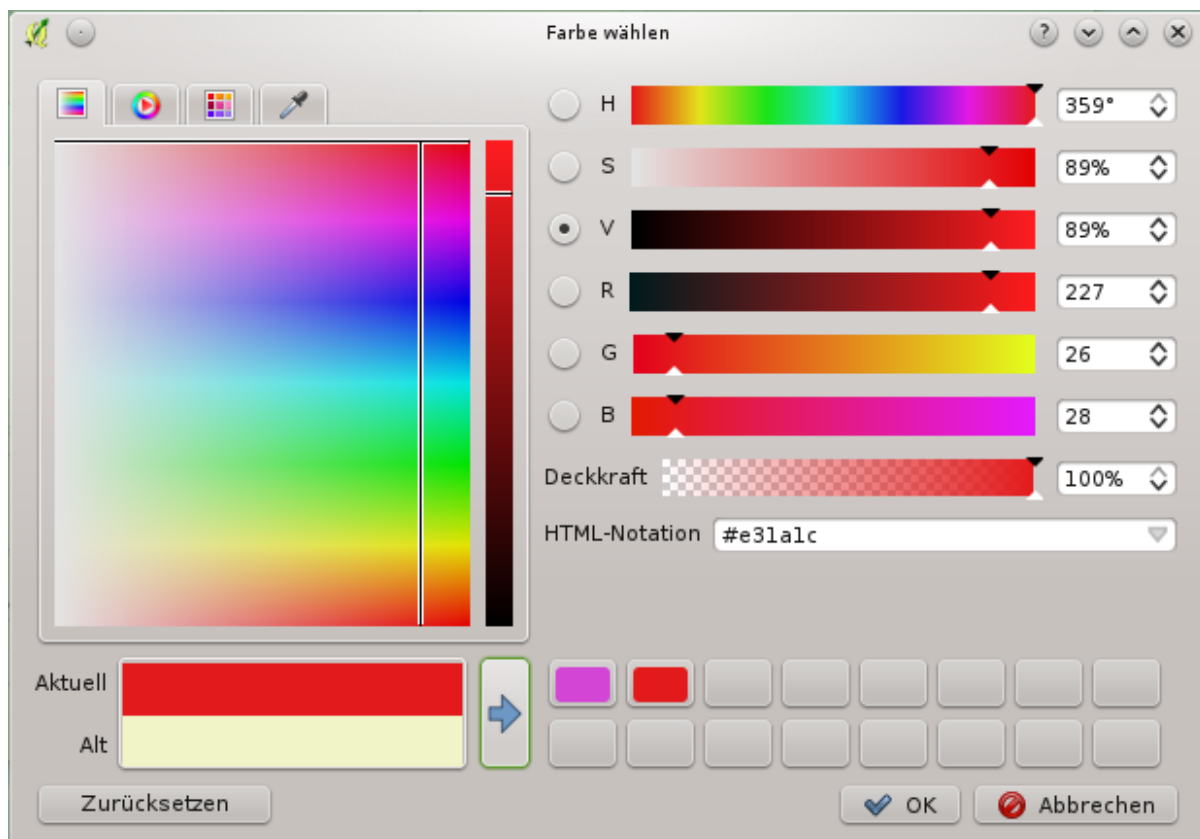



Abbildung 12.17: Farbwahl Verlauf Reiter 

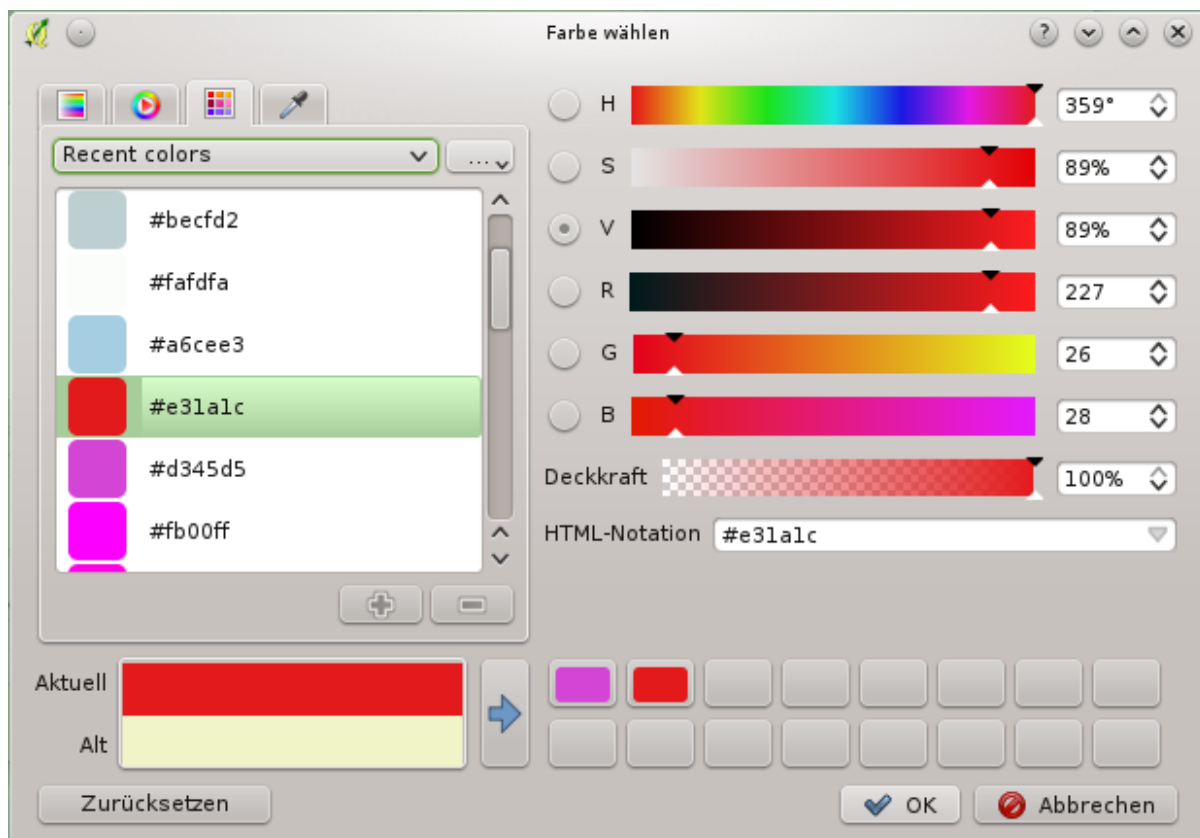



Abbildung 12.18: Farbwahl Farbproben Dialog 



Bitte beachten Sie, dass die Farbwahl betriebssystemabhängig ist und aktuell nicht von OSX unterstützt wird.

**Tipp:** \*\* schnelle Farbwahl + Farben kopieren/einfügen\*\*

Sie können schnell aus *Recent colors*, *Standard colors* auswählen oder einfach eine Farbe *Kopieren* oder *Einfügen* indem Sie den Drop-down-Pfeil klicken, der einer aktuellen Farbbox folgt.

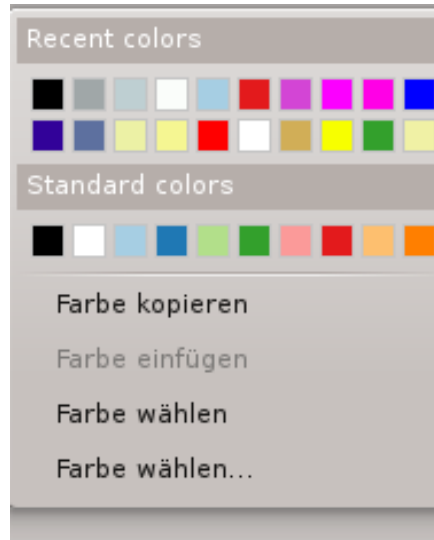




Abbildung 12.19: Schnelle Farbwahl Menü 

## Layerdarstellung

- *Layertransparenz* : Sie können den unten liegenden Layer in der Kartenansicht mit diesem Werkzeug sichtbar machen. Verwenden Sie den Slider um die Sichtbarkeit Ihres Vektorlayers an Ihre Bedürfnisse anzupassen. Sie können auch eine genaue Definition des Prozentgrades der Sichtbarkeit im Menü neben dem Slider vornehmen.
- *guilabel:Layermischmodi* und *Objektmischmodi*: Sie können spezielle Darstellungseffekte mit diesen Werkzeugen, die Sie vorher nur von Grafikprogrammen gekannt haben, erzielen. Die Pixel der oben auf liegenden und darunter liegenden Layer werden anhand der unten beschriebenen Einstellungen gemischt.
  - Normal: Dies ist der Standardmischmodus, der den Alphakanal des oben liegenden Pixels mit dem darunter liegenden Pixel vermischt. Die Farben werden nicht vermischt.
  - Heller: Dies wählt das Maximum jeder Komponente der Vordergrund- und Hintergrundpixel. Seien Sie sich bewusst dass die Ergebnisse zackig und hart aussehen können.
  - Bildschirm: Helle Pixel der Quelle werden über die des Ziels gezeichnet wohingegen dunkle Pixel nicht verwendet werden. Dieser Modus ist am nützlichsten für das Mischen der Textur eines Layers mit einem anderen Layer (z.B. kann man eine Schummerung dazu verwenden einen anderen Layer mit einer Textur zu versehen).
  - Abwedeln: das Abwedeln erhellt und sättigt unten liegende Pixel auf Basis der Helligkeit des oben liegenden Pixels. Demzufolge erhöhen hellere oben liegende Pixel die Sättigung und Helligkeit des unten liegenden Pixels. Dies funktioniert am Besten wenn die oben liegenden Pixel nicht zu hell sind; andernfalls ist der Effekt zu extrem.
  - Addition: Dieser Mischmodus fügt einfach die Pixelwerte eines Layers denen eines anderen Layers hinzu. Im Falle von Werten größer 1 (im Fall von RGB) wird weiß dargestellt. Dieser Modus ist dafür geeignet Objekte hervorzuheben.
  - Dunkler: Dies erstellt ein Ergebnispixel das die kleinste Komponente der Vordergrund und Hintergrundpixel erhält. Wie das Aufhellen neigen die Ergebnisse dazu zackig und hart zu sein.


- Multiplizieren: Hier werden die Nummern für jedes Pixel des oben liegenden Layers mit den entsprechenden Pixeln des unteren Layers multipliziert. Das Ergebnis sind dunklere Bilder.
- Einbrennen: Dunklere Farben im oben liegenden Layer bewirken ein Verdunkeln des unten liegenden Layers. Einbrennen kann dazu benutzt werden um unten liegende Layer zu optimieren und zu colorieren.
- Überlagern: Dieser Modus kombiniert die Multiplizieren und Bildschirm Mischmodi. Im Ergebnispixel werden helle Bereiche heller und dunkle Bereiche dunkler.
- Weiches Licht: Dieses ist dem Überlagern sehr ähnlich nur dass anstelle Multiplizieren/Bildschirm Einbrennen/Abwedeln verwendet wird. Hier soll das Leuchten eines weichen Lichtes auf ein Bild nachgeahmt werden.
- Hartes Licht: Auch Hartes Licht ist dem Überlagerungsmodus sehr ähnlich. Hier soll die Projektion eines sehr intensiven Lichts auf ein Bild nachgeahmt werden.
- Unterschied: Unterschied subtrahiert das oben liegende Pixel von dem unten liegenden Pixel oder andersherum um immer einen positiven Wert zu bekommen. Das Mischen mit Schwarz produziert keinen Unterschied, da die Differenz mit allen Farben Null ist.
- Abziehen: Dieser Mischmodus zieht einfach die Pixelwerte eines Layers von dem anderen ab. Im Fall von negativen Werten wird Schwarz dargestellt.



## 12.3.2 Menü Beschriftungen

Die  Beschriftungen Kernanwendung stellt intelligentes Beschriften für Punkt-, Linien- und Polygonlayer zur Verfügung und erfordert nur wenige Parameter. Diese neue Anwendung unterstützt auch spontan transformierte Layer. Die Kernfunktionen der Anwendung wurden überarbeitet. In QGIS gibt es eine Anzahl von anderen Funktionen die das Beschriften verbessern. Die folgenden Menüs wurden erstellt um die Vektorlayer zu beschriften:

- Text
- Formatierung
- Puffer
- Hintergrund
- Schatten
- Platzierung
- Darstellung


Lassen Sie uns sehen wie die Menüs für verschiedene Vektorlayer benutzt werden können. **Punktlayer beschriften**

Starten Sie QGIS und laden Sie einen Punktlayer. Aktivieren Sie den Layer in der Legende und klicken Sie auf das  Layerbeschriftungseinstellungen Icon in der QGIS Werkzeugleiste.

Der erste Schritt besteht darin das  *Layer beschriften mit* Kontrollkästchen zu aktivieren und eine Attributspalte für das Beschriften auszuwählen. Klicken Sie  wenn Sie ausdrucksbasierte Beschriftungen definieren wollen - Siehe [labeling\\_with\\_expressions](#).

Die folgenden Schritte beschreiben ein einfaches Beschriften ohne die Verwendung der *Datendefinierten Übersetzung* Funktionen, die neben den Drop-Down Menüs untergebracht sind.

Sie können den Textstil im *Text* Menü definieren (siehe [Figure\\_labels\\_1](#)). Verwenden Sie die *Schriftart Groß-/Kleinschreibung* Option um die Textdarstellung zu beeinflussen. Sie haben die Möglichkeit den Text mit 'Nur Großbuchstaben', 'Nur Kleinbuchstaben' oder 'Erstes Zeichen groß' darzustellen. Verwenden Sie Mischmodi um Effekte die Sie von Grafikprogrammen kennen zu erstellen (siehe [blend\\_modes](#)).

Im Menü *Formatierung* können Sie einen Buchstaben für einen Zeilenumbruch in den Beschriftungen mit der 'Bei Zeichen umbrechen' Funktion definieren. Verwenden Sie die  *Zahlenformatierung* Option um die Nummern

in einer Attributtabelle zu formatieren. Hier können Dezimalstellen eingefügt werden. Wenn Sie diese Option aktivieren werden erst einmal drei Dezimalstellen als Standard eingestellt.

Um einen Puffer zu erstellen aktivieren Sie einfach die  *Textpuffer zeichnen* im Menü *Puffer*. Die Pufferfarbe ist variabel. Auch hier können Sie Mischmodi verwenden ( siehe [blend\\_modes](#)).

Wenn das  *Pufferfüllung einfärben* Kontrollkästchen aktivieren, wird es mit teiltransparentem Text interagieren und gemischte Farbtransparenzerggebnisse liefern. Das Abschalten der Pufferfüllung behebt das Problem (es sein denn der innere Aspekt der Pufferausdehnung überschneidet sich mit der Textfüllung) und ermöglicht das Erstellen von umrandetem Text.

Im Menü *Hintergrund* können Sie mit *Größe X* und *Größe Y* die Form Ihres Hintergrunds definieren. Verwenden Sie *Größenart* um einen zusätzlichen ‘Puffer’ in Ihren Hintergrund einzufügen. Der Hintergrund besteht dann aus dem Puffer mit dem Hintergrund in *Größe X* und *Größe Y*. Sie können eine *Drehung* festlegen wobei Sie zwischen ‘Mit Beschriftung abgleichen’, ‘Beschriftungsversatz’ und ‘Fest’ wählen können. Indem Sie ‘Beschriftungsversatz’ und ‘Fest’ verwenden, können Sie den Hintergrund rotieren. Definieren Sie *X-,Y-Versatz* mit X und Y Werten und der Hintergrund wird versetzt. Wenn Sie *X-, Y-Radius* verwenden erhält der Hintergrund runde Ecken. Auch hier ist es möglich den Hintergrund mit den darunterliegenden Layern in der Kartenansicht anhand von *Mischmodi* zu mischen (siehe [blend\\_modes](#)).

Verwenden Sie das *Schatten* Menü für einen benutzerdefinierten *Schattenwurf*. Das Zeichnen des Hintergrunds ist sehr variabel. Wählen Sie zwischen ‘Niedrigste Beschriftungskomponente’, ‘Text’, ‘Puffer’ und ‘Hintergrund’.

Der *Versatz*-Winkel hängt von der Orientierung der Beschriftung ab. Wenn Sie das Kontrollkästchen  *Globalen Schatten verwenden* dann ist der Ausgangspunkt des Winkels immer nach Norden orientiert und hängt nicht von der Orientierung der Beschriftung ab. Sie können das Erscheinungsbild des Schattens mit *Radius verschmieren* beeinflussen. Je höher die Nummer desto weicher sind die Schatten. Das Erscheinungsbild des Schattenwurfs kann auch durch das Benutzen eines Mischmodus verändert werden (siehe [blend\\_modes](#)).

Verwenden Sie das Menü *Platzierung* für die Platzierung der Beschriftung und die Beschriftungspriorität. Wenn Sie die  *Abstand vom Punkt* Einstellung verwendne haben jetzt die Option *Quadrant* zum Platzieren Ihrer Label zu benutzen. Zusätzlich können Sie den Winkel der Beschriftungsplatzierung mit der *Drehung* Einstellung verwenden. So ist die Platzierung in einem bestimmten Quadranten mit einer bestimmten Drehung möglich.

Im Menü *Darstellung* können sie Beschriftungs- und Objektoptionen definieren. Unter *Beschriftungsoptionen* können Sie jetzt die maßstabsabhängige Sichtbarkeitseinstellung vornehmen. Sie können QGIS davon abhalten nur ausgewählte Beschriftungen darzustellen indem Sie das  *Alle Beschriftungen auf diesem Layer anzeigen (einschließlich kollidierender)* Kontrollkästchen benutzen. Unter *Objektoptionen* können Sie definieren ob jeder Teil eines Multipart Features beschriftet werden soll. Es ist möglich zu definieren ob die Nummer von Objekten die beschriftet werden soll begrenzt ist und  *Möglichst keine Objekte durch Beschriftungen verdecken*.

### Linienlayer beschriften

Der erste Schritt ist es das  *Layer beschriften mit* Kontrollkästchen im *Beschriftungen* Menü zu aktivieren und eine Attributspalte für das Beschriften auszuwählen. Klicken Sie *E...* wenn Sie ausdrucksbasierte Beschriftungen definieren wollen - siehe [labeling\\_with\\_expressions](#).

Danach können Sie den Textstil im *Text* Menü definieren. Hier können Sie die gleichen Einstellungen wie für Punktlayer verwenden.

Auch im Menü *Formatierung* sind die gleichen Einstellungen wie für Punktlayer möglich.

Das Menü *Puffer* hat die gleichen Funktionen wie in Abschnitt [labeling\\_point\\_layers](#).

Das Menü *Hintergrund* hat die gleichen Einträge wie in Abschnitt [labeling\\_point\\_layers](#) beschrieben.

Auch das *Schatten* Menü hat die gleichen Einträge wie in section [labeling\\_point\\_layers](#) beschrieben.

Im Menü *Platzierung* finden Sie spezielle Einstellungen für Linienlayer. Die Beschriftung kann  *Parallel*,  *Gebogen* oder  *Horizontal* platziert werden. Mit der  *Parallel* und radiobuttonoffl *Gebogen* Option können Sie die Position  *Über Linie*,  *Auf der Linie* und  *Unter Linie* definieren. Es ist möglich mehrere Optionen auf einmal auszuwählen. In diesem Fall wird QGIS nach der optimalen Position der Beschriftung suchen.

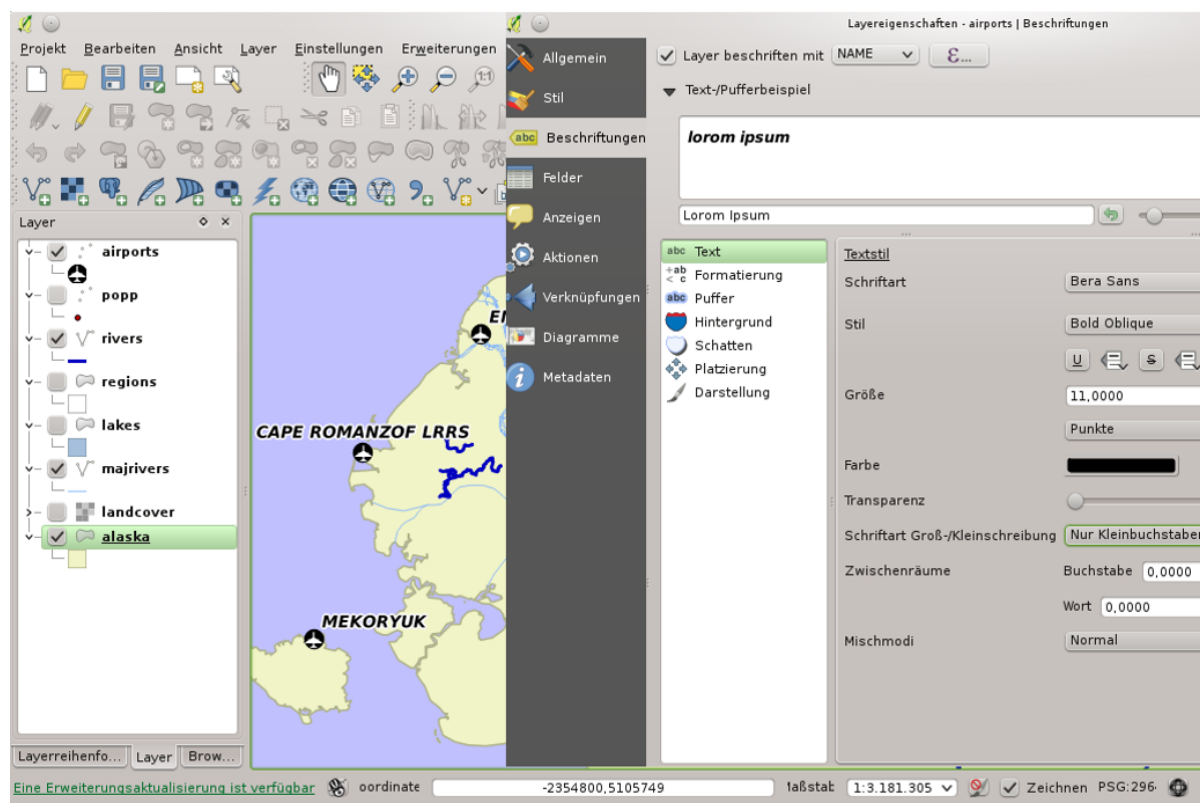


Abbildung 12.20: Intelligentes Beschriften von Punktlayern 


Zusätzlich können Sie *Größter Winkel zwischen Zeichen auf Kurven* definieren wenn Sie die  *Gebogen* Option auswählen (siehe [Figure\\_labels\\_2](#)).

Sie können eine Minimaldistanz zum Wiederholen von Beschriftungen einstellen. Der Abstand kann in mm oder Karteneinheiten angegeben werden.

Unter *Platzierung* werden weitere Optionen dargestellt, z.B. *Gebogen* und *Parallel* Platzierungen ermöglichen es dem Anwender die Position des Labels einzustellen (über, unter oder auf der Linie), die *Entfernung* von der Linie und bei *Gebogen* kann der Anwender auch den inneren/äußeren größten Winkel zwischen Zeichen auf Kurven einstellen.

Das *Darstellung* Menü hat fast die gleichen Einträge wie das bei Punktlayern. In den *Objektoptionen* können Sie jetzt *Objekte nicht beschriften, wenn kürzer als*.

### Polygonlayer beschriften

Der erste Schritt ist es das  *Layer beschriften mit* Kontrollkästchen zu aktivieren und eine Attributspalte für das Beschriften auszuwählen. Klicken Sie  wenn Sie ausdrucksbasierte Beschriftungen definieren wollen - Siehe [labeling\\_with\\_expressions](#).

Definieren Sie den Textstil im *Text* Menü. Die Einträge sind die gleichen wie die für Punkt- und Linienlayer.

Das Menü *Formatierung* ermöglicht es Ihnen mehrzeilige Zeilen zu formatieren wie schon bei Punkt- und Linienlayern.

Wie bei Punkt- und Linienlayern können Sie einen Textpuffer im *Puffer* Menü erstellen.

Verwenden Sie das Menü *Hintergrund* um einen komplexen benutzerdefinierten Hintergrund für den Polygonlayer zu erstellen. Sie können das Menü wie bei Punkt- und Linienlayern benutzen.

Die Einträge in dem Menü *Schatten* sind die gleichen wie für Punkt- und Linienlayer.

Im Menü *Platzierung* finden Sie spezielle Einstellungen für Polygonlayer (siehe [Figure\\_labels\\_3](#)).  *Abstand vom Zentrum*,  *Horizontal (langsam)*,  *radiobuttonoff* : `guiabel: 'Um Zentrum`,  *Frei (langsam)* und  *Nach*

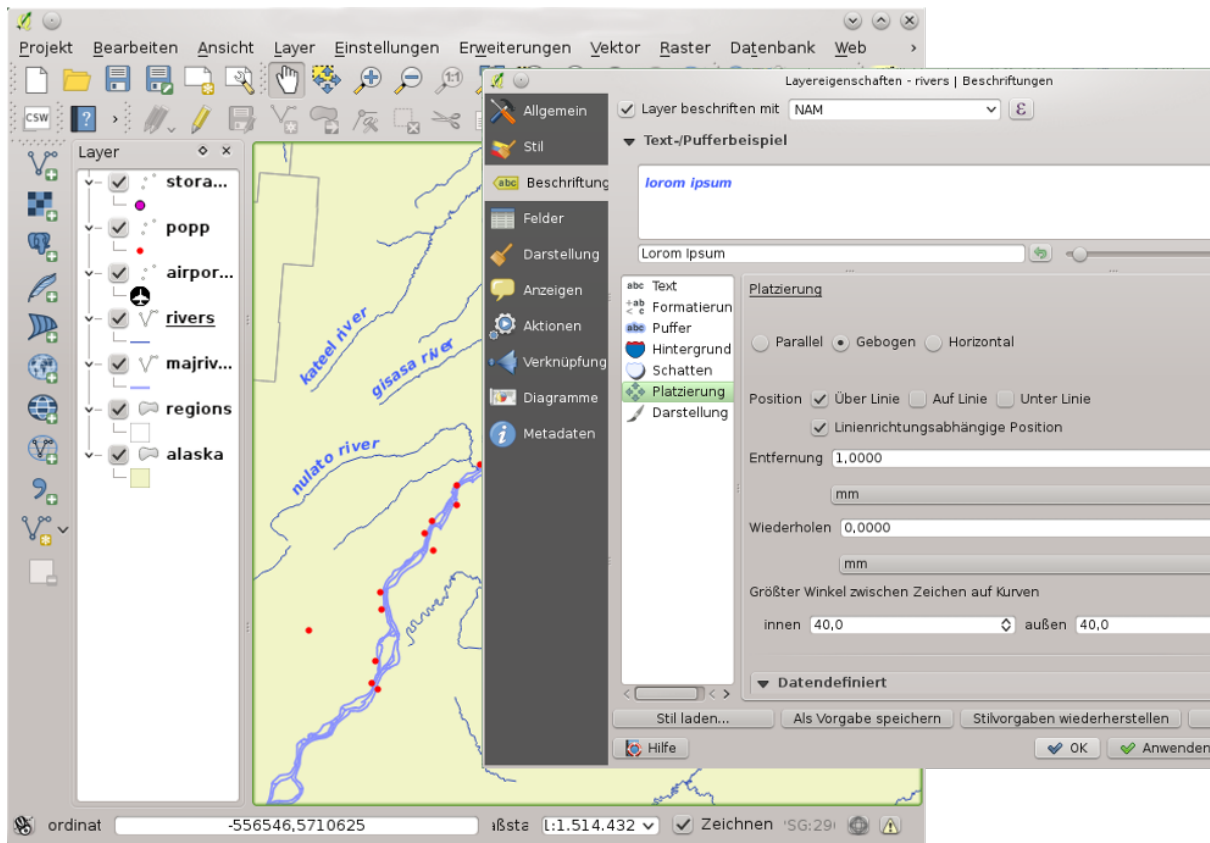




Abbildung 12.21: Intelligentes Beschriften von Linienlayern 

Umfang sind möglich.

In den  *Abstand vom Zentrum* Einstellungen können Sie festlegen ob der Zentroid sich auf  *sichtbarem Polygon* oder  *ganzem Polygon* bezieht. Das heißt dass entweder der Zentroid für das Polygon das Sie auf der Karte sehen verwendet wird oder der Zentroid für das ganze Polygon bestimmt wird egal ob Sie das ganze Objekt auf der Karte sehen. Sie können hier Ihre Beschriftung anhand von Quadranten platzieren sowie Versatz und Drehung definieren. Die  *Um Zentrum* Einstellung macht es möglich die Beschriftung um einen Zentroiden herum mit einer bestimmten Distanz zu platzieren. Auch hier können Sie  *sichtbarem Polygon* oder  *ganzem Polygon* für den Zentroiden definieren. Mit den  *Nach Umfang* Einstellungen können Sie eine Position und einen Abstand für die Beschriftung definieren. Für die Position sind  *Über Linie*,  *Auf Linie*,  *Unter Linie* und  *Linienrichtungsabhängige Position* möglich.

Was das Auswählen der Beschriftungsplatzierung anbetrifft, ergeben sich mehrere Optionen. Wie bei Punktverdrängung können Sie Entfernung für die Polygonumrandung wählen und die Beschriftung um den Polygonumfang wiederholen.

Die Einträge in das Menü *Darstellung* sind die gleichen wie für Linienlayer. Sie können auch *Objekte nicht beschriften wenn kürzer als* in den *Objektoptionen*. **Ausdrucksbasierte Beschriftungen definieren**

Mit QGIS können Sie Ausdrücke benutzen um Objekte zu beschriften. Klicken Sie einfach das  Icon im Menü  *Beschriftungen* des *Eigenschaften Dialogs*. In [figure\\_labels\\_4](#) sehen Sie einen Beispielausdruck um die Alaska Regionen mit Namen- und Flächengröße, abhängig vom Feld 'NAME\_2', etwas beschreibenden Text und die Funktion '\$area()' in Kombination mit 'format\_number()' damit sie besser aussieht.

Mit dem ausdrucksbasierten Beschriften können Sie auf einfache Art und Weise arbeiten. Alles worum Sie sich kümmern müssen ist dass Sie alle Elemente (Zeichenketten, Felder und Funktionen) mit einem String-Verkettungszeichen '|' kombinieren und dass Felder in "doppelten Anführungszeichen" und Zeichenketten in 'einfachen Anführungszeichen' geschrieben werden. Lassen Sie uns einen Blick auf einige Beispiele werfen:

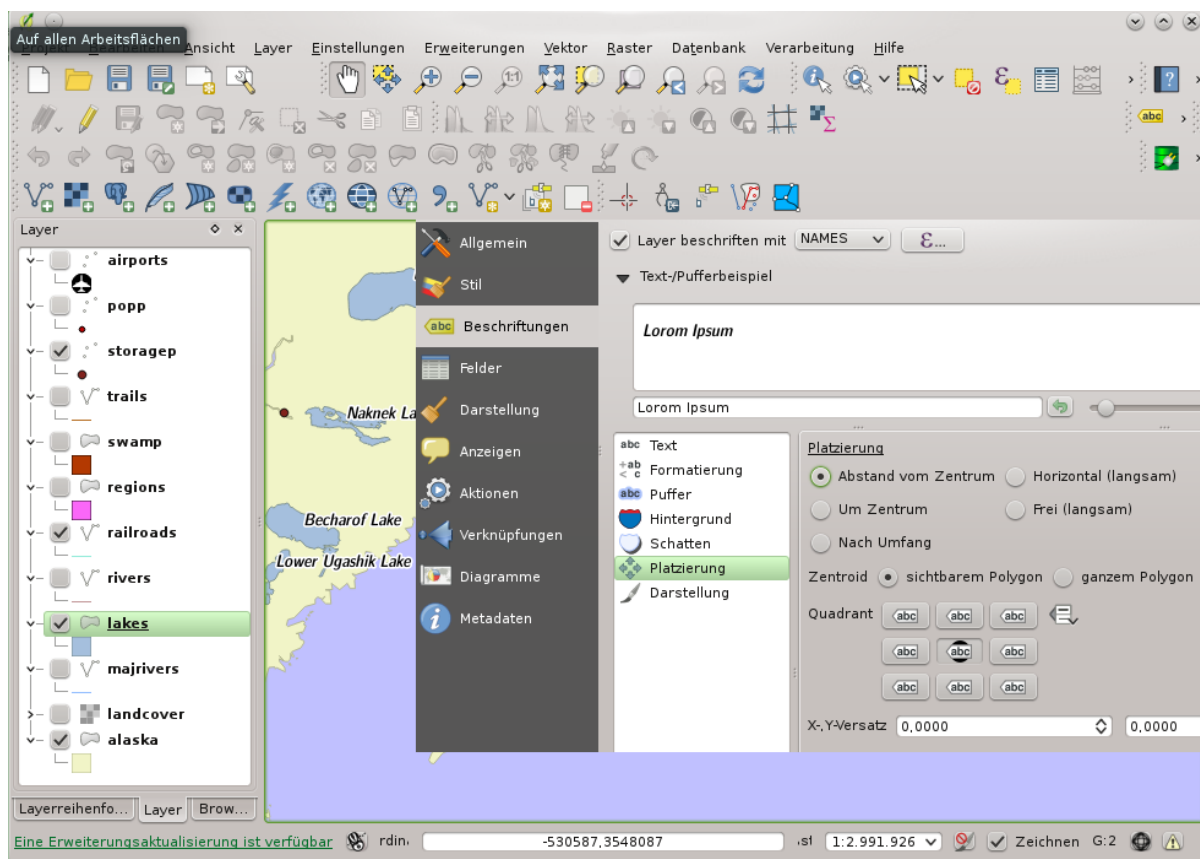


Abbildung 12.22: Intelligentes Beschriften von Polygonlayern 🐧

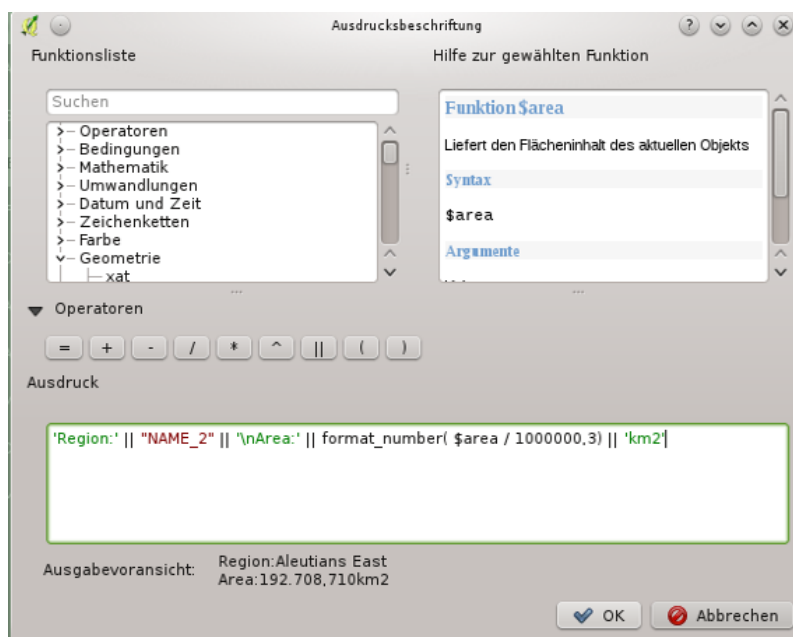


Abbildung 12.23: Ausdrücke für das Beschriften verwenden 🐧



```
# label based on two fields 'name' and 'place' with a comma as separator
"name" || ', ' || "place"

-> John Smith, Paris

# label based on two fields 'name' and 'place' separated by comma
'My name is ' || "name" || 'and I live in ' || "place"

-> My name is John Smith and I live in Paris

# label based on two fields 'name' and 'place' with a descriptive text
# and a line break (\n)
'My name is ' || "name" || '\nI live in ' || "place"

-> My name is John Smith
    I live in Paris

# create a multi-line label based on a field and the $area function
# to show the place name and its area size based on unit meter.
'The area of ' || "place" || 'has a size of ' || $area || 'm²'


-> The area of Paris has a size of 105000000 m²



# create a CASE ELSE condition. If the population value in field
# population is <= 50000 it is a town, otherwise a city.
'This place is a ' || CASE WHEN "population <= 50000" THEN 'town' ELSE 'city' END

-> This place is a town
```





Wie Sie im Ausdruckeditor sehen können stehen Ihnen hunderte von Funktionen zur Verfügung um einfache und sehr komplexe Ausdrücke zum Beschriften Ihrer Daten in QGIS zu erstellen. Siehe das *Ausdrücke* Kapitel für weitere Informationen und ein Beispiel zu Ausdrücken.

### Datendefinierte Übersteuerung für das Beschriften

Mit der datendefinierten Übersteuerung werden die Einstellungen für das Beschriften von Einträgen in der Attributtabelle überschrieben. Sie können die Funktion mit dem Rechte-Maus-Knopf aktivieren und deaktivieren. Fahren Sie über das Symbol und Sie sehen die Information über die datendefinierte Übersteuerung einschließlich des aktuellen Definitionsfeldes. Wir beschreiben jetzt ein Beispiel indem wir die datendefinierte Übersteuerungsfunktion für die  Funktion verwenden (siehe [figure\\_labels\\_5](#)).

1. Importieren Sie `lakes.shp` aus dem QGIS Beispieldatensatz.
2. Doppelklicken Sie den Layer um die Layereigenschaften zu öffnen. Klicken Sie auf *Beschriftungen* und *Platzierung*. Wählen Sie  *Abstand vom Punkt*.
3. Suchen Sie nach den *Datendefiniert* Einträgen. Klicken Sie das  Icon um den Feldtyp für die *Koordinate* zu definieren. Wählen Sie 'xlabel' für X und 'ylabel' für Y aus. Die Icons sind jetzt in gelb hervorgehoben.
4. Zoomen Sie auf einen See.
5. Gehen Sie zur Beschriftung Werkzeugleiste und klicken Sie das  Icon. Jetzt können Sie die Beschriftung manuell in eine andere Position verschieben (siehe [figure\\_labels\\_6](#)). Die neue Position der Beschriftung ist in den 'xlabel' und 'ylabel' Spalten der Attributtabelle gespeichert.

### 12.3.3 Menü Felder

 Innerhalb des *Felder* Menüs können die Feldattribute des ausgewählten Datensatzes manipuliert werden. Die Knöpfe  *Neue Spalte* und  *Spalte löschen* können benutzt werden wenn der Datensatz im  Bearbeitungsstatus umschalten Modus ist.

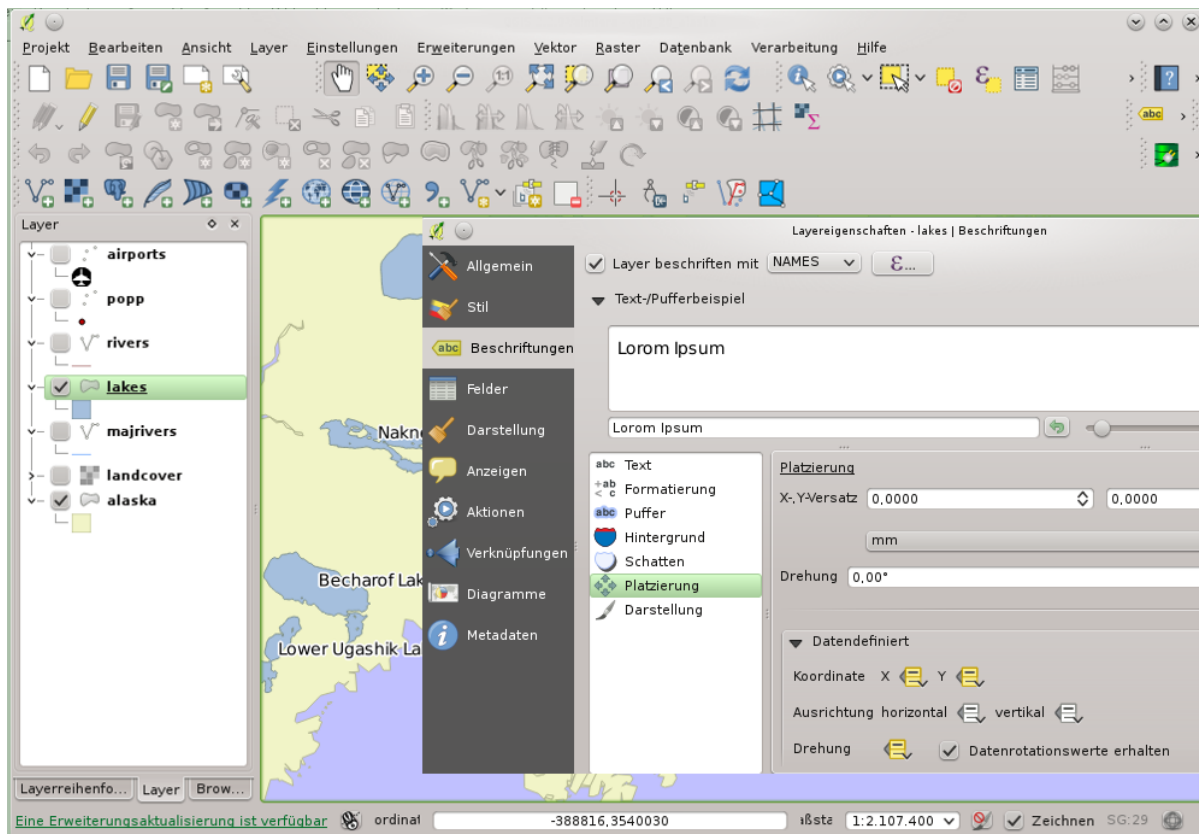


Abbildung 12.24: Das Beschriften von Polygonlayern mit datendefinierter Übersteuerung 

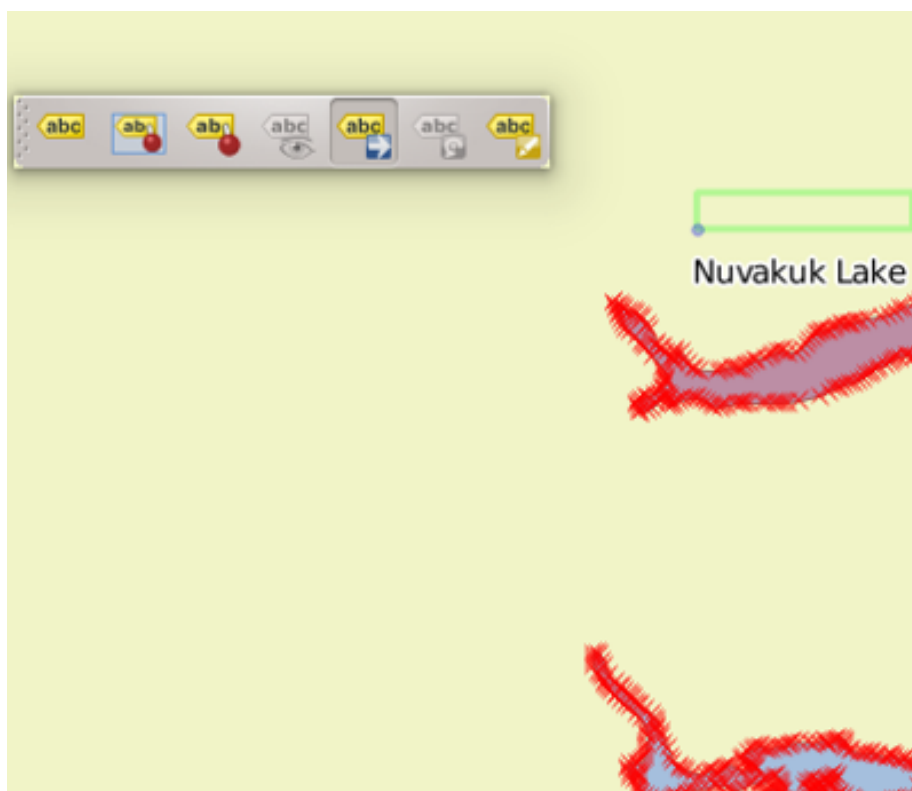



Abbildung 12.25: Beschriftungen verschieben 



## Bearbeitungselement

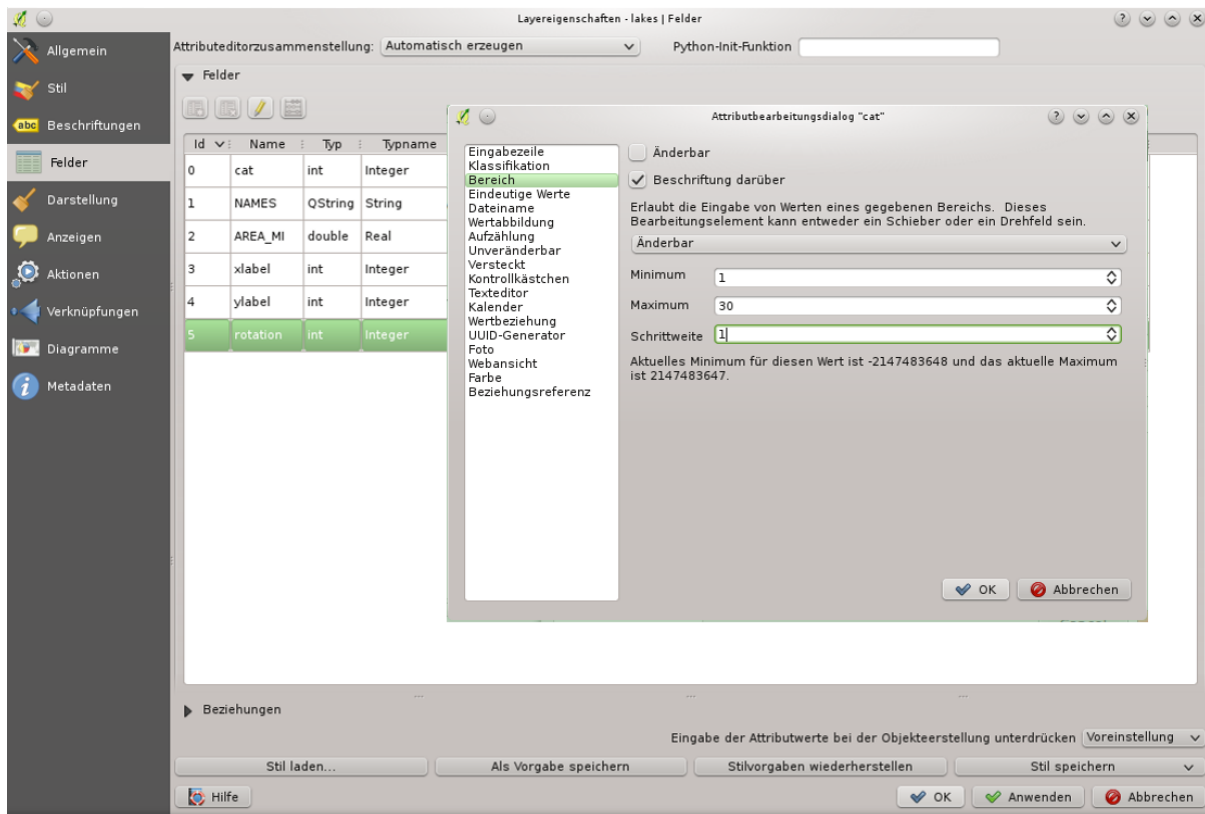




Abbildung 12.26: Dialog um ein Bearbeitungselement für eine Attributspalte auszuwählen 🐧

Innerhalb des Menüs *Felder* finden Sie auch eine **Bearbeitungselement** Spalte. Diese Spalte kann dazu benutzt werden Werte oder eine Spanne von Werten zu definieren die zu der bestimmten Attributtabellenspalte hinzugefügt werden dürfen. Wenn Sie auf den **[Eingabezeile]** Knopf klicken öffnet sich ein Dialog indem Sie verschiedene Elemente definieren können. Diese Elemente sind:

- **Kontrollkästchen:** Gibt ein Kontrollkästchen wieder und Sie können definieren welches Attribut der Spalte hinzugefügt wird wenn das Kontrollkästchen aktiviert ist oder nicht.
- **Klassifikation:** Auswahlliste mit den Attributwerten, die im Menü *Stil* als Legendentyp *Eindeutiger Wert* für die Klassifikation benutzt werden.
- **\*\*Farbe\*:** Stellt einen Farbknopf dar, der es dem Anwender ermöglicht eine Farbe von einem Farbdialogfenster auszuwählen.
- **Datum/Zeit:** Stellt ein Linienfeld dar, das ein Kalender-Widget zum Öffnen eines Datums, einer Zeit oder beidem, darstellt. Der Spaltentyp muss Text sein. Sie könne ein benutzerdefiniertes Format auswählen, einen Pop-up-Kalender, etc.
- **Aufzählung:** Öffnet eine Kombobox mit Werten die innerhalb eines Spaltentyps benutzt werden können. Dieses wird aktuell nur vom PostgreSQL Provider unterstützt.
- **Dateiname:** Vereinfacht die Dateiauswahl durch einen Dateiauswahldialog.
- **Versteckt:** Ein verstecktes Attribut ist unsichtbar. Der Anwender kann den Inhalt nicht sehen.
- **Foto:** Feld enthält einen Dateinamen für ein Bild. Die Breite und Höhe des Feldes kann definiert werden.
- **Bereich:** Erlaubt Ihnen numerische Werte eines bestimmten Wertebereichs festzulegen. Das Bearbeitungselement kann entweder ein Schieber oder ein Drehfeld sein.
- **Beziehungsreferenz:** Mit diesem Element können Sie das Objektformular des referenzierten Layers in das Objektformular des aktuellen Layers einbetten. Siehe *Ein-Zu-Mehrere-Beziehungen erstellen*.

- **Texteditor** (voreingestellt): Dies öffnet einen Textbearbeitungsfeld mit dem Sie einfachen Text oder mehrere Zeilen verwenden können. Wenn Sie mehrzeilig gewählt haben können Sie auch HTML wählen.
- **Eindeutige Werte**: Sie können einen der Werte die bereits in der Attributtabelle verwendet werden aus-suchen. Wenn 'Änderbar' aktiviert ist, wird eine Eingabezeile mit Autovervollständigungsunterstützung gezeigt, andernfalls wird eine Kombobox verwendet.
- **UUID Generator**: Erstellt ein schreibgeschütztes UUID (Universally Unique Identifiers)-Feld wenn es leer ist.
- **Wertabbildung**: Eine Kombobox mit vordefinierten Elementen. Der Wert ist im Attribut gespeichert, die Beschreibung wird in der Kombobox gezeigt. Sie können Werte manuell definieren oder sie aus einem Layer oder einer CSV-Datei laden.
- **Wertbeziehung**: Bietet Werte aus einer verknüpften Tabelle in einem Auswahlménü an. Sie können Layer, Schlüsselspalte und Wertspalte auswählen.
- **Webansicht**: Feld enthält eine URL. Die Breite und Höhe des Feldes ist variabel.

Mit der **Attributeditorzusammenstellung** können Sie jetzt integrierte Formulare für Dateneingabeaufgaben definieren (siehe [figure\\_fields\\_2](#)). Wählen Sie 'Mit Drag und Drop zusammenstellen' und eine Attributspalte.

Verwenden Sie das  Icon um eine Kategorie die dann während der Digitalisiersitzung gezeigt wird zu erstellen (siehe [figure\\_fields\\_3](#)). Im nächsten Schritt werden dann die relevanten Felder mit dem  Icon der Kategorie zugeordnet. Sie können noch mehr Kategorien erstellen und dabei die gleichen Felder verwenden. Wenn Sie eine neue Kategorie erstellen fügt QGIS einen neuen Reiter für die Kategorie im integrierten Formular ein.

Andere Optionen im Dialog sind 'Autogenerate' und 'Provide ui-file'. 'Automatisch erzeugen' erstellt einfach Editoren für alle Felder und tabelliert sie. Bei der 'UI-Datei verwenden' Option können Sie komplexe Dialoge, die mit dem Qt-Designer erstellt werden, verwenden. Das Verwenden einer UI-Datei ermöglicht ein hohes Maß an Freiheit beim Erstellen eines Dialogs. Detaillierte Informationen können Sie unter <http://nathanw.net/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/> finden.

QGIS Dialoge können eine Python-Funktion beinhalten, die aufgerufen wird wenn der Dialog geöffnet wird. Verwenden Sie diese Funktion um Ihren Dialogen eine zusätzliche Logik hinzuzufügen. Ein Beispiel ist (im Modul MyForms.py):

```
def open(dialog, layer, feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget, "My line edit")
```

Die Referenz in der Python Init Function sieht in etwa so aus: MyForms.open

MyForms.py muss in den PYTHONPATH wie in .qgis2/python oder innerhalb des Projektordners eingetragen werden.

### 12.3.4 Menü Allgemein



Verwenden Sie dieses Menü um allgemeine Einstellungen für den Vektorlayer zu machen. Es stehen mehrere Optionen zur Verfügung:

Layerinformation

- Ändern Sie den Anzeigenamen des Layers in *angezeigt als*
- Definieren Sie die *Layerquelle* des Vektorlayers
- Definieren Sie *Datenquellenkodierung* um providerspezifische Optionen zu definieren und um in der Lage zu sein die Datei zu lesen.

Koordinatenbezugssystem

- Verwenden Sie *Festlegen* für das Koordinatensystem. Hier können Sie die Projektion eines bestimmten Vektorlayers ansehen oder verändern.

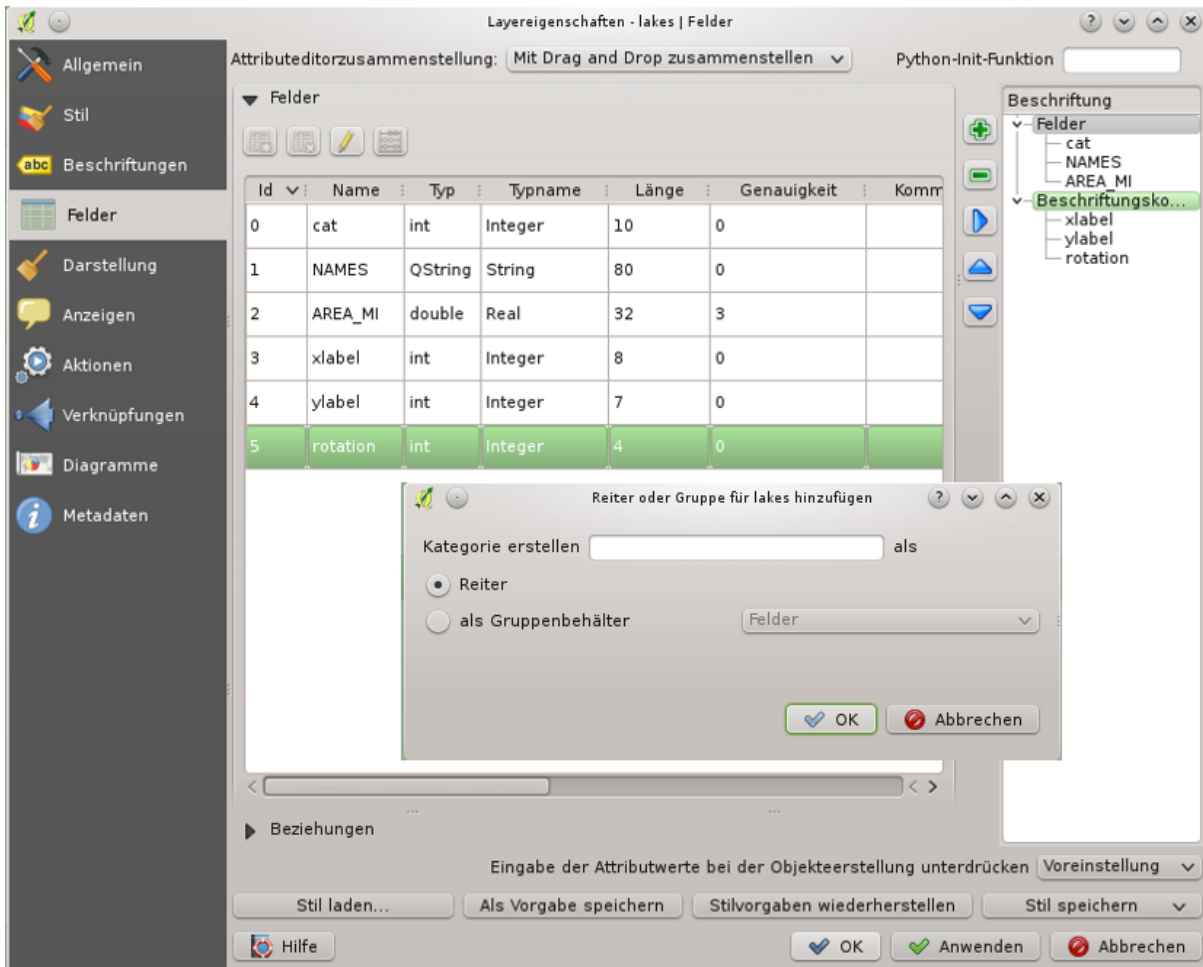


Abbildung 12.27: Dialog **Attributeditorzusammenstellung** um Kategorien zu erstellen

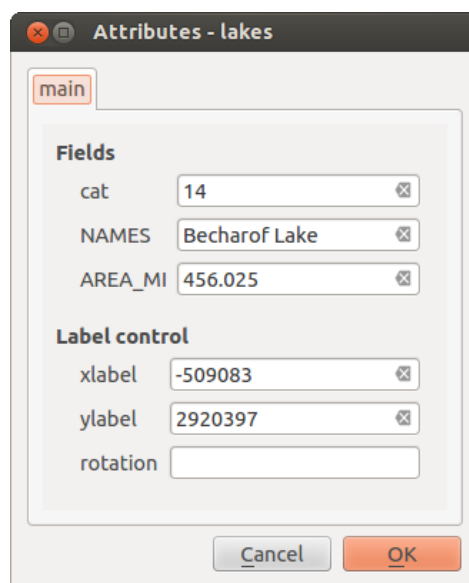


Abbildung 12.28: Das Ergebnis eines integrierten Formulars in einer Dateneingabesitzung.

- Erstellen Sie einen *Räumlichen Index* (nur OGR-unterstützte Formate)
- Die *Ausmaße aktualisieren* für einen Layer
- Sehen Sie sich die Projektion eines spezifischen Vektorlayers an oder ändern Sie diese indem Sie auf *Festlegen ...* klicken.

*Maßstabsabhängige Sichtbarkeit*

- Sie können den *Maximum (inklusive)* und *Minimum (exklusiv)* Maßstab einstellen. Der Maßstab kann auch mit den **[Aktuell]** Knöpfen eingestellt werden.

Objektuntermenge

- Mit dem **[Abfrageerstellung]** Knopf können Sie eine Untermenge von Objekten im Layer erstellen, die dann dargestellt wird (lesen Sie auch die Informationen in *Abfrageeditor*).

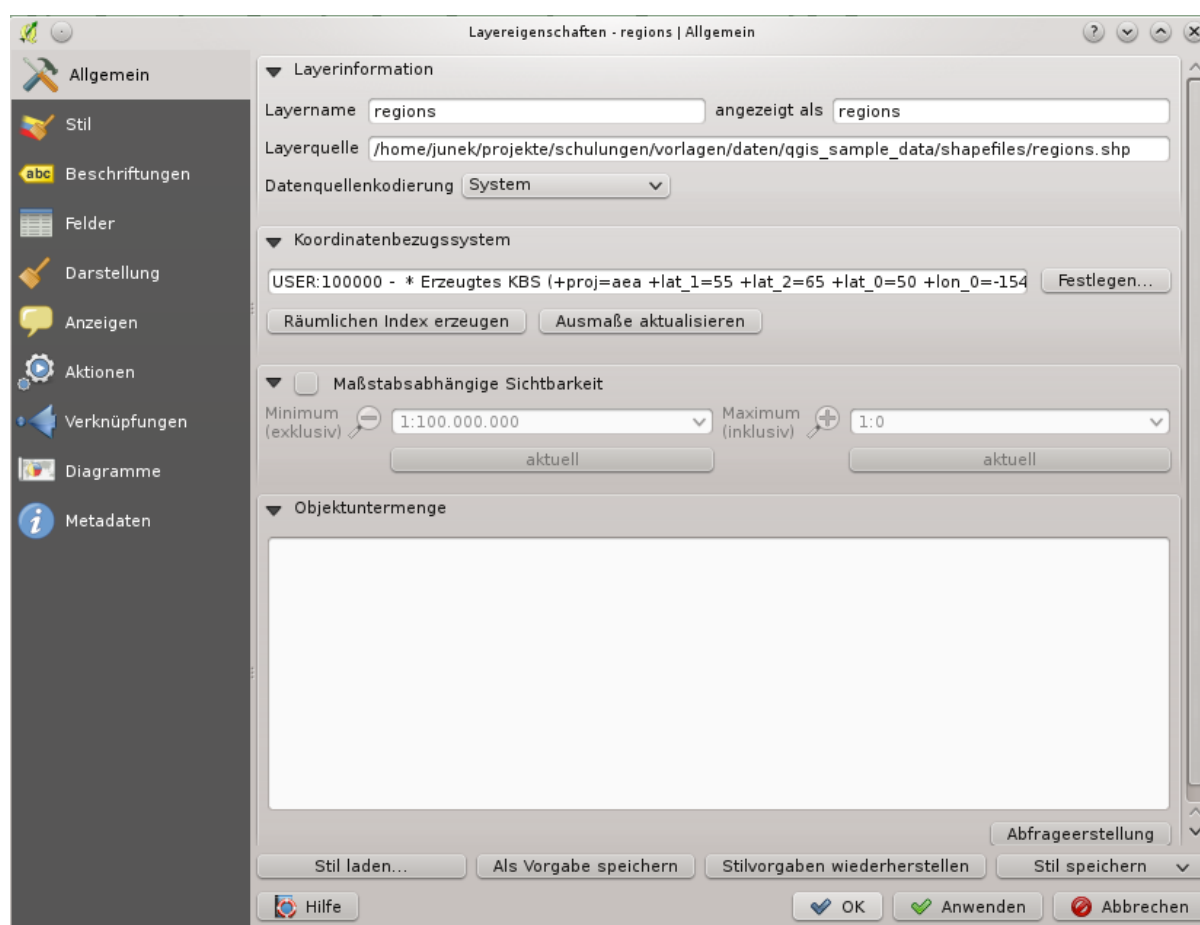




Abbildung 12.29: Menü Allgemein in Vektorlayer Eigenschaften Dialog 

### 12.3.5 Menü Darstellung

QGIS 2.2 führt Unterstützung für Geometrievereinfachung ein. Dies kann die Darstellungszeit verbessern wenn viele komplexe Objekte in kleinem Maßstab gezeichnet werden sollen. Dieses Objekt kann in den Layereinstellungen ein- oder ausgeschaltet werden indem  *Geometrie vereinfachen* verwendet wird. Es gibt zusätzlich eine neue globale Einstellung die die Vereinfachung standardmäßig für neu hinzugefügte Layer einschaltet (siehe Abschnitt *Optionen*): **Beachten Sie:** Geometrievereinfachung kann zu Splitterpolygonen und ungenauer Darstellung führen wenn offset-basierte Symbollayer verwendet werden.

### 12.3.6 Menü Darstellung

 Dieses Menü ist speziell für Kartenhinweise erstellt worden. Es enthält eine neue Funktion: Kartentippanzeigetext in HTML. Während Sie weiterhin ein  *Feld* das angezeigt werden soll wenn Sie über ein Objekt auf der Karte gehen wählen können ist es jetzt möglich HTML Code der eine komplexe Anzeige erstellt wenn man darüber geht einzugeben. Um Kartenhinweise zu aktivieren wählen Sie die Menüoption *Ansicht → Kartenhinweise*. Figure Display 1 zeigt ein Beispiel für HTML Code.

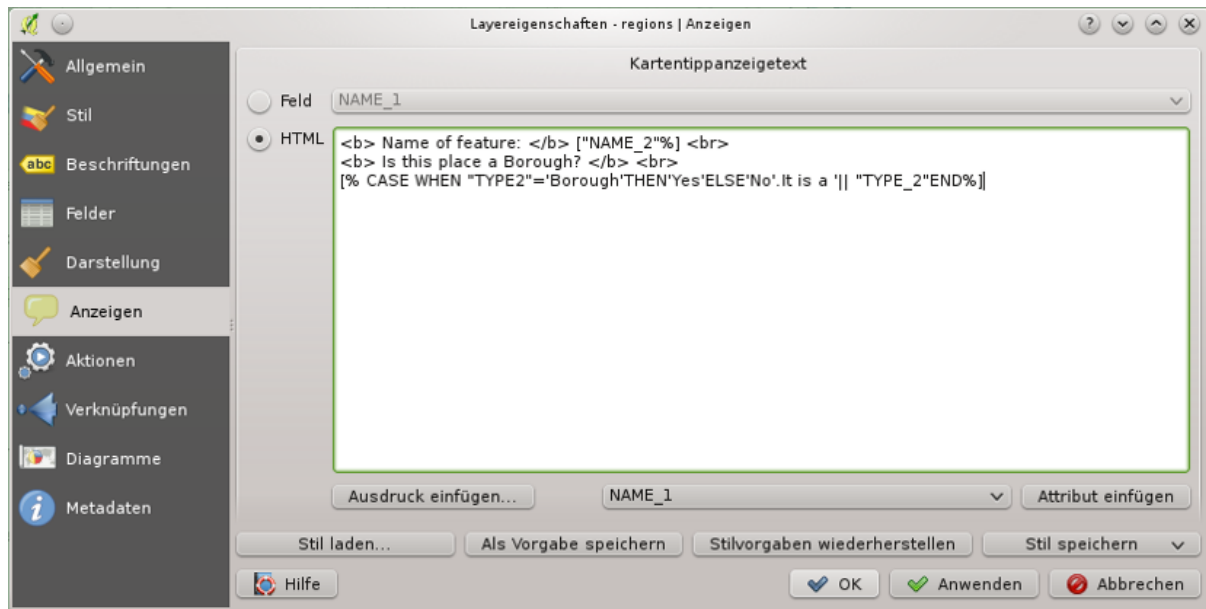



Abbildung 12.30: HTML-Code für Kartenhinweis 

### 12.3.7 Menü Aktionen

 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 um Parameter an ein Web-Reporting-Tool weiterzugeben.

Aktionen auf Basis von Attributen sind sinnvoll wenn sie häufig eine externe Anwendung starten oder eine Internetseite auf Basis von einem oder mehreren Werten in Ihrem Vektorlayer visualisieren wollen. Sie sind in 6 Typen aufgeteilt und können wie folgt verwendet werden:

- Allgemein, Mac, Windows und Unix Aktionen starten einen externen Prozess.
- Python Aktionen führen einen Python-Ausdruck aus.
- Allgemeine und Pythonaktionen sind überall sichtbar.
- Mac, Windows und Unix Aktionen sind nur sichtbar auf der entsprechenden Plattform (z.B. können Sie drei 'Bearbeiten' Aktionen definieren um einen Editor zu öffnen und die Benutzer können nur die eine 'Bearbeiten' Aktion für Ihr Betriebssystem sehen und ausführen um den Editor zu starten).

Es gibt verschiedene Beispiele in diesem Dialog. Sie können Ihn laden indem Sie auf **[Voreingestellte Aktion]** klicken. In einem Beispiel wird eine Suche auf Basis eines Attributwertes durchgeführt. Dieses Konzept wird in der folgenden Diskussion verwendet.

#### Aktionen definieren

Attributaktionen werden im Vektor *Layer Eigenschaften* Dialog definiert. Um eine Aktion zu definieren öffnen Sie den Vektor *Layer Eigenschaften* Dialog und klicken Sie auf das Menü *Aktionen*. Gehen Sie zu den *Aktion-*

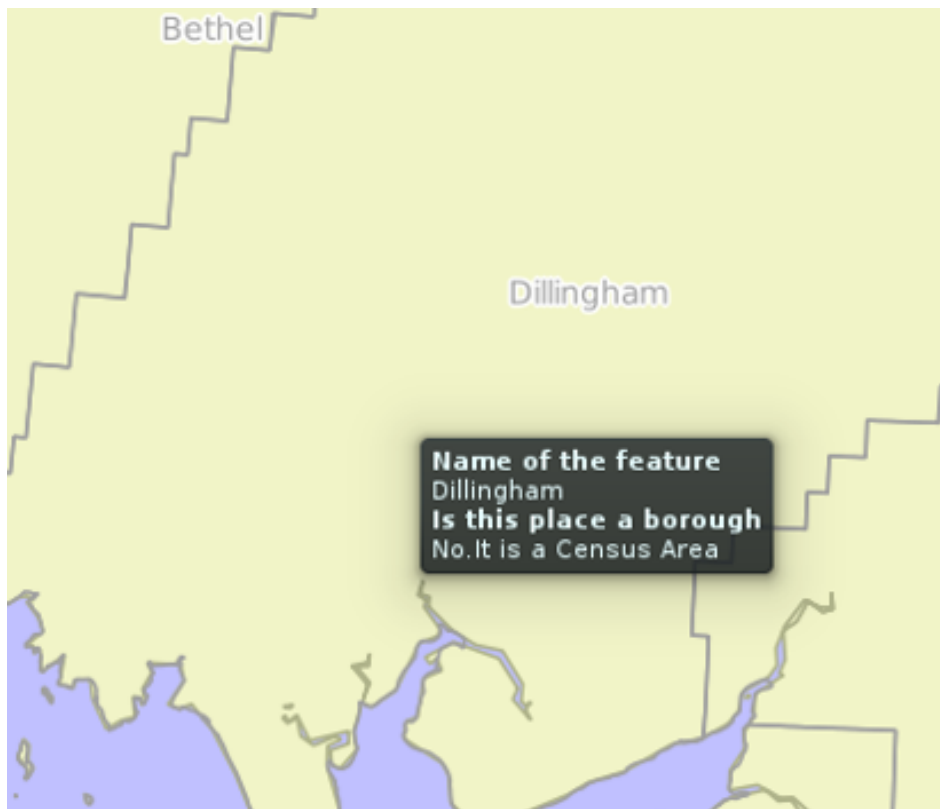



Abbildung 12.31: Kartenhinweis erstellt mit HTML-Code 

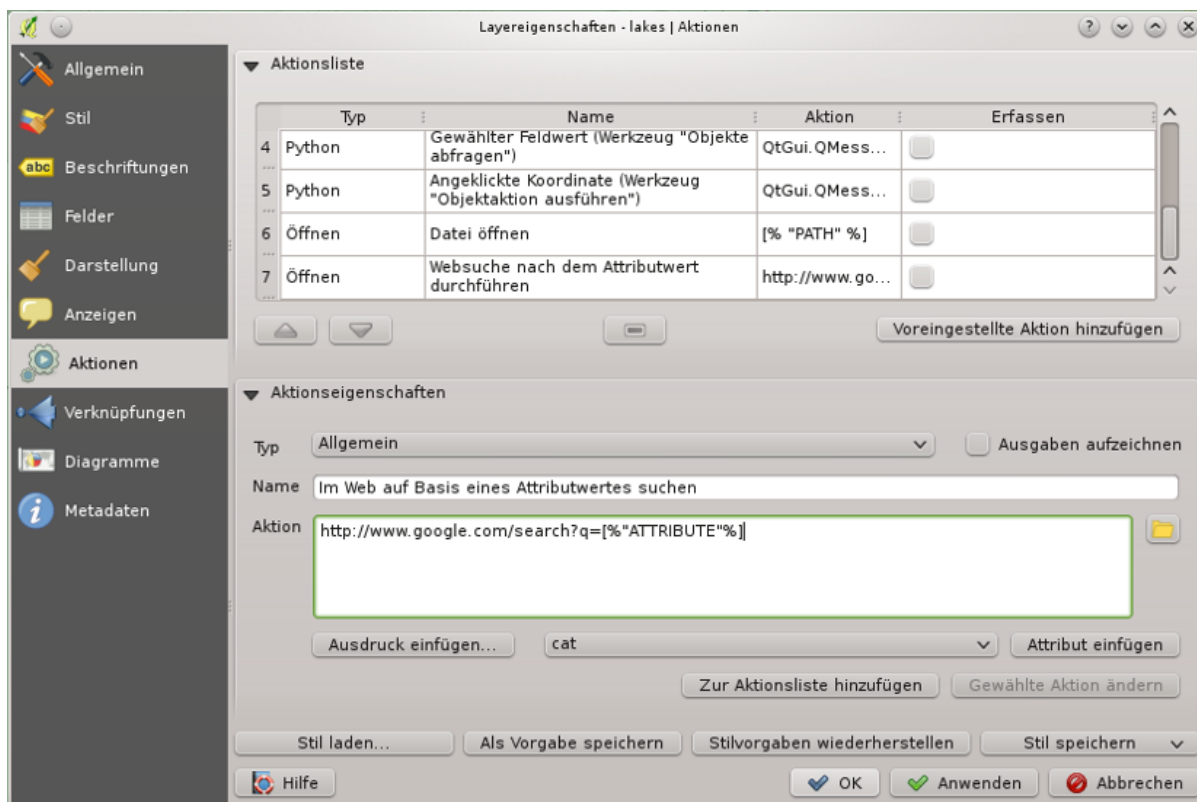


Abbildung 12.32: Überblick über den Dialog Aktionen mit einigen Beispielaktionen 

*seigenschaften*. Wählen Sie 'Allgemein' als Typ und vergeben Sie einen beschreibenden Namen für die Aktion. Die Aktion selbst muss den Namen der Anwendung, die ausgeführt wird wenn die Aktion zum Einsatz kommt, enthalten. Sie können einen oder mehrere Attributfeldwerte als Argumente der Applikation hinzufügen. Wenn die Aktion aufgerufen wird wird jeder Satz von Buchstaben, der mit einem % beginnt und auf den der Name eines Feldes folgt, durch den Wert des entsprechenden Feldes ersetzt. Die speziellen Buchstaben %% werden durch den Wert des Feldes, das durch das Objekte abfragen Werkzeug oder die Attributtabelle ausgewählt wurde, ersetzt (siehe [using\\_actions](#) unten). Anführungszeichen werden ignoriert wenn Ihnen ein Backslash vorausgeht.

Wenn Sie Feldnamen vorfinden die Substrings anderer Feldnamen sind (z.B., `col1` und `col10`) sollten Sie das angeben indem Sie den Feldnamen (und den % Buchstaben) mit eckigen Klammern umrunden (z.B. `[%col10]`). Hiermit wird vermieden dass der `%col10` Feldname mit dem, `%col1` Feldnamen mit einem 0 am Ende verwechselt wird. Die Klammern werden von QGIS entfernt wenn es im Wert des Feldes ersetzt. Wenn Sie wollen dass das zu ersetzende Feld von eckigen Klammern umrandet wird verwenden Sie ein zweites Set wie hier: `[[%col10]]`.




Wenn Sie das *Objekte abfragen* Werkzeug verwenden, können Sie den *Identifikationsergebnis* Dialog öffnen. Es enthält eine (*Abgeleitet*) Item das layertyprelevante Informationen enthält. Die Werte in diesem Element können auf ähnliche Weise den anderen Feldern zugeordnet werden indem dem Abgeleitet Feldnamen ein (*Derived*) vorangeht. Zum Beispiel hat ein Punktlayer ein X und Y Feld und die Werte dieser Felder können in der Aktion mit `%(Derived).X` und `%(Derived).Y` verwendet werden. Die Abgeleitet Attribute sind nur in der *Objekte abfragen* Dialog Box erhältlich, jedoch nicht in der *Attributtabelle* Dialogbox.

Nachfolgend werden zwei Beispielaktionen gezeigt:


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

Im ersten Beispiel wird der Webbrowser Konqueror eingebunden und öffnet eine URL. Die URL führt eine Googlesuche für den Wert des `nam` Feldes unseres Vektorlayers durch. Beachten Sie das die Anwendung oder das Skript das von der Anwendung aufgerufen wird im Pfad sein muss oder den vollen Pfad vermitteln muss. Um sicher zu sein könnten wir das erste Beispiel wie folgt umschreiben: `/opt/kde3/bin/konqueror http://www.google.com/search?q=%nam`. Dies wird versichern dass die Konqueroranwendung ausgeführt wird wenn die Aktion aufgerufen wird.

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 aus *Objekte abfragen* oder der Attributtabelle ersetzt. **Aktionen anwenden**

Aktionen können entweder über den *Objekte abfragen* Dialog, den *Attributtabelle* Dialog oder über *Objektaktion durchführen* aufgerufen werden (erinnern Sie sich daran dass diese Dialoge durch Klicken von  *Objekte abfragen* oder  *Attributtabelle öffnen* oder  *Objektaktion ausführen* geöffnet werden können). Um eine Aktion aufzurufen, klicken Sie mit der rechten Maustaste auf einen Eintrag im Popup Menü und wählen die gewünschte Aktion aus der Liste aus. Aktionen sind anhand des Namens den Sie beim Definieren der Aktion vergeben haben im Popup Menü aufgeführt. Klicken Sie auf die Aktion die sie aufrufen wollen.

Wenn Sie eine Aktion mit `%%` Notation verwenden, machen Sie einen Rechtsklick auf den Feldwert im *Objekte abfragen* Dialog oder im *Attributtabelle* Dialog den Sie der Anwendung oder dem Skript übergeben wollen.

In einem weiteren Beispiel soll gezeigt werden, wie Attributwerte eines Vektorlayers abgefragt und in eine Textdatei mit Hilfe der Bash und des `echo` Kommandos geschrieben werden (funktioniert also nur unter  und evtl. **X**). Der Abfragelayer enthält die Felder `Art taxon_name`, `Latitude lat` und `Longitude long`. Wir möchten jetzt eine räumliche Selektion von Örtlichkeiten machen und diese Feldwerte in eine Textdatei für den ausgewählten Datensatz (in der QGIS Kartenansicht in gelb gezeigt) exportieren. Hier ist die Aktion, um dies zu erreichen:

```
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 können wir eine Aktion erstellen die eine Googlesuche auf Basis des `lakes` Layers durchführt. Zuerst müssen wir die URL, die gebraucht wird um eine Suche nach einem Stichwort durchzuführen, festlegen. Dies lässt sich einfach durchführen indem man einfach Google aufruft und eine einfache Suche durchführt und dann die URL aus der Adressleiste Ihres Browsers entnimmt. Mit diesem kleinen Aufwand können wir sehen dass das Format `http://google.com/search?q=qgis` ist, wobei QGIS das Suchwort ist. Anhand dieser Informationen können wir fortfahren:

1. Laden Sie den Layer file:`lakes.shp`.
2. Öffnen Sie den *Layerigenschaften* Dialog indem Sie einen Doppelklick auf den Layer in der Legende machen und wählen Sie *Eigenschaften* aus dem Popup-Menü.
3. Klicken Sie auf das Menü *Aktionen*
4. Geben Sie einen Namen für die Aktion ein, z.B. `Google Search`.
5. Für diese Aktion ist es notwendig den Namen des externen Programms anzugeben. In diesem Fall können wir Firefox verwenden. Wenn das Programm sich nicht im Pfad befindet müssen Sie den vollständigen Pfad angeben.
6. 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=`
7. Der Text im Feld *Aktion* sollte nun folgendermaßen aussehen: `firefox http://google.com/search?q=`
8. Klicken Sie nun auf die Drop-Down Box mit dem Spaltennamen der Attributtabelle des Layers `lakes`. Der Knopf ist gleich links neben dem Knopf [**Attribut einfügen**].
9. Selektieren Sie 'NAMES' aus der Drop-Down Box und klicken Sie [**Attribut einfügen**].
10. Die Aktion sieht nun so aus:  
`firefox http://google.com/search?q=%NAMES`
11. Um die Aktion abzuschließen klicken Sie auf den [**Zur Aktionsliste hinzufügen**] Knopf.

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

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

Damit ist die Aktion fertig für den Einsatz. Schließen Sie den *Eigenschaften* Dialog und zoomen Sie in einen Bereich Ihrer Wahl. 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:

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 im *Identifikationsergebnis* Dialog angezeigt.

Sie sehen, man kann sich eine Vielzahl interessanter Aktionen ausdenken. Wenn Sie z.B. einen Punktlayer mit einzelnen Punkten haben, an denen Photos geschossen wurden, dann können Sie eine Aktion erstellen, über die Sie dann das entsprechende Foto anzeigen lassen können, wenn Sie auf den Punkt in der Karte klicken. Man kann 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.

Wir können auch komplexere Beispiele erstellen, indem wir z.B. **Python** Aktionen verwenden.

Normalerweise wenn wir beim Erstellen von Aktionen zum Öffnen einer Datei mit einer externen Anwendung absolute Pfade, oder letztendlich relative Pfade verwenden ist im zweiten Fall der Pfad relativ zum Ort der aus-



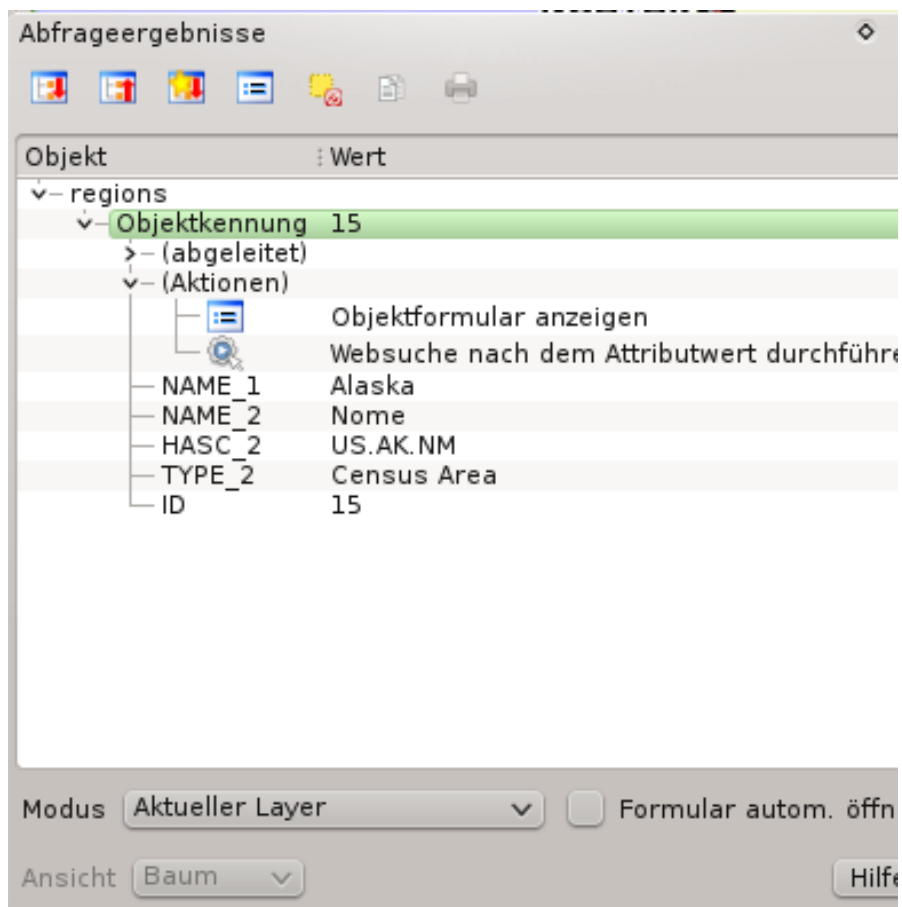


Abbildung 12.33: Wählen Sie ein Objekt und eine Aktion aus 🐧

fühbaren Datei. Was aber wenn wir relative Pfade, die relativ zum ausgewählten Layer (eine dateibasierte, wie ein Shape oder SpatialLite) sind, benutzen müssen ? Mit dem folgenden Code können wir einen Trick anwenden:

```
command = "firefox";
imagerelpath = "images_test/test_image.jpg";
layer = qgis.utils iface.activeLayer();
import os.path;
layerpath = layer.source() if layer.providerType() == 'ogr'
    else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None);
path = os.path.dirname(str(layerpath));
image = os.path.join(path, imagerelpath);
import subprocess;
subprocess.Popen( [command, image ] );
```

Wir müssen uns nur ins Gedächtnis rufen dass es sich um eine *Python* Aktion handelt und dass das Ändern der *command* und *imagerelpath* Variablen auf unsere Bedürfnisse angepasst wird.

Was aber wenn der relative Pfad relativ zur (gespeicherten) Projektdatei sein muss? Der Code der Python Aktion würde wie folgt lauten:

```
command="firefox";
imagerelpath="images/test_image.jpg";
projectpath=qgis.core.QgsProject.instance().fileName();
import os.path; path=os.path.dirname(str(projectpath)) if projectpath != '' else None;
image=os.path.join(path, imagerelpath);
import subprocess;
subprocess.Popen( [command, image ] );
```

Ein anderes Python Aktion Beispiel ist das mit wir dem Projekt neue Layer hinzufügen können. Z.B. wird in den folgenden Beispielen dem Projekt ein Vektorlayer beziehungsweise ein Rasterlayer hinzugefügt. Die Namen der Dateien, die dem Projekt hinzugefügt werden sollen, und die Namen, die den Layern gegeben werden, sind datengesteuert (*filename* und *layername* sind Spaltennamen der Attributtabelle des Vektorlayers in dem die Aktion erstellt wurde):



```
qgis.utils iface.addVectorLayer('/yourpath/[% "filename" %].shp', '[% "layername" %]',
    'ogr')
```

Um eine Rasterdatei hinzuzufügen (ein TIF-Bild in diesem Beispiel) wird daraus:

```
qgis.utils iface.addRasterLayer('/yourpath/[% "filename" %].tif', '[% "layername" %]
')
```


### 12.3.8 Menü Verknüpfungen



Mit dem *Verknüpfungen* Menü können Sie eine geladene Attributtabelle mit einem geladenen Vektorlayer verknüpfen. Nach dem Klicken von  öffnet sich der *Vektorverknüpfung hinzufügen* Dialog. Als Schlüsselspalten müssen Sie einen Joinlayer definieren, den Sie mit dem Zielvektorlayer verbinden wollen. Dann müssen Sie das Verknüpfungsfeld, das der Joinlayer und der Zielvektorlayer gemeinsam haben, festlegen. Jetzt können Sie auch eine Untermenge von Feldern aus dem verknüpften Layer auf Basis des Kontrollkästchens  *Verknüpfte Felder wählen* festlegen. Als Ergebnis der Verknüpfung werden alle Informationen des Joinlayers und des Zielvektorlayers in der Attributtabelle des Zielvektorlayers als verknüpfte Information dargestellt. wenn Sie eine Untermenge von Feldern festgelegt haben dann werden nur diese Felder in der Attributtabelle des Zielvektorlayers dargestellt.

QGIS bietet zur Zeit Unterstützung für das Verknüpfen von nicht-räumlichen Tabellenformaten die von OGR unterstützt werden (z.B. CSV, DBF und Excel) und von Delimited Text und für den PostgreSQL Provider (siehe [figure\\_joins\\_1](#)).

Zusätzlich können Sie mit dem *Vektorverknüpfung hinzufügen* Dialog:

-  *Verknüpfung im Speicher cachen*

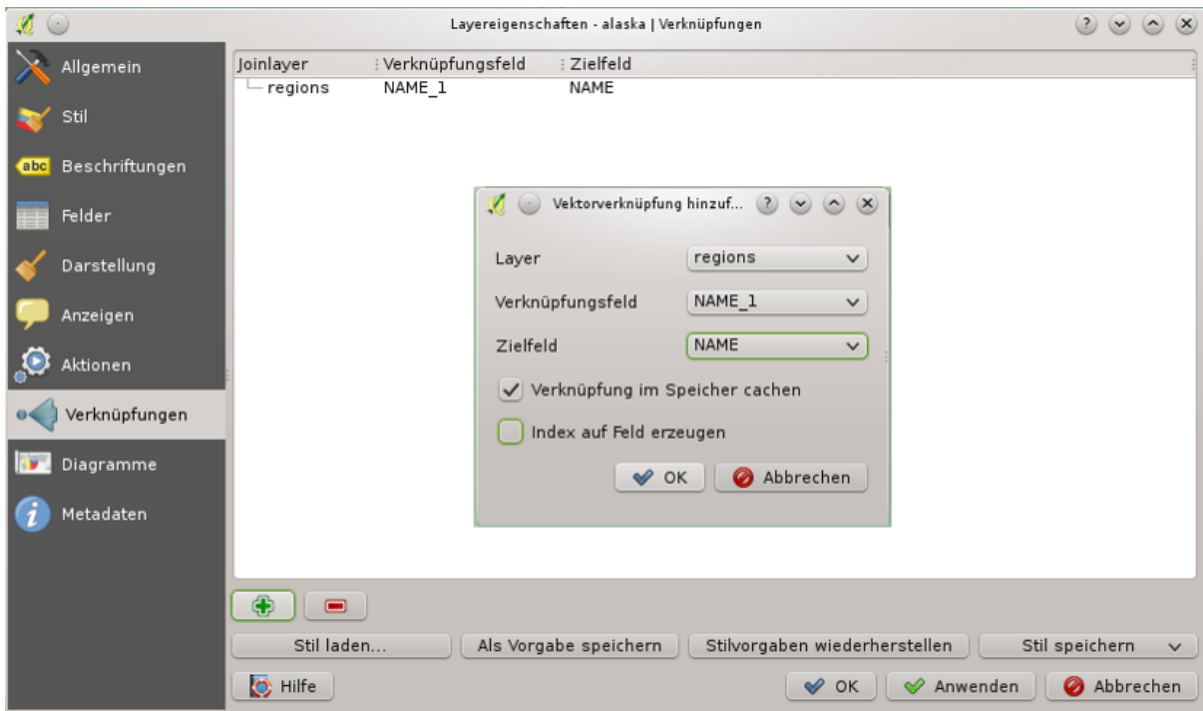



Abbildung 12.34: Verknüpfe eine Attributtabelle zu einem vorhandenen Vektorlayer 

- *Index auf Feld erzeugen*

### 12.3.9 Menü Diagramme



 Das Menü *Diagramme* ermöglicht es, ein Diagramm als Grafik über einen Vektorlayer zu visualisieren (siehe [figure\\_diagrams\\_1](#)).

Die aktuelle Kernimplementierung von Diagrammen bietet Unterstützung von Kuchendiagrammen, Textdiagrammen und Histogrammen.

Das Menü ist in vier Reiter aufgeteilt: *Darstellung*, *Größe*, *Position* und *Optionen*.

Im Fall des Textdiagramms und Kuchendiagramms werden Textwerte aus verschiedenen Datenspalten untereinander mit einem Kreis oder einer Box mit Teilern dargestellt. Im *Größe* Reiter basiert die Diagrammgröße auf einer festen Größe oder auf linearer Skalierung entsprechend eines Klassifikationsattributs. Die Platzierung der Diagramme, die im *Position* Reiter vorgenommen wird, interagiert mit dem neuen Beschriften, also werden Positionskonflikte zwischen Diagrammen und Beschriftungen aufgedeckt und gelöst. Zusätzlich können die Diagramme manuell befestigt werden.

Wir werden ein Beispiel zeigen und dem Alaskagrenzlayer ein Textdiagramm das Temperaturdaten von einem *climate* Vektorlayer zeigt überlagern. Beide Vektorlayer sind Teil des QGIS Beispieldatensatzes (siehe Abschnitt [Beispieldaten](#)).

1. Klicken Sie erst auf das  *Vektorlayer hinzufügen* Icon, browsen Sie zum QGIS Beispieldatensatzordner und laden Sie die beiden Vektorlayer *alaska.shp* und *climate.shp*.
2. Doppelklicken Sie auf den *climate* Layer in der Kartenlegende um den Dialog *Layer Eigenschaften* zu öffnen.
3. Klicken Sie auf das Menü *Diagramme*, aktivieren Sie  *Diagramme anzeigen* und wählen Sie 'Textdiagramm' aus der *Diagrammtyp*  Kombo-Box aus.

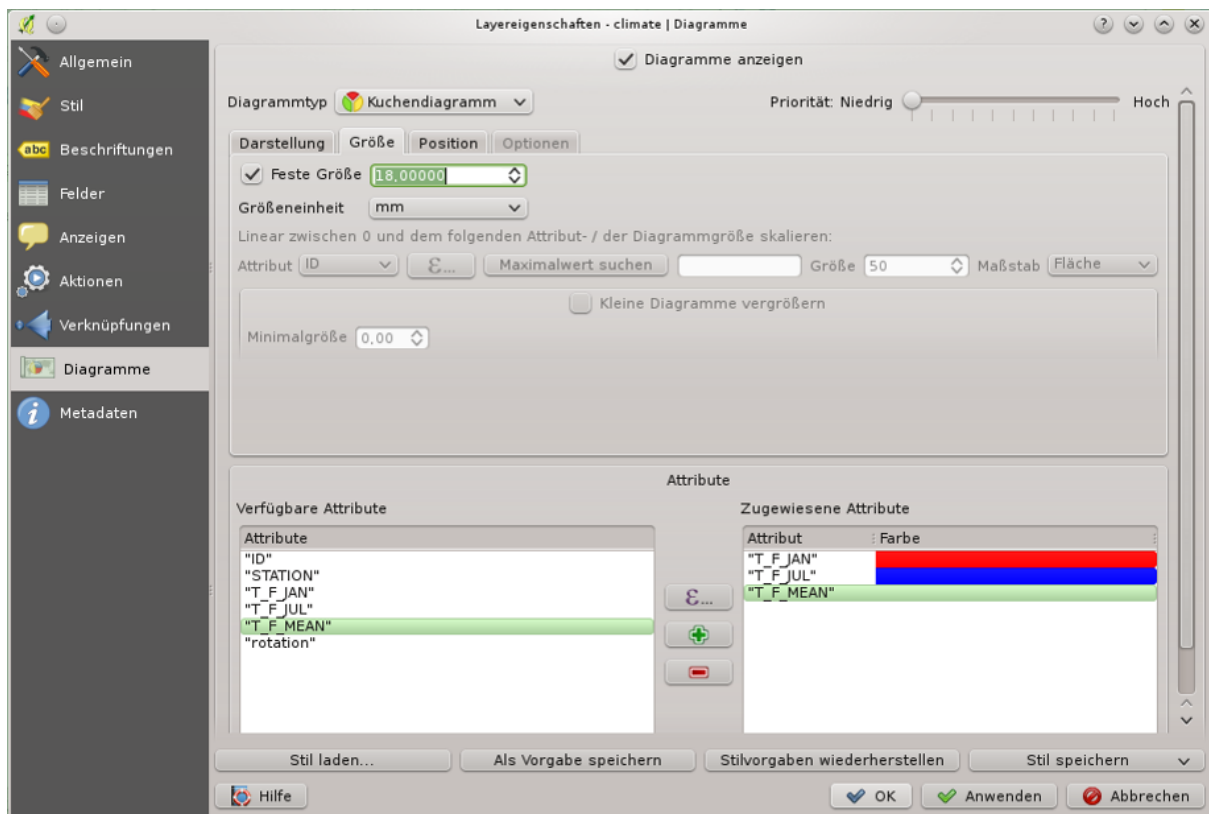



Abbildung 12.35: Vektorlayer Eigenschaften Dialog mit Menü Diagramme 

4. Im *Darstellung* Reiter wählen wir ein Hellblau als Hintergrundfarbe und im Reiter *Größe* stellen wir eine feste Größe von 18 mm ein.
5. Im Reiter *Position* könnte die Platzierung auf 'Um Punkt' eingestellt werden.
6. Im Diagramm wollen wir die Werte der drei Spalten T\_F\_JAN, T\_F\_JUL und T\_F\_MEAN darstellen. Wählen Sie erst T\_F\_JAN als *Attribute* und klicken Sie den `|mActionSignPlus|` Knopf, dann 'T\_F\_JUL' und schließlich T\_F\_MEAN.
7. Klicken Sie jetzt **[Anwenden]** um das Diagramm in QGIS anzuzeigen.
8. Sie können die Diagrammgröße im *Größe* Reiter anpassen. Deaktivieren Sie  *Feste Größe* und stellen Sie die Größe des Diagramms auf Basis eines Attributes mit dem **[Maximalwert suchen]** Knopf und dem *Größe* Menü ein. Wenn die Diagramme auf dem Bildschirm zu klein erscheinen können Sie das  *Kleine Diagramme vergrößern* Kontrollkästchen aktivieren und die Minimalgröße des Diagramms definieren.
9. Verändern Sie die Attributfarben indem Sie auf die Farbwerte im *Zugewiesene Attribute* Feld doppelklicken. [Figure\\_diagrams\\_2](#) zeigt eine Vorstellung von dem Ergebnis.
10. Klicken Sie schließlich auf **[Ok]**.

Behalten Sie im Hinterkopf dass im Reiter *Position* eine *Datendefinierte Position* der Diagramme möglich ist. Sie können hier Attribute verwenden um die Position des Diagramms zu definieren. Sie können auch eine maßstababhängige Sichtbarkeit im *Darstellung* Reiter einstellen.

Die Größe und die Attribute können auch ein Ausdruck sein. Verwenden Sie den `ε...` Knopf um einen Ausdruck einzufügen. Siehe Kapitel [Ausdrücke](#) für weitere Informationen und Beispiele.

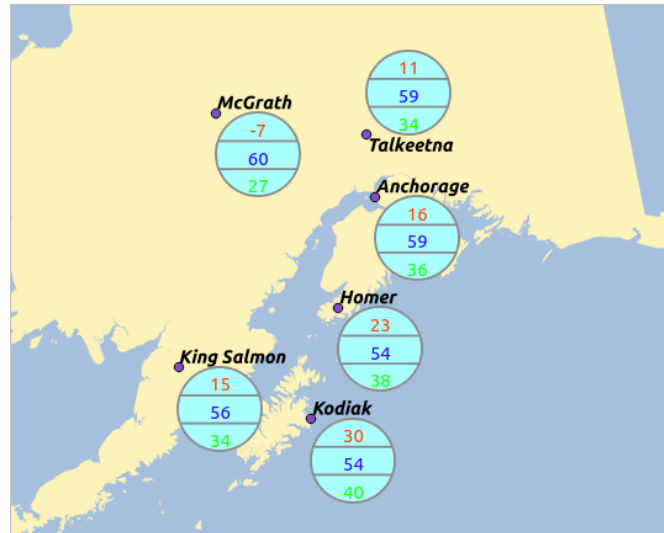



Abbildung 12.36: Diagramm aus Temperaturdaten auf einer Karte dargestellt 

### 12.3.10 Menü Metadaten




Das *Metadaten* Menü besteht aus *Beschreibung*, *Beschreibung*, *Metadaten-URL* und *:guilabel: 'Eigenschaften* Abschnitten.

Im Abschnitt *Eigenschaften* erhalten Sie allgemeine Informationen über den Layer, darunter Einzelheiten über den Typ und die Verortung, Anzahl der Objekte, Objekttyp und Bearbeitungsmöglichkeiten. Die *Ausdehnung* Tabelle zeigt Ihnen Layerausdehnungsinformationen und das *Räumliches Bezugssystem des Layers* Informationen über das CRS des Layers. Dies ist ein schneller Weg Informationen über den Layer herauszufinden.

Zusätzlich können Sie einen Titel und eine Zusammenfassung für den Layer im Abschnitt *Beschreibung* hinzufügen oder bearbeiten. Es ist hier außerdem möglich eine *:guilabel: Stichwortliste* zu definieren. Diese Stichwortlisten können in einem Metadatenkatalog verwendet werden. Wenn Sie einen Titel aus einer XML Metadatendatei verwenden wollen müssen Sie einen Link im *DateURL* Feld ausfüllen. Verwenden Sie *Beschreibung* um Attributdaten aus einem XML-Metadatenkatalog zu erhalten. In *Metadaten-URL* können Sie den allgemeinen Pfad zum XML-Metadatenkatalog definieren. Diese Information wird in der QGIS Projektdatei für nachfolgende Sitzungen gespeichert und wird für QGIS Server verwendet.

## 12.4 Ausdrücke

Die **Ausdrücke** Funktion sind durch den Feldrechner oder den Neue Spalte hinzufügen Knopf in der Attribut-tabelle oder im Feld Reiter in den Layereigenschaften erreichbar; über die abgestufte, kategorisierte und regelbasierte Darstellung im Stil Reiter der Layereigenschaften; über das ausdrucksbasierte Beschriften  in der **Beschriftung** Kernanwendung; über die Objektauswahl und über den Diagramm Reiter der Layereigenschaften.

Es gibt mächtige Werkzeuge um Attributwerte zu ändern um den Endwert dynamisch zu ändern um den Geometriestil zu ändern, den Inhalt der Beschriftung, den Wert für ein Diagramm zu ändern, ein Objekt auszuwählen oder eine virtuelle Spalte zu erstellen.

### 12.4.1 Funktionsliste

Die **Funktionsliste** enthält Funktionen genauso wie Felder und Werte. Sehen Sie sich die Hilfefunktionen im **Hilfe zu gewählten Funktionen** Bereich an. In **Ausdruck** sehen Sie die Berechnungsausdrücke, die Sie mit der

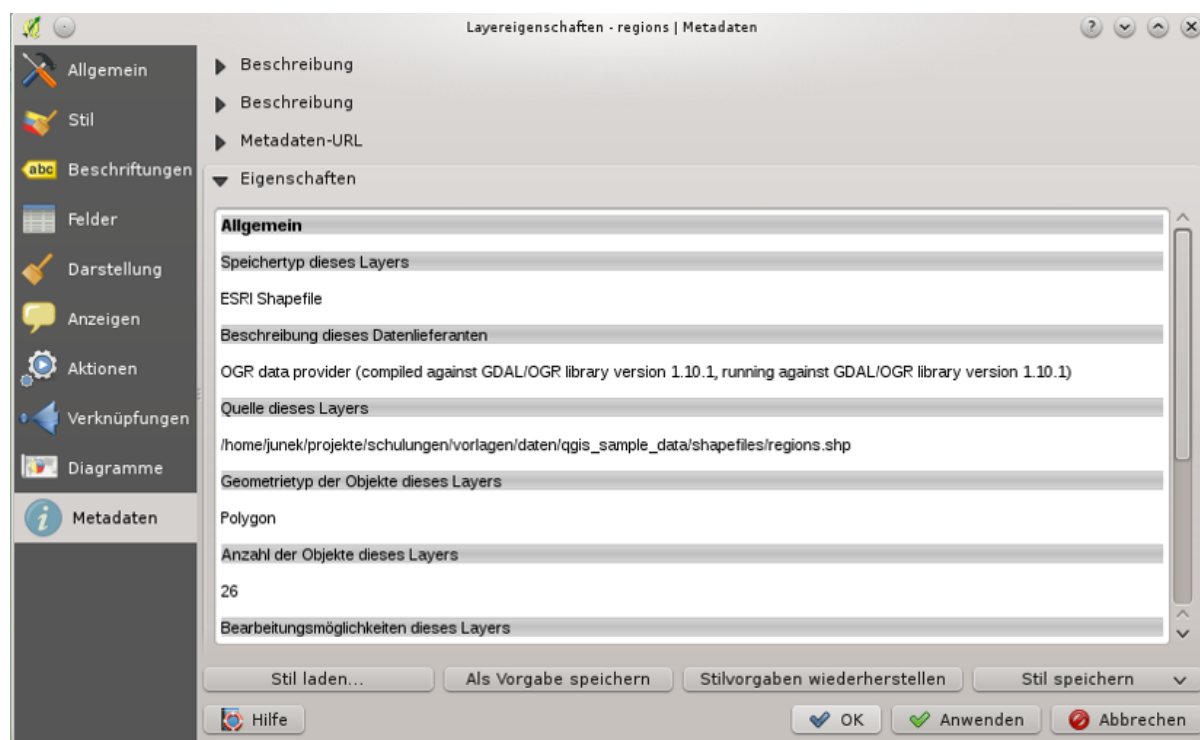



Abbildung 12.37: Menü Metadaten im Vektorlayer Eigenschaften Dialog 

**Funktionsliste** erstellt haben. Für die gebräuchlichsten Operatoren siehe **Operatoren**.

Klicken Sie in der **Funktionsliste** auf *Felder und Werte* um alle durchsuchbaren Attribute der Attributtabelle anschauen zu können. Um ein Attribut dem Feldrechner **Ausdruck** Feld hinzuzufügen doppelklicken Sie seinen Namen in der *Felder und Werte* Liste. Im Allgemeinen können Sie die diversen Felder, Werte und Funktionen benutzen um den Berechnungsausdruck aufzubauen, oder Sie geben in einfach direkt ins Fenster ein. Um die Werte eines Feldes darzustellen müssen Sie einfach einen Rechtsklick auf das entsprechende Feld machen. Sie können zwischen *10 Stichproben* und *Alle eindeutigen* wählen. Auf der rechten Seite öffnet sich die **Feldwerte** Liste mit den eindeutigen Werten. Um dem Feldrechner **Ausdruck** Fenster einen Wert hinzuzufügen, machen Sie einen Doppelklick auf den Namen in der **Feldwerte** Liste.

Die Gruppen *Operatoren*, *Mathematik*, *Umwandlungen*, *Zeichenketten*, *Geometrie* und *Datensatz* stellen zahlreiche Funktionen zur Verfügung. In *Operatoren* finden Sie mathematische Operatoren. Suchen Sie in *Mathematik* nach mathematischen Funktionen. Die *Umwandlungen* Gruppe enthält Funktionen die einen Datentyp in einen anderen konvertieren. Die *Zeichenkette* Gruppe stellt Funktionen für Datenketten zur Verfügung. In der *Geometrie* Gruppe finden Sie Funktionen für Geometrieobjekte. Mit den Funktionen der *Datensatz* Gruppe können Sie Ihren Datensatz mit einer Nummerierung versehen. Um eine Funktion in die **Ausdruck** Box des Feldrechners hinzuzufügen klicken Sie auf > und doppelklicken Sie dann die Funktion.

## Operatoren

Diese Gruppe enthält Operatoren (z.B. +,-,\*).

a + b	a plus b
a - b	a minus b
a * b	a multiplied by b
a / b	a divided by b
a % b	a modulo b (for example, 7 % 2 = 1, or 2 fits into 7 three times with remainder 1)
a ^ b	a power b (for example, 2^2=4 or 2^3=8)
a = b	a and b are equal
a > b	a is larger than b
a < b	a is smaller than b

a <> b	a and b are not equal
a != b	a and b are not equal
a <= b	a is less than or equal to b
a >= b	a is larger than or equal to b
a ~ b	a matches the regular expression b
+ a	positive sign
- a	negative value of a
	joins two values together into a string 'Hello'    ' world'
LIKE	returns 1 if the string matches the supplied pattern
ILIKE	returns 1 if the string matches case-insensitive the supplied pattern (ILIKE can be used instead of LIKE to make the match case-insensitive)
IS	returns 1 if a is the same as b
OR	returns 1 when condition a or b is true
AND	returns 1 when condition a and b are true
NOT	returns 1 if a is not the same as b
column name "column name"	value of the field column name, take care to not be confused with simple quote, see below
'string'	a string value, take care to not be confused with double quote, see above
NULL	null value
a IS NULL	a has no value
a IS NOT NULL	a has a value
a IN (value[,value])	a is below the values listed
a NOT IN (value[,value])	a is not below the values listed

### Einige Beispiele:

- Verbindet eine Zeichenkette und einen Wert von einem Spaltennamen:

```
'My feature's id is: ' || "gid"
```

- Testen Sie ob das "description" Attributfeld mit einer 'Hello' Zeichenfolge im Wert startet (beachten Sie die Position des % Zeichens):

```
"description" LIKE 'Hello%'
```

### Bedingungen

Diese Gruppe enthält Funktionen um bedingte Prüfungen in Ausdrücken zu handhaben.

CASE	evaluates multiple expressions and returns a result
CASE ELSE	evaluates multiple expressions and returns a result
coalesce	returns the first non-NULL value from the expression list
regexp_match	returns true if any part of a string matches the supplied regular expression

### Einige Beispiele:

- Sende einen Wert zurück wenn die erste Bedingung wahr ist, sonst einen anderen Wert:

```
CASE WHEN "software" LIKE '%QGIS%' THEN 'QGIS' ELSE 'Other'
```

### Mathematische Funktionen

Diese Gruppe enthält mathematische Funktionen (z.B. sqrt, sin und cos).

sqrt(a)	square root of a
abs	returns the absolute value of a number
sin(a)	sine of a
cos(a)	cosine of a
tan(a)	tangent of a
asin(a)	arcsin of a
acos(a)	arccos of a
atan(a)	arctan of a
atan2(y,x)	arctan of y/x using the signs of the two arguments to determine the quadrant of the result
exp	exponential of a value
ln	value of the natural logarithm of the passed expression
log10	value of the base 10 logarithm of the passed expression
log	value of the logarithm of the passed value and base
round	round to number of decimal places
rand	random integer within the range specified by the minimum and maximum argument (inclusive)
randf	random float within the range specified by the minimum and maximum argument (inclusive)
max	largest value in a set of values
min	smallest value in a set of values
clamp	restricts an input value to a specified range
scale_linear	transforms a given value from an input domain to an output range using linear interpolation
scale_exp	transforms a given value from an input domain to an output range using an exponential curve
floor	rounds a number downwards
ceil	rounds a number upwards
\$pi	pi as value for calculations

## Umwandlungen

Diese Gruppe enthält Funktionen, um einen Datentypen in einen anderen umzuwandeln (z.B. Zeichenketten zu Ganzzahlen oder umgekehrt).

toint	converts a string to integer number
toreal	converts a string to real number
tostring	converts number to string
todatetime	converts a string into Qt data time type
todate	converts a string into Qt data type
totime	converts a string into Qt time type
tointerval	converts a string to an interval type (can be used to take days, hours, months, etc. off a date)

## Datum und Zeit Funktionen

Diese Gruppe enthält Funktionen die auf Datums- und Zeitdaten angewendet werden können.

\$now	current date and time
age	difference between two dates



year	extract the year part from a date, or the number of years from an interval
month	extract the month part from a date, or the number of months from an interval
week	extract the week number from a date, or the number of weeks from an interval
day	extract the day from a date, or the number of days from an interval
hour	extract the hour from a datetime or time, or the number of hours from an interval
minute	extract the minute from a datetime or time, or the number of minutes from an interval
second	extract the second from a datetime or time, or the number of minutes from an interval

### Einige Beispiele:

- Lassen Sie sich den Monat und das Jahr von heute im Format "10/2014" herausgeben:

```
month($now) || '/' || year($now)
```

### Zeichenkettenfunktionen

Diese Gruppe enthält Funktionen für Zeichenketten (z.B. Ersetzen und in Großbuchstaben umwandeln).

lower	convert string a to lower case
upper	convert string a to upper case
title	converts all words of a string to title case (all words lower case with leading capital letter)
trim	removes all leading and trailing white space (spaces, tabs, etc.) from a string
wordwrap	returns a string wrapped to a maximum/minimum number of characters
length	length of string a
replace	returns a string with the supplied string replaced
regexp_replace(a,this,that)	returns a string with the supplied regular expression replaced
regexp_substr	returns the portion of a string which matches a supplied regular expression
substr(*a*,from,len)	returns a part of a string
concat	concatenates several strings to one
strpos	returns the index of a regular expression in a string
left	returns a substring that contains the n leftmost characters of the string
right	returns a substring that contains the n rightmost characters of the string
rpad	returns a string with supplied width padded using the fill character
lpad	returns a string with supplied width padded using the fill character
format	formats a string using supplied arguments
format_number	returns a number formatted with the locale separator for thousands (also truncates the number to the number of supplied places)
format_date	formats a date type or string into a custom string format

## Farbfunktionen

Diese Gruppe enthält Funktionen zur Farbmanipulation.

color_rgb	returns a string representation of a color based on its red, green, and blue components
color_rgba	returns a string representation of a color based on its red, green, blue, and alpha (transparency) components
ramp_color	returns a string representing a color from a color ramp
color_hsl	returns a string representation of a color based on its hue, saturation, and lightness attributes
color_hsla	returns a string representation of a color based on its hue, saturation, lightness and alpha (transparency) attributes
color_hsv	returns a string representation of a color based on its hue, saturation, and value attributes
color_hsva	returns a string representation of a color based on its hue, saturation, value and alpha (transparency) attributes
color_cmyk	returns a string representation of a color based on its cyan, magenta, yellow and black components
color_cmyka	returns a string representation of a color based on its cyan, magenta, yellow, black and alpha (transparency) components

## Geometriefunktionen

Dies Gruppe enthält Funktionen für das Arbeiten mit Geometrieobjekten (z.B. Länge und Flächeninhalt).

\$geometry	returns the geometry of the current feature (can be used for processing with other functions)
\$area	returns the area size of the current feature
\$length	returns the length size of the current feature
\$perimeter	returns the perimeter length of the current feature
\$x	returns the x coordinate of the current feature
\$y	returns the y coordinate of the current feature
xat	retrieves the nth x coordinate of the current feature. n given as a parameter of the function
yat	retrieves the nth y coordinate of the current feature. n given as a parameter of the function
xmin	returns the minimum x coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
xmax	returns the maximum x coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
ymin	returns the minimum y coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
ymax	returns the maximum y coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
geomFromWKT	returns a geometry created from a well-known text (WKT) representation
geomFromGML	returns a geometry from a GML representation of geometry
bbox	
disjoint	returns 1 if the geometries do not share any space together
intersects	returns 1 if the geometries spatially intersect (share any portion of space) and 0 if they don't
touches	returns 1 if the geometries have at least one point in common, but their interiors do not intersect
crosses	returns 1 if the supplied geometries have some, but not

	all, interior points in common
contains	returns true if and only if no points of b lie in the exterior of a, and at least one point of the interior of b lies in the interior of a
overlaps	returns 1 if the geometries share space, are of the same dimension, but are not completely contained by each other
within	returns 1 if geometry a is completely inside geometry b
buffer	returns a geometry that represents all points whose distance from this geometry is less than or equal to distance
centroid	returns the geometric center of a geometry
bounds	returns a geometry which represents the bounding box of an input geometry. Calculations are in the Spatial Reference System of this Geometry.
bounds_width	returns the width of the bounding box of a geometry. Calculations are in the Spatial Reference System of this Geometry.
bounds_height	returns the height of the bounding box of a geometry. Calculations are in the Spatial Reference System of this Geometry.
convexHull	returns the convex hull of a geometry (this represents the minimum convex geometry that encloses all geometries within the set)
difference	returns a geometry that represents that part of geometry a that does not intersect with geometry b
distance	returns the minimum distance (based on spatial ref) between two geometries in projected units
intersection	returns a geometry that represents the shared portion of geometry a and geometry b
symDifference	returns a geometry that represents the portions of a and b that do not intersect
combine	returns the combination of geometry a and geometry b
union	returns a geometry that represents the point set union of the geometries
geomToWKT	returns the well-known text (WKT) representation of the geometry without SRID metadata

## Datensatzfunktionen

Diese Gruppe enthält Funktionen die sich auf datensatzbezeichner beziehen.

\$rownum	returns the number of the current row
\$id	returns the feature id of the current row
\$currentfeature	returns the current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature.
\$scale	returns the current scale of the map canvas
\$uuid	generates a Universally Unique Identifier (UUID) for each row. Each UUID is 38 characters long.
getFeature	returns the first feature of a layer matching a given attribute value.
attribute	returns the value of a specified attribute from a feature.
\$map	returns the id of the current map item if the map is being drawn in a composition, or "canvas" if the map is being drawn within the main QGIS window.

### Felder und Werte

Enthält eine Liste von Feldern vom Layer. Auf Beispielwerte kann über einen Rechtsklick zugegriffen werden.

Wählen Sie einen Feldnamen aus der Liste, machen Sie dann einen Rechtsklick, um auf ein Kontextmenü mit Optionen zum Laden von Beispielwerten aus dem gewählten Feld zuzugreifen.

Feldnamen sollten in doppelte Anführungsstriche gesetzt werden. Werte oder Zeichenketten sollten in einfache Anführungsstriche gesetzt werden.

.

## 12.5 Editierfunktionen

QGIS unterstützt vielfältige Möglichkeiten OGR, SpatiaLite, PostGIS, MSSQL Spatial und Oracle Spatial Vektorlayer und -tabellen zu editieren.

---

**Bemerkung:** Die Vorgehensweise GRASS Layer zu bearbeiten ist anders - siehe Abschnitt *Digitalisieren und Editieren eines GRASS Vektorlayers* für Details.

---

### Tipp: Zeitgleiches Editieren

Diese Version von QGIS kontrolliert nicht, ob noch jemand ein Objekt zur gleichen Zeit editiert wie Sie. Die zuletzt schreibende Person gewinnt.




---

### 12.5.1 Einstellen der Fangtoleranz und des Suchradius

Bevor wir damit beginnen können Stützpunkte zu editieren ist es sehr wichtig die Fangtoleranz und den Suchradius für Stützpunkte festzulegen um Vektorlayergeometrien optimal editieren zu können.

#### Fangtoleranz

Die Fangtoleranz ist der Abstand den QGIS verwendet um den nächstgelegenen Stützpunkt bzw. das nächstgelegene Liniensegment zu suchen mit dem sie versuchen sich zu verbinden wenn sie einen neuen Stützpunkt setzen wollen oder einen bestehenden Stützpunkt verschieben wollen. Wenn Sie sich nicht innerhalb der definierten Fangtoleranz befinden wird QGIS den Stützpunkt dort lassen wo sich der Mauszeiger gerade befindet anstatt zu einem bestehenden Stützpunkt oder Liniensegment zu snappen. Die Fangtoleranz Einstellungen betreffen alle Werkzeuge die mit Toleranz arbeiten.

1. Eine globale, projektweite Fangtoleranz kann definiert werden indem Sie *Einstellungen* →  *Optionen* auswählen. Gehen Sie unter Mac zu *QGIS* →  *Einstellungen ...*. Unter Linux: *Bearbeiten* →  *Optionen*. Im Menü *Digitalisierung* können Sie zwischen 'zum Stützpunkt', 'zum Segment' oder 'Zum Stützpunkt und Segment' als Standard Fangmodus wählen. Sie können auch eine Voreingestellte Fangtoleranz und einen Suchradius für die Stützpunktbearbeitung definieren. Die Toleranz kann entweder in Karteneinheiten oder in Pixeln eingestellt werden. Der Vorteil wenn man Pixel wählt ist dass die Fangtoleranz sich nicht nach Zoomoperationen verändert. In unserem kleinen Digitalisierprojekt (Arbeiten mit dem Alaskadatensatz) definieren wir die Fangeinheiten in Fuß. Ihre Ergebnisse können variieren etwas in der Größenordnung von 300 ft bei einem Maßstab von 1:10000 sollte eine hinreichende Einstellung sein.
2. Eine layerbezogene Fangtoleranz kann unter *Einstellungen* → (oder *Datei* →) *Fangoptionen ...* um den Fangmodus und die Fangtoleranz auf Layerbasis zu ermöglichen und anzupassen definiert werden (siehe [figure\\_edit\\_1](#)).

Beachten Sie dass dieses layerbasierte Fangen die globalen Fangoptionen im Menü Digitalisieren überschreibt. Also wenn Sie einen Layer bearbeiten wollen und dass seine Stützpunkte zu einem anderen Layer snappen sollen, dann aktivieren Sie das Fangen nur auf den zu snappenden Layer und verringern Sie die globale Fangtoleranz auf einen kleineren Wert. Darüberhinaus wird das Fangen nicht mit einem Layer funktionieren der nicht im


Fangoptionen Dialog aktiviert ist, was unabhängig von der globalen Fangtoleranz funktioniert. Vergewissern Sie sich also dass das Kontrollkästchen für die Layer auf die Sie snappen wollen aktiviert ist.



Abbildung 12.38: Das Bearbeiten von Snappingoptionen auf Layerbasis 

## Suchradius

Der Suchradius ist der Abstand den QGIS verwendet um nach dem nächsten Stützpunkt den Sie versuchen zu verschieben wenn Sie auf die Karte klicken zu suchen. Wenn Sie sich nicht innerhalb des Suchradius befinden wird QGIS keinen Stützpunkt für das Bearbeiten finden und auswählen und dementsprechend wird eine lästige Warnung auftauchen. Die Fangtoleranz und der Suchradius werden in Karteneinheiten oder Pixeln eingestellt so dass Sie vielleicht experimentieren müssen um Sie passend einzustellen. Wenn Sie eine zu große Toleranz angeben snappt QGIS zum falschen Stützpunkt, besonders wenn Sie es mit einer großen Anzahl von Stützpunkten in kurzer Distanz zu tun haben. Stellen Sie den Suchradius zu klein ein wird nichts zum Verschieben gefunden.


Der Suchradius für die Stützpunktbearbeitung in Layereinheiten kann im Menü *Digitalisierung* unter *Einstellungen* →  *Optionen* definiert werden. An der gleichen Stellen wo Sie die allgemeine, projektweite Fangtoleranz definieren.

## 12.5.2 Zoomen und Karte verschieben

Bevor Sie einen Layer editieren sollten Sie in den Bereich hineinzoomen, den Sie bearbeiten wollen. Dadurch vermeiden Sie lange Wartezeiten, bis alle Stützpunkte des Layers visualisiert werden.

Neben den  Karte verschieben und  Hineinzoomen /  Hinauszoomen Icons in der Werkzeugleiste können Sie auch mit dem Mausrad, der Leertaste und den Pfeiltasten navigieren.

### Mit dem Mausrad im Kartenfenster zoomen und verschieben

Während des Digitalisierens können Sie das Mausrad zum Verschieben innerhalb des Hauptfensters drücken und Sie können das Mausrad zum hinein- und hinauszoomen drehen. Zum Zoomen platzieren Sie den Mauscursor in die Karte und rollen es vorwärts (von Ihnen weg) um hineinzuzoomen und rückwärts (zu Ihnen hin) um herauszuzoomen. Die Position des Mausursors wird das Zentrum des Sie interessierenden Bereichs sein. Sie können das Verhalten des Mausrad-Zooms anhand des Reiters *Kartenwerkzeuge* unter *Einstellungen* →  *Optionen* anpassen.

## Den Layer mit den Pfeiltasten verschieben

Das Verschieben der Karte während des Digitalisierens ist mit den Pfeiltasten möglich. Platzieren Sie die Maus in das Kartenfenster und klicken Sie auf die rechts-Pfeiltaste um nach Osten zu verschieben, auf die links-Pfeiltaste um nach Westen zu verschieben, auf die Aufwärts-Pfeiltaste um nach Norden zu verschieben und die Abwärts-Pfeiltaste um nach Süden zu verschieben.

Sie können auch die Leertaste verwenden um vorübergehend Mausbewegungen die die Karte verschieben auszulösen. Die `Bild hoch` und `Bild runter` Tasten auf Ihrer Tastatur bewirken bei der Kartenanzeige hinein- oder hinauszuzoomen ohne Ihre Digitalisiersitzung zu unterbrechen.

### 12.5.3 Topologisches Editieren

Abgesehen von der Einstellung des layerbasierten Fangmodus können im Menü *Einstellungen* → *Fangoptionen...* (oder *Datei* → *Fangoptionen*) auch topologische Funktionen aktiviert werden. Hier können Sie  *Topologische Bearbeitung an* definieren und/oder für Polygonlayer können Sie die Spalte  *Überschn. verm.* aktivieren, die eine Überschneidung neuer Polygone vermeidet.

#### Topologisches Editieren ermöglichen

Die Option  *Topologische Bearbeitung an* dient dem Bearbeiten und Aufrechterhalten von gemeinsamen Grenzen in Polygonmosaiken. QGIS 'erkennt' gemeinsame Grenzen in Polygonmosaiken so dass Sie einfach den Stützpunkt einmal verschieben müssen und QGIS dann die Aufgabe übernimmt die andere Grenze zu updaten.

#### Überschneidung neuer Polygone vermeiden

Die zweite topologische Option in der  *Überschn. verm.* Spalte, *Überschneidungen von neuen Polygonen vermeiden*, verhindert ein Überlappen in Polygonmosaiken. Es dient dazu aneinanderhängende Polygone schneller zu digitalisieren. Wenn Sie bereits ein Polygon erstellt haben ist es mit dieser Option möglich das zweite zu digitalisieren, so dass beide sich überschneiden und QGIS schneidet dann das zweite Polygon gemäß der gemeinsamen Grenze aus. Der Vorteil ist dass Sie nicht alle Stützpunkte der gemeinsamen Grenze digitalisieren müssen.

#### Fang auf Schnittpunkte aktivieren

Ein andere Möglichkeit stellt das  *Fang auf Schnittpunkte aktivieren* dar. Sie können damit auf einen Schnittpunkt von Hintergrundlayern snappen selbst wenn es keinen Stützpunkt auf dem Schnittpunkt gibt.

### 12.5.4 Einen vorhandenen Layer editieren

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

Im Allgemeinen werden die Werkzeuge zum Bearbeiten von Vektorlayern in eine Digitalisierung und eine Erweiterte Digitalisierung Werkzeugleiste aufgeteilt, wie in Abschnitt *Erweiterte Digitalisierung* beschrieben. Sie können beides unter *Einstellungen* → *Werkzeuge* → auswählen oder abwählen. Wenn Sie die grundlegenden Digitalisierertools verwenden können Sie die folgenden Funktionen ausführen:














Icon	Funktion	Icon	Funktion
	Aktuelle Änderungen		Bearbeitungsstatus umschalten
	Objekt hinzufügen: Punkt hinzufügen		Objekt hinzufügen: Linie hinzufügen
	Objekt hinzufügen: Polygon hinzufügen		Objekt(e) verschieben
	Knotenwerkzeug		Ausgewähltes Löschen
	Ausgewählte Objekte ausschneiden		Objekte kopieren
	Objekte einfügen		Layeränderungen speichern

Tabelle Bearbeiten: Funktionen der Werkzeugleiste Digitalisierung




Alle Editiersitzungen fangen mit dem Anwählen der  Bearbeitungsstatus umschalten Option an. Sie können dies im Kontextmenu nach dem Rechtsklicken auf den Legendeneintrag für einen vorhandenen Layer finden.

Alternativ können Sie den Bearbeitungsstatus umschalten Knopf aus der Digitalisierung Werkzeugleiste verwenden um den Bearbeitungsstatus ein- oder auszuschalten. Sobald der Layer im Bearbeitungsstatus ist, erscheinen die Stützpunkte als das von Ihnen gewählte Stützpunktsymbol und zusätzlich werden die Digitalisier-Tools in der Werkzeugleiste aktiv.

**Tipp: Regelmäßiges Sichern der Daten**

Denken Sie daran  Layeränderungen regelmäßig zu speichern. Dies überprüft auch dass Ihre Datenquelle alle Änderungen akzeptiert.


**Objekte digitalisieren**

Danach können Sie mit den Icons  Objekt hinzufügen,  Objekt hinzufügen oder  Objekt hinzufügen 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.


Das Attribute Fenster wird erscheinen in dem Sie Informationen für das neue Objekt eingeben können. [Figure\\_edit\\_2](#) zeigt die Eingabe von Attributen für einen fiktiven neuen Fluss in Alaska. In dem Menü *Digitalisierung* unter *Einstellungen* → *Optionen* können Sie auch  *Attributeingabeformular bei der Objekteerstellung unterdrücken* und  *Letzte Attributwerteingaben wiederverwenden* aktivieren.

Mit dem  Objekt(e) verschieben Icon in der Werkzeugleiste können Sie bestehende Objekte verschieben.

**Tipp: Datentyp der Attribute**

Beim Editieren werden die Attributtypen während der Dateneingabe überprüft. Deswegen ist es nicht möglich eine Zahl in eine Textspalte im Dialog *Attribute* oder andersherum einzugeben. Wenn Sie das tun müssen sollten Sie die Attribute in einem zweiten Schritt innerhalb des *Attributtabelle* Dialogs bearbeiten.

**Aktuelle Änderungen**

Mit dieser neuen Funktion können Sie mehrere Layer digitalisieren. Wählen Sie  *Layeränderungen speichern* um alle Änderungen, die Sie in mehreren Layern gemacht haben, zu speichern. Sie haben auch die Möglichkeit



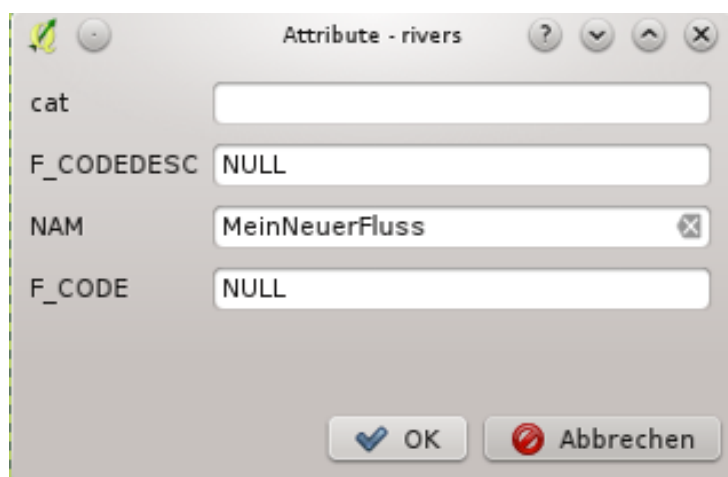







Abbildung 12.39: Der Attributwertedialog nach dem Digitalisieren eines neuen Vektorobjekts 

 *Verwerfen für gewählte Layer* zu benutzen so dass die Digitalisierung für alle selektierten Layer rückgängig gemacht werden kann. Wenn Sie das Bearbeiten der selektierten Layer beenden wollen kann man das einfach mit der Funktion  *Abbruch für gewählte Layer* erreichen.


Die gleichen Funktionen sind für das Bearbeiten aller Layer des Projektes zugänglich.

## Knotenwerkzeug


Für Shapedatei basierte Layer genauso wie für SpatialLite, PostgreSQL/Post GIS, MSSQL Spatial und Oracle Spatial Tabellen stellt das  *Knotenwerkzeug* Bearbeitungsmöglichkeiten für Stützpunkte ähnlich wie CAD Programme zur Verfügung. Es ist möglich einfach mehrere Stützpunkte auf einmal auszuwählen und diese zu verschieben, hinzuzufügen und alle zusammen zu löschen. Das Knotenwerkzeug funktioniert auch mit angeschalteter ‘Spontan’ Reprojektion und unterstützt die topologische Bearbeitungsfunktion. Dieses Werkzeug ist, anders als andere Werkzeuge in QGIS, nachhaltig, so dass wenn eine Operation durchgeführt wird die Auswahl für dieses Objekt und Werkzeug aktiv bleibt. Wenn das Knotenwerkzeug nicht in der Lage ist Objekte zu finden wird eine Warnung gezeigt.

Es ist wichtig die Eigenschaften unter menuselection: *Einstellungen* →  *Optionen* → *Digitalisierung* → *Suchradius*:  auf eine Zahl größer als Null (z.B. 10) einzustellen. Andernfalls ist QGIS nicht in der Lage mitzuteilen welcher Stützpunkt bearbeitet werden soll.

### Tipp: Stützpunktmarken

Die aktuelle Version von QGIS unterstützt drei Arten von Stützpunktmarkern: ‘Teiltransparenter Kreis’, ‘Kreuz’ und ‘Keine’. Um den Markierungsstil zu ändern wählen Sie  *Optionen* aus dem *Einstellungen* Menü, klicken Sie auf das *Digitalisierung* Menü und wählen Sie den entsprechenden Eintrag.



## Eine einfache Übung

Beginnen Sie, indem Sie das  *Knotenwerkzeug* aktivieren und dann ein Objekt selektieren. Rote Boxen erscheinen an jedem Stützpunkt des Objektes. Nun sind folgende Funktionen vorhanden.

- **Stützpunkte auswählen:** Sie können Stützpunkte auswählen indem Sie nacheinander darauf klicken, indem Sie auf eine Ecke klicken um die Stützpunkte an beiden Ecken auszuwählen oder indem Sie ein Rechteck um einige Stützpunkte klicken und ziehen. Wenn ein Stützpunkt ausgewählt ist ändert sich die Farbe zu blau. Um der aktuellen Auswahl mehr Stützpunkte hinzuzufügen halten Sie die `Strg` Taste beim Klicken gedrückt. Halten Sie `Strg` oder `Umschalt` -Taste gedrückt wenn Sie Klicken um den Auswahlstatus der




Stützpunkte umzuschalten (Stützpunkte die gerade nicht selektiert sind werden wie immer ausgewählt, aber auch Stützpunkte die schon ausgewählt sind werden deselektiert).

- **Stützpunkte hinzufügen:** Um einen Stützpunkt hinzuzufügen machen Sie einfach einen Doppelklick in der Nähe einer Ecke und ein neuer Stützpunkt erscheint auf der Ecke in der Nähe des Mauszeigers; deswegen muss er wenn nötig noch verschoben werden.
- **Stützpunkte löschen:** Wählen Sie dazu Stützpunkte aus, die gelöscht werden sollen und klicken Sie dann auf die Taste **Entfernen**. Beachten Sie dass das  **Knotenwerkzeug** nicht für das Löschen eines vollständigen Objekts benutzen können, QGIS stellt sicher dass die minimal erforderliche Anzahl von Stützpunkten für das Objekt an dem Sie arbeiten erhalten bleibt. Um ein vollständiges Objekt zu löschen müssen Sie das  **Ausgewähltes löschen Werkzeug** benutzen.
- **Stützpunkte verschieben:** Wählen Sie alle Stützpunkte aus, die verschoben werden sollen. Alle ausgewählten Stützpunkte werden in dieselbe Richtung wie der Mauspfel verschoben. Wenn eine Fangtoleranz eingestellt ist, können die Stützpunkte zu dem nächstgelegenen Stützpunkt oder Segment snappen.

Jede Veränderung die mit dem Knotenwerkzeug gemacht wurde ist als separater Eintrag im Rückgängig Dialog eingetragen. Beachten Sie dass alle Operationen topologisches Bearbeiten unterstützen wenn dies eingeschaltet ist. Spontanreprojektion wird auch unterstützt und das Knotenwerkzeug erstellt Tooltips um einen Stützpunkt ausfindig zu machen wenn man mit dem Mauszeiger darübergeht.




## Objekte ausschneiden, kopieren und einfügen

Ausgewählte Objekte können ausgeschnitten, kopiert und an andere Ebenen im aktuellen QGIS-Projekt übergeben (eingefügt) werden, wenn sich der Ziellayer auch im Editiermodus befindet, indem Sie nach Auswahl des Ziellayers auf den Knopf  **Bearbeitungsstatus umschalten** klicken.

Objekte können an externe Anwendungen als Text abgelegt werden. Dabei werden die Objekte im CSV-Format dargestellt und die Geometriedaten im OGC Well-Know Text (WKT) Format.

In dieser Version von QGIS können jedoch Textobjekte von außen nicht in einen Layer innerhalb von QGIS eingefügt werden. Wann würde dann die Kopieren und Einfügen Funktion zum Einsatz kommen? Wie sich herausstellt kann man mehr als einen Layer zur gleichen Zeit bearbeiten und Objekte zwischen Layern Kopieren/Einfügen. Warum würden wir das tun wollen? Sagen wir wir müssen an einem Layer arbeiten, brauchen davon aber nur ein- bis zwei Seen, nicht die gesamten 5.000 aus unserem `big_lakes` Layer. Wir können einen neuen Layer erstellen und Kopieren/Einfügen verwenden um die erforderlichen Seen hineinzukopieren.

Als Beispiel werden wir einige Seen in einen neuen Layer kopieren:

1. Laden Sie den Layer, von dem Sie einige Objekte kopieren wollen (Quelle)
2. Laden oder erstellen Sie einen Layer, in den die kopierten Objekte eingefügt werden sollen (Ziel)
3. Schalten Sie für den Ziel Layer den Bearbeitungsstatus ein
4. Stellen Sie die Quelle aktiv, indem Sie es in der Legende anklicken
5. Benutzen Sie das  **Einzelnes Objekt auswählen Werkzeug** 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 Bearbeitungsstatus für beide Layer und speichern Sie das Ergebnis ab



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.



### **Tipp: Deckungsgleichheit eingefügter Objekte**

Wenn Ihre Quell- und Ziellayer die gleiche Projektion verwenden dann haben die eingefügten Objekte die identische Geometrie wie der Quelllayer. Wenn der Ziellayer jedoch eine andere Projektion hat dann kann QGIS nicht garantieren dass die Geometrie identisch ist. Dies ist einfach aus dem Grund so, dass sich kleine Rundungsfehler ergeben wenn zwischen Projektionen konvertiert wird.



---

### **Ausgewählte Objekte löschen**

Wenn Sie ein ganzes Polygon löschen möchten, selektieren Sie es erst mit dem  Einzelnes Objekt auswählen Werkzeug. Sie können dabei auch mehrere Objekte gleichzeitig auswählen. Wenn Sie die Objekte ausgewählt haben, klicken Sie auf das Icon .

Das Werkzeug  Ausgewählte Objekte ausschneiden 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 ein Fehler unterlaufen ist, wieder rückgängig machen, indem wir auf das Werkzeug  Objekte einfügen drücken. Ausschneiden, kopieren und übergeben von Objekten funktioniert mit den gerade ausgewählten Objekten und können nach Bedarf kombiniert verwendet werden.

### **Änderungen speichern**

Wenn ein Layer im Bearbeitungsmodus ist behält QGIS alle Änderungen im Speicher. Aus diesem Grund werden diese nicht umgehend der Datenquelle oder -platte übermittelt. Wenn Sie Bearbeitungen in dem aktuellen Layer speichern wollen aber mit dem Bearbeiten fortfahren wollen ohne den Bearbeitungsmodus verlassen zu wollen können Sie den  Layeränderungen speichern Knopf klicken. Wenn Sie den Editiermodus mit  Bearbeitungsstatus umschalten ausschalten wollen werden Sie ebenfalls gefragt ob Sie Ihre Änderungen speichern oder verwerfen wollen.

Wenn die Änderungen nicht gespeichert werden können (z.B. weil die Festplatte voll ist oder Attribute Werte aufweisen, die außerhalb der Wertespanne liegen), bleiben die Änderungen erstmal im QGIS Arbeitsspeicher. Dies ermöglicht es, Änderungen vorzunehmen und dann nochmals die Daten zu speichern.

---

### **Tipp: Datenintegrität**

Es ist immer gut ein Backup von Ihren Daten zu machen bevor Sie mit dem Bearbeiten starten. Während die Autoren von QGIS sich bemühen die Integrität Ihrer Daten zu bewahren bieten wir keine Garantie in dieser Hinsicht.

---

## 12.5.5 Erweiterte Digitalisierung

Icon	Funktion	Icon	Funktion
	Rückgängig		Wiederholen
	Objekt(e) drehen		Objekt vereinfachen
	Ring hinzufügen		Teil hinzufügen
	Ring füllen		Ring löschen
	Teil löschen		Objekte überarbeiten
	Linie versetzen		Objekte trennen
	Teile zerlegen		Gewählte Objekte verschmelzen
	Attribute gewählter Objekte vereinen		Punktsymbole drehen

Tabelle Erweiterte Digitalisierung: Werkzeugleiste Erweiterte Digitalisierung für Vektorlayer

### Rückgängig und Wiederholen

Mit dem Rückgängig und Wiederholen Werkzeug können Sie Vektorbearbeitungsoperationen rückgängig machen oder wiederholen. Es gibt auch ein andockbares Bedienfeld das alle Operationen in der Rücknahme/Wiederholung History zeigt (siehe [Figure\\_edit\\_3](#)). Dieses Bedienfeld wird nicht standardmäßig angezeigt; es kann angezeigt werden indem man auf die Werkzeugleiste klickt und das Rücknahme/Wiederholung Kontrollkästchen aktiviert. Rücknahme/Wiederholung ist auch dann noch aktiv wenn das Bedienfeld nicht angezeigt wird.

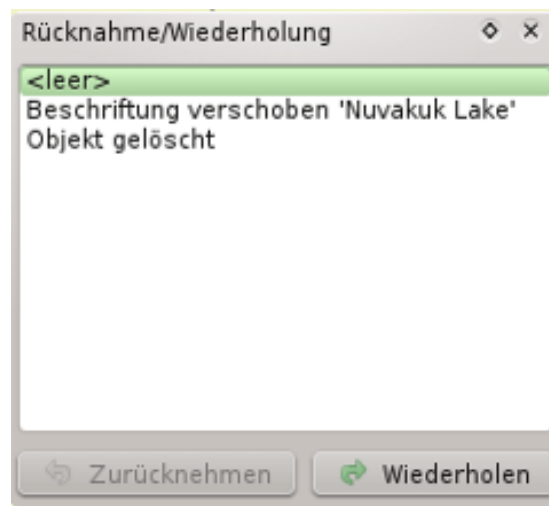





Abbildung 12.40: Rückgängig und Wiederholen von Digitalisierschritten

Wenn Rückgängig gedrückt wird wird der Status aller Objekte und Attribute auf den Status bevor die zurückgenommene Aktion durchgeführt wurde zurückgesetzt. Änderungen die keine normalen Vektorbearbeitungsoperationen sind (z.B. Änderungen die durch ein Plugin durchgeführt werden) können oder können nicht rückgängig gemacht werden. Dies hängt davon ab wie die Änderungen durchgeführt werden.


Wenn Sie Rücknahme/Wiederholung verwenden wollen klicken Sie einfach auf eine Operation in der History; alle Objekte werden dann auf den Stand vor der ausgewählten Operation zurückgesetzt.

## Objekt(e) drehen


Verwenden Sie  Objekt(e) drehen um ein oder mehrere ausgewählte Objekte in der Kartenansicht zu drehen. Sie müssen erst die Objekte auswählen und dann das  Objekt(e) drehen Icon drücken. Der Zentroid des/der Objekte(s) erscheint und wird zum Drehungsankerpunkt. Wenn Sie mehrere Objekte auswählen liegt der Drehungsankerpunkt im gemeinsamen Zentrum der Objekte. Drücken und ziehen Sie den linke Maustaste in die gewünschte Richtung um die ausgewählten Objekte zu drehen.

Es ist auch möglich einen benutzerdefinierten Drehungsankerpunkt, um den dann die ausgewählten Objekte rotieren, zu erstellen. Wählen Sie die Objekte, die gedreht werden sollen und aktivieren Sie das  Objekt(e) drehen Werkzeug. Drücken und halten Sie den `Strg` Knopf und bewegen Sie den Mauszeiger (ohne die Maustaste zu drücken) zu der Stelle wohin Sie den Rotationsanker verschieben wollen. Lassen Sie den `Strg` Knopf los wenn der gewünschte Rotationsankerpunkt erreicht ist. Jetzt können Sie die Linke Maustaste in die gewünschte Richtung drücken und ziehen um das/die ausgewählte(n) Objekt(e) zu drehen.


## Objekt vereinfachen

Das Werkzeug  Objekt vereinfachen ermöglicht es, die Anzahl der Stützpunkte eines Objektes zu verringern, solange sich die Geometrie nicht verändert und der Geometrietyp nicht eine Mehrfachgeometrie ist. Sie müssen dazu eines oder mehrere Objekte auswählen. Dadurch werden diese von einem roten Gummiband umrandet und ein Schieber erscheint. Wenn Sie die Stellung des Schiebers verändern, können Sie sehen, wie das Objekt vereinfacht wird. Durch einen Klick auf **[OK]** wird die neue Geometrie abgespeichert. Wenn ein Objekt nicht vereinfacht werden kann (z.B. bei MultiPolygonen), erscheint ein Fenster mit einer entsprechenden Meldung.




## Ring hinzufügen

Sie können Ring-Polygone mit dem Werkzeug  Ring hinzufügen erstellen. Das bedeutet, dass Sie innerhalb eines bestehenden Polygons weitere Polygone digitalisieren können. Diese erscheinen dann als 'Loch', so dass nur der Zwischenraum des inneren und des äußeren Polygons als Ring-Polygon übrig bleibt.


## Teil hinzufügen

Sie können mit  Teil hinzufügen einem ausgewählten multipolygon Polygone hinzufügen. Das neue Teilpolygon muss außerhalb des ausgewählten Multipolygons digitalisiert werden.


## Ring füllen

Sie können die  Ring füllen Funktion verwenden um gleichzeitig einen Ring in ein Polygon einzufügen und dem Layer ein neues Objekt zuzufügen. So brauchen Sie nicht erst das  Ring hinzufügen Icon und dann die  Objekt hinzufügen Funktion benutzen.


## Ring löschen

Das Werkzeug  Ring löschen ermöglicht es, Ringpolygone innerhalb eines existierenden Fläche zu löschen. Das Werkzeug funktioniert nur mit Polygonlayern und kann für Polygon- und Multipolygon-Objekte benutzt werden. Es findet keine Veränderung statt, wenn es auf den äußeren Ring eines Polygons angewendet wird. Bevor Sie die Stützpunkte des Ringes selektieren, stellen Sie bitte die Fangtoleranz entsprechend ein.

## Teil löschen

Mit dem  Teil löschen Werkzeug können Teile eines Multi-Feature Objektes gelöscht werden (z.B. ein Polygon von einem Multipolygon Objekt löschen). Dabei wird der letzte Teil des Multi-Feature Objektes nicht gelöscht. Das Werkzeug funktioniert mit allen Multi-Feature Geometrien Punkte, Linien und Polygone. Bevor Sie die Stützpunkte des Teils selektieren, stellen Sie bitte die Fangtoleranz entsprechend ein.

## Objekte überarbeiten

Die Geometrien von Linien und Polygonen können mit dem Werkzeug  Objekt überarbeiten verändert werden. Dabei ersetzt die gezogene Linie die originale Linie oder Flächenlinie eines Objektes von der ersten bis zur zweiten Überschneidung. Für Polygone kann dies manchmal zu unerwünschten Resultaten führen und ist vorwiegend sinnvoll, wenn man nur kleinere Teile einer Fläche ersetzen möchte und nicht für eine vollständige Überarbeitung. Dabei ist es für die Vereinfachungslinie auch nicht erlaubt, mehrere Polygongrenzen zu überschneiden.


Als Beispiel können Sie die Grenze eines Polygons mit diesem Tool bearbeiten. Klicken Sie als erstes in die Fläche des Polygons neben den Punkt wo Sie den neuen Stützpunkt hinzufügen wollen. Dann überqueren Sie die Grenze und fügen sie Stützpunkte außerhalb des Polygons ein. Machen Sie einen Rechtsklick in die Fläche des Polygons zum Vervollständigen. Das Werkzeug wird automatisch einen Knoten an der Stelle wo die neue Linie die Grenze kreuzt hinzufügen. Es ist auch möglich einen Teil der Polygonfläche zu entfernen indem man außerhalb des Polygons anfängt, Stützpunkte innerhalb des Polygons hinzufügt und außerhalb des Polygons mit einem Rechtsklick aufhört.



---

**Bemerkung:** Das Objekte überarbeiten Werkzeug kann die Startposition eines Polygonringes oder einer geschlossenen Linie verändern. Der Punkt, der zweimal abgebildet ist wird also nicht mehr der gleiche sein. Dies mag kein Problem für die meisten Anwendungen sein, sollte aber beachtet werden.

---


## Linie versetzen

Das  Linie versetzen Werkzeug erstellt parallele Linien von Linienlayern. Das Werkzeug kann auf den bearbeiteten Layer (die Geometrien werden verändert) und auch auf Hintergrundlayer (in diesem Fall erstellt es Kopien von Linien/ Ringen und fügt Sie dem bearbeiteten Layer hinzu) angewendet werden. Es ist auf diese Weise ideal geeignet Abstandslinienlayer zu erstellen. Der Versatz wird unten links in der Taskleiste angezeigt.


Um einen Versatz eines Linienlayers zu erstellen müssen Sie erst in den Bearbeitungsmodus gehen und dann das Objekt auswählen. Sie können das  Linie versetzen Werkzeug aktiv machen und das Kreuz in den gewünschten Abstand ziehen. Ihre Änderungen können dann mit dem  Layeränderungen speichern Werkzeug gespeichert werden.

Der QGIS Optionen Dialog (Digitalisierung Reiter dann **Werkzeug zum Linien versetzen** Abschnitt) ermöglicht es Ihnen einige Parameter wie **Verbindungsstil**, **Quadrantsegmente**, **Eckengrenze** zu konfigurieren.


## Objekte trennen

Objekte können mit dem Werkzeug  Objekt teilen geteilt werden. Zeichnen Sie dazu einfach eine Linie durch das Objekt, das Sie teilen wollen.



## Teile zerlegen

In QGIS 2.0 ist es jetzt möglich die Teile eines Multi-Part Features zu zerlegen so dass die Anzahl der Teile sich erhöht. Zeichnen Sie einfach eine Linie über den Teil den Sie zerlegen wollen indem Sie dafür das  Teile zerlegen Icon verwenden.


## Gewählte Objekte verschmelzen

Mit dem  **Gewählte Objekte verschmelzen** Werkzeug können Sie Objekte verschmelzen, die gemeinsame Grenzen haben. Mit einem neuen Dialog können Sie auswählen welcher Wert zwischen jedem ausgewählten Objekt gewählt wird oder eine Funktion auswählen (Minimum, Maximum, Mittel, Summe, Attribut überspringen), die für jede Spalte benutzt werden soll, auswählen.

## Attribute gewählter Objekte vereinen

Mit dem  **Attribute gewählter Objekte vereinen** Werkzeug können Sie Attribute von Objekten vereinen die gemeinsame Grenzen und Attribute haben ohne Ihre Grenzen zu vereinen. Wählen Sie zuerst mehrere Objekte auf einmal aus. Drücken Sie dann den  **Attribute gewählter Objekte** Knopf. Jetzt fragt Sie QGIS welche Attribute auf alle ausgewählten Objekte angewendet werden sollen. Im Ergebnis haben alle ausgewählten Objekte die gleichen Attributeinträge.

## Punktsymbole drehen

Mit  **Punktsymbole drehen** können Sie die Rotation von Punktsymbolen in der Kartenansicht verändern. Sie müssen erst ein Drehungsfeld in der Attributtabelle des Punktlayers im *Erweitert* Menü des *Stil* Menü der *Layereigenschaften* definieren. Sie werden auch nach 'SVG Markierung' gehen müssen und *Datendefinierte Eigenschaften ...* wählen. Aktivieren Sie  *Winkel* und wählen Sie 'Drehung' als Feld. Ohne diese Einstellungen ist das Werkzeug inaktiv.

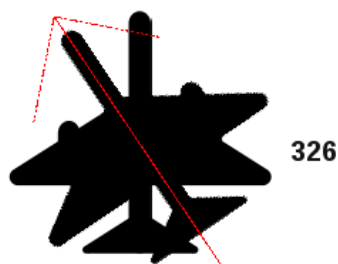



Abbildung 12.41: Drehen von Punktsymbolen 

Um die Drehung zu ändern, wählen Sie einen Punkt im Kartenfenster aus und drehen diesen, indem Sie die linke Maustaste gedrückt halten. Ein roter Pfeil mit dem Drehwinkel wird dann angezeigt (siehe [Figure\\_edit\\_4](#)). Wenn Sie die linke Maustaste wieder loslassen, wird der neue Wert in der Attributtabelle aktualisiert.

---


**Bemerkung:** Wenn Sie zusätzlich die `Strg`-Taste gedrückt halten, findet die Drehung in 15 Grad Schritten statt.

---

## 12.5.6 Neue Vektorlayer erstellen


Mit QGIS können Sie neue Shapedateilayer, neue SpatialLite Layer und neue GPX Layer erstellen. Die Erstellung eines neuen GRASS Layers wird durch das GRASS Plugin unterstützt. Bitte gehen Sie zu Abschnitt [Einen neuen GRASS Vektorlayer erstellen](#) für mehr Informationen über das Erstellen von GRASS Vektorlayern.


## Eine neue Shapedatei erstellen

Um einen neuen Shapelayer zum Bearbeiten zu erstellen wählen Sie *Neu* →  *Neuer Shapedateilayer ...* aus dem *Layer* Menü. Der *Neuer Vektorlayer* Dialog wird dargestellt wie in [Figure\\_edit\\_5](#) gezeigt. Wählen Sie den Layertyp (Punkt, Linie oder Polygon) und das KBS (Koordinatenbezugssystem).

Beachten Sie dass QGIS noch nicht die Erstellung von 2.5D Objekten unterstützt (d.h. Objekte mit X,Y,Z Koordinaten).




Abbildung 12.42: Neuer Vektorlayer Dialog 


Um die Erstellung des neuen Shapedateilayer zu vervollständigen fügen Sie die gewünschten Attribute ein indem Sie den **[Der Attributliste hinzufügen]** Knopf drücken und einen Namen und einen Typ für das Attribut festlegen. Eine erste 'id' Spalte wird standardmäßig hinzugefügt, kann aber, falls nicht gewünscht, wieder gelöscht werden. Es werden nur *Typ:Text* , *Typ: Ganzzahl*, *Typ: Dezimalzahl* und *Typ: Datum* Attribute unterstützt. Zusätzlich und gemäß des Attributtyps können Sie auch die Breite und Präzision der neuen Attributspalte definieren. Sind Sie einmal zufrieden mit den Attributen, klicken Sie **[OK]** und vergeben Sie einen Namen für die Shapedatei. QGIS wird dem Namen automatisch eine `.shp` Erweiterung hinzufügen. Ist der Layer einmal erstellt wird dieser der



Karte hinzugefügt und Sie können ihn auf die gleiche Weise wie in Abschnitt *Einen vorhandenen Layer editieren* bearbeiten.

### Einen neuen SpatialLite Layer erstellen


Um einen neuen SpatialLite Layer zum Bearbeiten zu erstellen wählen Sie *Neu* →  *SpatialLite Layer erstellen ...* aus dem *Layer* Menü. Der *Neuer SpatialLite Layer* Dialog wird dargestellt wie in *Figure\_edit\_6* beschrieben.

Der erste Schritt ist eine bestehende SpatialLite Datenbank auszuwählen oder eine neue SpatialLite Datenbank zu erstellen. Dies kann mit dem Suchen Knopf  auf der rechten Seite des Datenbankfeldes durchgeführt werden. Fügen Sie dann einen Namen für den neuen Layer ein, definieren Sie den Layertyp und legen Sie das Koordinatenbezugssystem mit **[KBS angeben]** fest. Falls gewünscht können Sie einen  *automatisch inkrementierenden Primärschlüssel erzeugen*.

Um eine Attributtabelle für den neuen SpatialLite Layer zu definieren fügen Sie die Namen der Attributspalten, die Sie mit dem entsprechenden Spaltentyp erstellen wollen, ein und klicken Sie den **[Der Attributliste hinzufügen]** Knopf. Sind Sie mit den Attributen zufrieden klicken Sie **[OK]**. QGIS wird automatisch den neuen Layer in die Legende einfügen und Sie können diesen dann auf die gleiche Art und Weise bearbeiten wie in Abschnitt *Einen vorhandenen Layer editieren*.

Mit dem DB Manager können Sie SpatialLite Layer ausführlicher managen. Siehe *DB Manager Plugin*.




### Einen neuen GPX Layer erstellen

Um eine GPX-Datei zu laden, müssen Sie das GPS Plugin laden. Das Menü *Erweiterungen*  *Erweiterungen verwalten und installieren ...* öffnet den Erweiterungen Dialog . Aktivieren Sie das  *GPS Werkzeuge* Kontrollkästchen.

Wenn dieses Plugin geladen ist wählen Sie *Neu* →  *Erstelle neuen GPX Layer* im *Layer* Menü. Im *Neue GPX-Datei speichern als ...* Dialog können Sie auswählen wo der neue GPX Layer gespeichert werden soll.

## 12.5.7 Mit Attributtabelle arbeiten

Die Attributtabelle stellt die Objekte eines ausgewählten Layers dar. Jede Zeile in der Tabelle repräsentiert ein Kartenobjekt und jede Spalte enthält bestimmte Informationen über das Objekt. Objekte in der Tabelle können gesucht, ausgewählt, verschoben oder sogar bearbeitet werden.

Um die Attributtabelle eines Vektorlayers zu öffnen machen Sie ihn aktiv indem Sie auf ihn in der Legende klicken. Dann wählen Sie aus dem *Layer* Menü  *Attributtabelle öffnen*. Es ist auch möglich einen Rechtsklick auf einen Layer zu machen und  *Attributtabelle öffnen* aus dem Drop-Down Menü auszuwählen und auf den  *Attributtabelle öffnen* Knopf in der Attribute Werkzeugleiste zu klicken.

Dies wird ein neues Fenster öffnen das die Attribute des Layers zeigt (*figure\_attributes\_1*). Die Anzahl der Objekte und die Anzahl der gewählten Objekte werden im Attributtabellentitel gezeigt.

### Objekte einer Attributtabelle abfragen

**Jede selektierte Zeile** in einer Attributtabelle repräsentiert alle Attribute eines Objektes in dem ausgewählten Layer. Die Attributtabelle zeigt alle Veränderungen bei einer Abfrage in der Attributtabelle im Kartenfenster und umgekehrt. Eine neue Abfrage in der Attributtabelle verursacht also eine Veränderung der im Kartenfenster als ausgewählt dargestellten Objekte und eine im Kartenfenster geänderte Auswahl von Objekten spiegelt sich durch eine Veränderung der ausgewählten Zeilen in der Attributtabelle wider.



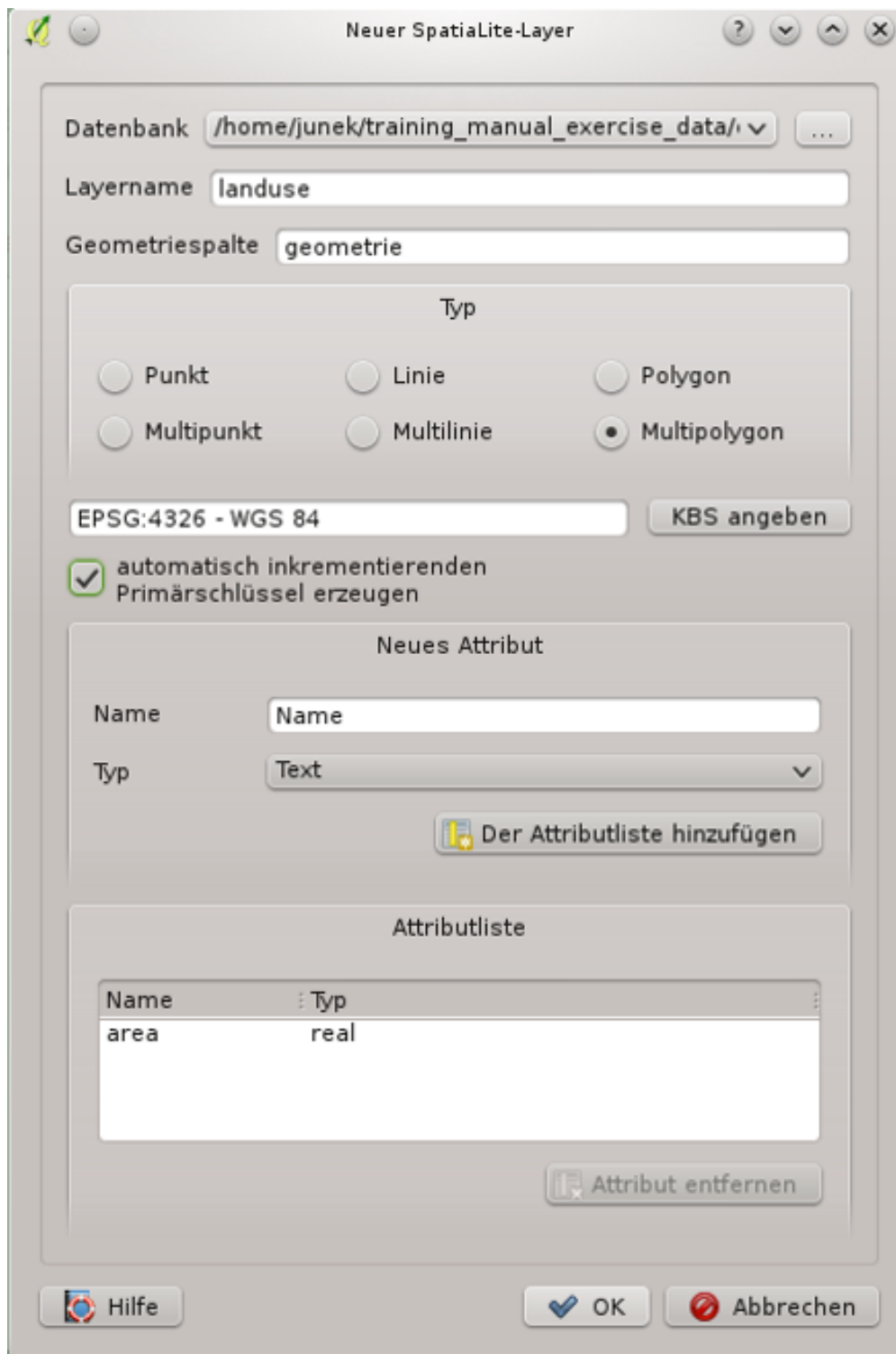


Abbildung 12.43: Neuer SpatiaLite-Layer Dialog 

	NAME_1	NAME_2	HASC_2	TYPE_2	ID
0	Alaska	Aleutians East	US.AK.AE	Borough	0
1	Alaska	Aleutians West	US.AK.AW	Census Area	1
2	Alaska	Anchorage	US.AK.AN	Municipality	2
3	Alaska	Bethel	US.AK.BE	Census Area	3
4	Alaska	Bristol Bay	US.AK.BR	Borough	4
5	Alaska	Denali	US.AK.DE	Borough	5
6	Alaska	Dillingham	US.AK.DI	Census Area	6
7	Alaska	Fairbanks Nor...	US.AK.FA	Borough	7
8	Alaska	Haines	US.AK.HA	Borough	8
9	Alaska	Juneau	US.AK.JU	City And Boro...	9
10	Alaska	Kenai Peninsula	US.AK.KN	Borough	10
11	Alaska	Ketchikan Gat	US.AK.KT	Borough	11

Abbildung 12.44: Attributtabelle des regions Layer

Zeilen können ausgewählt werden, indem Sie auf die Zeilennummer links neben der Zeile klicken. **Mehrere Zeilen** können ausgewählt werden, indem die `Strg` Taste während der Auswahl gedrückt wird. Eine **kontinuierliche Auswahl** ist möglich, indem Sie bei der Selektion die `Umschalt` Taste gedrückt halten, während Sie die Zeilennummern auswählen. Alle Zeilen zwischen der aktuell ausgewählten Zeile und der Mausfeilposition werden dadurch selektiert. Bewegt man den Mauszeiger in der Attributtabelle indem man in eine Zelle in der Tabelle klickt ändert die Zeilenauswahl nicht. Verändert man die Auswahl im Kartenfenster bewegt sich der Mauszeiger in der Attributtabelle nicht.

Jede Spalte kann sortiert werden, indem Sie auf die Kopfzeile klicken. Ein kleiner Pfeil zeigt die Sortierfolge an. Wenn er nach unten zeigt, werden die Werte von oben nach unten absteigend angezeigt. Wenn der Pfeil nach oben zeigt, werden die von oben nach unten aufsteigend angezeigt.

For a **simple search by attributes** on only one column, choose the *Column filter* → from the menu in the bottom left corner. Select the field (column) on which the search should be performed from the drop-down menu, and hit the **[Apply]** button. Then, only the matching features are shown in the attribute table.

Um eine Auswahl zu treffen müssen Sie das **Objekte mit einem Ausdruck wählen** Icon oberhalb der Attributtabelle verwenden. Mit **Objekte mit einem Ausdruck wählen** können Sie eine Untermenge einer Tabelle anhand einer *Funktionsliste*, wie Sie im **Feldrechner** vorhanden ist, definieren (siehe *Feldrechner*). Das Abfrageergebnis kann dann als neuer Vektorlayer gespeichert werden. Wenn Sie zum Beispiel Regionen vom Typ borough aus der Datei `regions.shp` der QGIS Beispieldaten finden wollen, müssen Sie das *Felder und Werte* Menü öffnen und das Feld, das Sie abfragen wollen, auswählen. Doppelklicken Sie das Feld 'TYPE\_2' und auch **[Alle eindeutig]**. Wählen und doppelklicken Sie 'Borough' in der Liste. Im Feld *Ausdruck* erscheint die folgende Abfrage:

```
"TYPE_2" = 'Borough'
```













Hier können Sie auch *Funktionsliste* → *Letztes (Selection)* verwenden um eine Auswahl zu treffen die Sie schon einmal verwendet haben. Der Ausdruckseditor speichert die letzten 20 verwendeten Ausdrücke ab.


Die zutreffenden Zeilen werden ausgewählt und die volle Anzahl zutreffender Zeilen erscheint dann in der Titelleiste der Attributtabelle sowie in der Statusleiste des Hauptfensters. Wollen Sie eine Suche durchführen, die dann nur die gewählten Objekte auf der Karte darstellt, verwenden Sie die Abfrageerstellung die in Kapitel *Abfrageeditor* beschrieben wird.

Um nur ausgewählte Objekte anzuzeigen verwenden Sie *Alle gewählten Objekte anzeigen* aus dem Menü links



unten.

Die anderen Knöpfe im Kopf der Attributtabelle stellen die folgenden Funktionalitäten zur Verfügung:

-  Bearbeitungsmodus umschalten um einen einzelnen Wert zu bearbeiten und um die unten beschriebenen Funktionalitäten zu ermöglichen (auch mit Strg+E)
-  Änderungen speichern (auch mit Strg+S)
-  Alles abwählen (auch mit Strg+U)
-  Auswahl nach Oben (auch mit Strg+T)
-  Auswahl umkehren (auch mit Strg+R)
-  Ausgewählte Zeilen in die Zwischenablage kopieren (auch mit Strg+C)
-  Zu den gewählten Zeilen zoomen (auch mit Strg+J)
-  Zu den gewählten Zeilen verschieben (auch mit Strg+P)
-  Gewählte Objekte löschen (auch mit Strg+D)
-  Neue Spalte für PostGIS Layer und für OGR Layer mit GDAL Version >=1.6 (auch mit Strg+W)
-  Spalte löschen für PostGIS Layer und für OGR Layer mit GDAL Version >=1.9 (auch mit Strg+L)
-  Feldrechner öffnen (auch mit Strg+I)

Unter diesen Knöpfen ist die Feldrechnerleiste, die Berechnungen, die schnell auf Attribute angewendet werden sollen, ermöglicht. Diese Leiste verwendet die gleichen Ausdrücke wie der  Feldrechner (see [Feldrechner](#)).

### Tipp: WKT Geometrie überspringen

Wenn Sie Attributdaten in externen Programmen (wie Excel) nutzen wollen verwenden Sie den  Ausgewählte Zeilen in die Zwischenablage kopieren Knopf. Sie können die Informationen ohne Vektorgeometrien kopieren wenn Sie *Einstellungen* → *Optionen* → *Datenquellen Menü*  *Geometrie in WKT aus Attributtabelle kopieren* deaktivieren.

### Ausgewählte Objekte als neuer Layer speichern


Die ausgewählten Objekte können in jedem OGR-unterstützten Vektorformat gespeichert werden und auch in ein anderes Koordinatenbezugssystem (KBS) transformiert werden. Öffnen Sie einfach das Rechte-Maus-Menü des Layers und klicken Sie auf :menuselection: *Speichern als* → um den Namen, das Format und das KBS der Ausgabedatei zu definieren (siehe Kapitel [Legende](#)). Um die Auswahl zu speichern vergewissern Sie sich dasscheckbox| *nur gewählte Objekt speichern* ausgewählt ist. Es ist genauso möglich OGR-Erstellungsoptionen innerhalb des Dialogs festzulegen.

### In neuen Layer einfügen

Objekte, die sich in der Zwischenablage befinden, können in einen neuen Layer eingefügt werden. Versetzen Sie den Layer zuerst in den Bearbeitungsmodus. Wählen Sie einige Objekte aus, kopieren Sie sie in die Zwischenablage und fügen Sie sie unter Zuhilfenahme von *Bearbeiten* → *Objekte einfügen als* und *Neuer Vektorlayer* oder *Neuer Memory-Vektorlayer* ein.

Dieses gilt für Objekte, die innerhalb von QGIS ausgewählt und kopiert wurden und auch für Objekte einer anderen Quelle die als Well-Known Text (WKT) definiert wurden.

## Arbeiten mit nicht räumlichen Attributtabelle

Mit QGIS können Sie auch nicht-räumliche Tabellen laden. Dies beinhaltet derzeit Tabellen die von OGR und Delimited Text, sowie vom PostgreSQL, MSSQL and Oracle Provider unterstützt werden. Die Tabellen können zum Nachschlagen benutzt werden oder nur um allgemein zu suchen und zu bearbeiten anhand der Tabellenansicht. Wenn Sie die Tabelle laden sehen Sie dies in der Legende. Sie kann mit dem  Attributtabelle öffnen Werkzeug geöffnet werden und ist dann editierbar wie jeder andere Attributtabelle.

Sie können zum Beispiel Spalten von nicht-räumlichen Tabellen verwenden um Attributwerte oder eine Spanne von Werten, die erlaubt ist, in einen bestimmten Vektorlayer während der Bearbeitung hinzugefügt zu werden definieren. Sehen Sie sich das Bearbeitungselement genauer im Abschnitt *Menü Felder* an, um mehr zu erfahren.

### 12.5.8 Ein-Zu-Mehrere-Beziehungen erstellen

Beziehungen sind eine Technik, die oft in Datenbanken zum Einsatz kommen. Das Konzept ist dass Objekte (Zeilen) verschiedener Layer (Tabellen) zueinander gehören können.

Als Beispiel nehmen wir einen Layer mit allen Regionen von Alaska (Polygon) in dem einige Attribute über den Namen und den Regionstyp sowie eine eindeutige ID (die als Primärschlüssel eingesetzt wird) vorhanden sind.

#### Fremdschlüssel

Dann erhalten Sie einen anderen Punktlayer oder Tabelle mit Informationen über Flughäfen die in den Regionen liegen und wollen diese ebenfalls im Auge behalten. Wenn Sie diese dem Regionen Layer hinzufügen wollen müssen Sie eine Eine-Zu-Mehrere-Beziehung anhand von Fremdschlüsseln erstellen weil es in den meisten Regionen mehrere Flughäfen gibt.

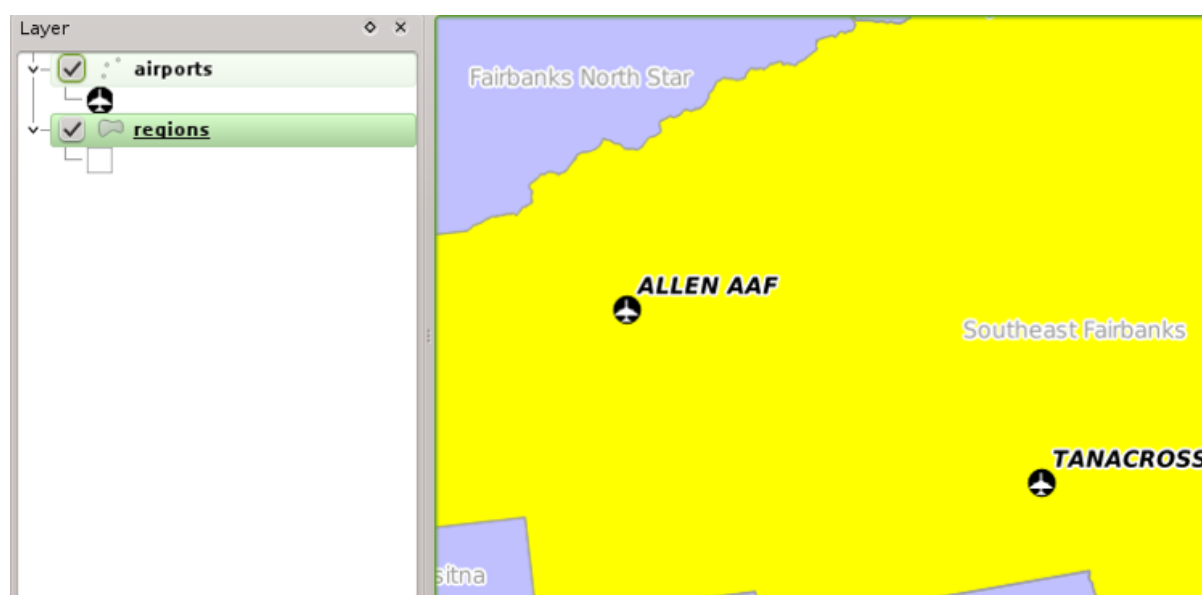


Abbildung 12.45: Die Region Alaska mit Flughäfen 

Zusätzlich zu den bereits bestehenden Attributen in der Airport Attributtabelle wird ein anderes Feld `fk_region` das als Fremdschlüssel verwendet werden soll, eingefügt (wenn Sie eine Datenbank verwenden würden Sie wahrscheinlich dafür einen Constraint definieren).

Dieses Feld `fk_region` enthält immer eine ID einer Region. Man kann es als Zeiger zu der Region, zu der es gehört, betrachten. Und sie können eine benutzerdefinierte Eingabemaske erstellen wobei QGIS sich um das Einrichten kümmert. Es funktioniert mit verschiedenen Anbietern (so dass sie es mit Shape- und CSV-Dateien benutzen können) und alles was Sie tun müssen ist QGIS die Beziehung zwischen Ihren Tabellen mitzuteilen.

## Layer

QGIS unterscheidet nicht zwischen einer Tabelle und einem Vektorlayer. Grundsätzlich gesehen ist ein Vektorlayer eine Tabelle mit einer Geometrie. Sie können Ihre Tabelle also als Vektorlayer einfügen. Zu Demonstrationszwecken können Sie die 'region' Shapedatei (mit Geometrien) und die 'airport' CSV-Tabelle (ohne Geometrien) und einen Fremdschlüssel (fk\_region) zu dem Layer 'region' laden (eine typische Ein-Zu-Mehrere Beziehung).

### Definition (Relation Manager)

Das erste was wir machen müssen ist QGIS über die Beziehungen zwischen den Layern wissen lassen. Dies wird in *Einstellungen* → *Projekteigenschaften* gemacht. Öffnen Sie das *Beziehungen* Menü und klicken Sie auf *Beziehung hinzufügen*.

- **Name** wird als Titel verwendet. Es sollte eine lesbare Zeichenfolge sein, die beschreibt, wofür die Beziehung verwendet werden soll. Wir werden ihn in diesem Fall einfach 'Airports' nennen.
- **Referenzierender Layer** ist der in dem der Fremdschlüssel steht. In diesem Fall ist dies der 'airports' Layer.
- **Referenzierendes Feld** zeigt an welches Feld auf den anderen Layer zeigt, also ist dies in diesem Fall fk\_region
- **Referenzierter Layer** ist der mit dem Primärschlüssel, auf den gezeigt wird, also ist dies in diesem Fall der 'regions' Layer
- **Referenziertes Feld** ist der Primärschlüssel des referenzierten Layers, also die ID.
- **ID** wird für interne Zwecke benutzt und muss eindeutig sein. Sie brauchen es vielleicht um benutzerdefinierte Eingabemasken zu erstellen sobald dies unterstützt wird. Wenn Sie es leer lassen wird eines für Sie erstellt aber Sie können auch selber eine zuweisen die einfacher zu handhaben ist.

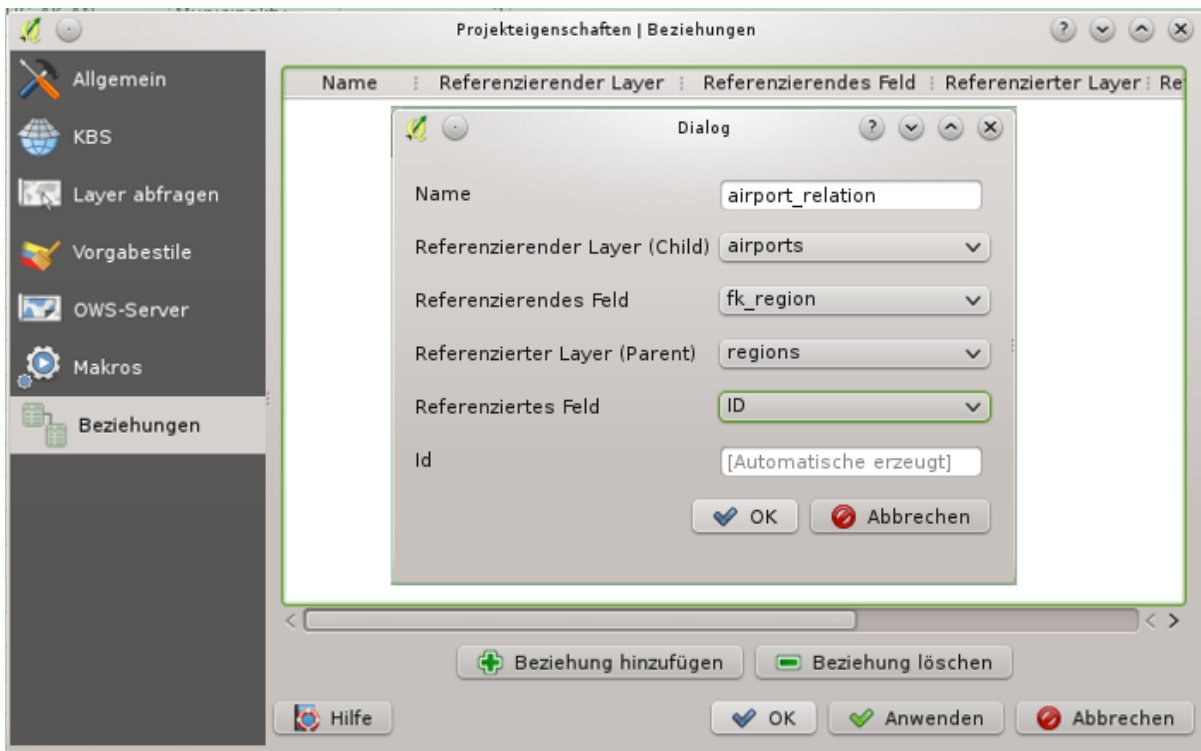



Abbildung 12.46: Relation Manager 

## Formulare

Jetzt wo QGIS von der Beziehung weiss wird es dazu benutzt das Formular das es erstellt zu verbessern. Da wir die Standard Formular Methode (autogenerated) nicht verändert haben, wird es Ihrem Formular einfach ein neues Bearbeitungselement hinzufügen. Also lassen Sie uns einen Layer 'region' in der Legende auswählen und das Objekte abfragen Werkzeug benutzen. Abhängig von Ihren Einstellungen öffnet sich das Formular entweder direkt oder Sie müssen den Identifizieren Dialog unter Aktionen öffnen.

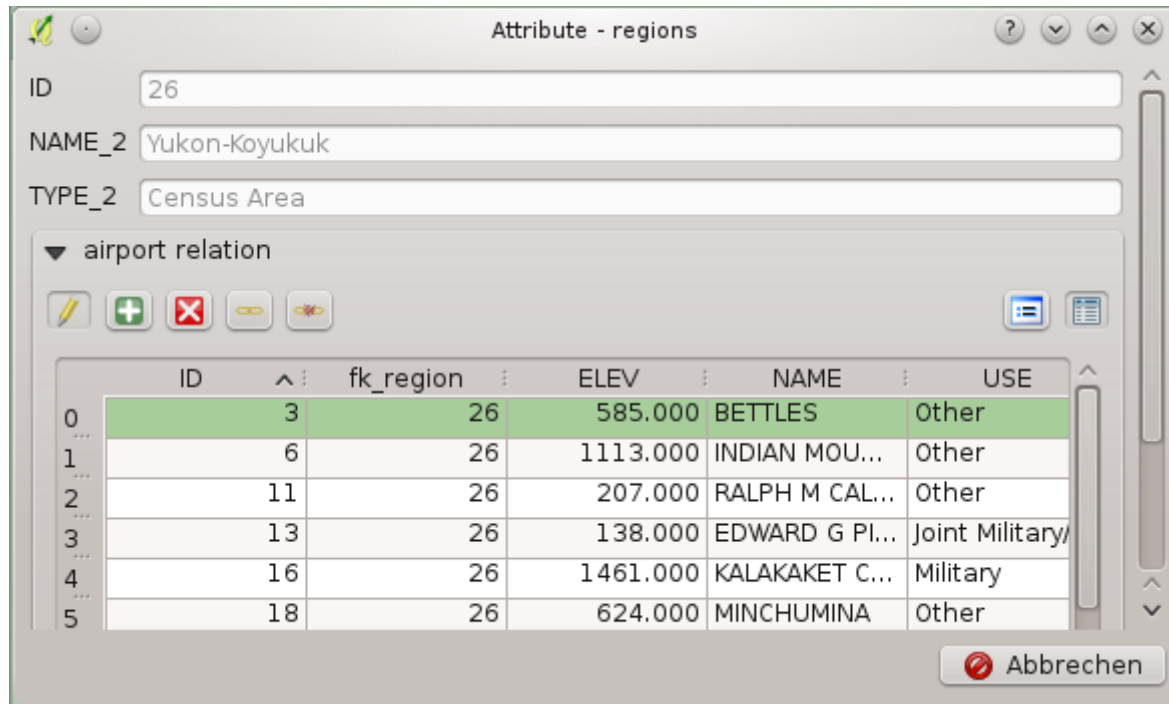


Abbildung 12.47: identifikationsergebnis Dialog von regions mit Beziehung zu airports 🐧

Wie Sie sehen können werden die dieser bestimmten region zugeordneten airports alle in einer Tabelle gezeigt. Und es können außerdem einige Knöpfe benutzt werden. Lassen Sie uns einen kurzen Blick darauf werfen

- Der Knopf ist zum Umschalten des Bearbeitungsmodus da. Seien Sie sich dessen bewusst dass es den Bearbeitungsmodus des 'airport' Layers umschaltet, trotzdem wir uns im Objektformular eines Objektes aus dem 'region' Layer befinden. Die Tabelle jedoch stellt Objekte des 'airport' Layer dar.
- Der Knopf wird dem 'airport' Layer ein neues Objekt hinzufügen. Und es wird den neuen airport der aktuellen region standardmäßig zuordnen.
- Der Knopf löscht den ausgewählten airport dauerhaft.
- Das Symbol öffnet einen neuen Dialog in dem Sie jeden vorhandenen 'airport' auswählen können was dann der aktuellen 'region' zugewiesen wird. Dies kann nützlich sein wenn Sie den 'airport' aus Versehen in der falschen 'region' erstellt haben.
- Das Symbol hebt die Verbindung zwischen dem ausgewählten 'airport' und der aktuellen 'region' wieder auf so dass dieser auf wirksame Art und Weise nicht mehr zugewiesen ist (der Fremdschlüssel wird auf NULL gesetzt).
- Die zwei Knöpfe rechts wechseln zwischen Tabellenansicht und Formularansicht wobei Sie bei der letzten Einstellung alle 'airports' in ihrer jeweiligen Form sehen können.

Wenn Sie an der 'airport' Tabelle arbeiten steht ein neuer Typ von Bearbeitungselement zur Verfügung, mit dem Sie das Objektformular der referenzierten 'region' in das Objektformular der 'airports' einbetten können. Es kann

verwendet werden wenn Sie die Layereigenschaften der 'airports' Tabelle öffnen, zum *Felder* Menü wechseln und den Typ des Bearbeitungselements des Fremdschlüssels 'fk\_region' auf Beziehungsreferenz ändern.

Wenn Sie jetzt einen Blick auf den Objektdialog werfen werden Sie sehen dass das Formular der 'region' in das 'airports' Formular eingebettet ist und sogar eine Kombobox vorhanden ist mit dem der aktuelle 'airport' zu einer anderen 'region' zugeordnet werden kann.

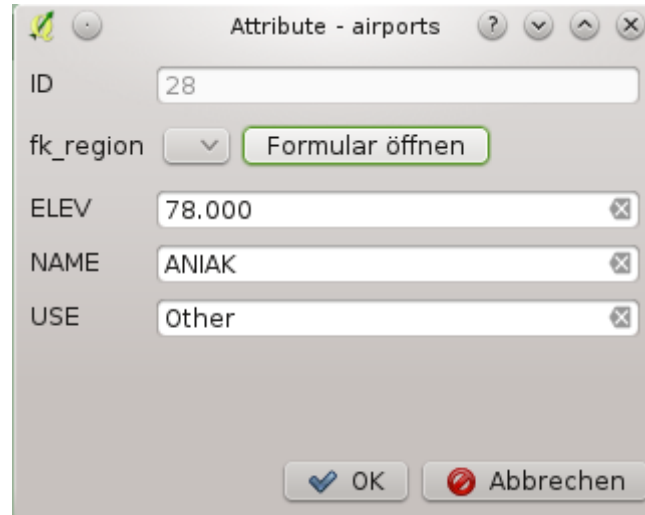


Abbildung 12.48: Objekte abfragen Dialog von 'airport' mit Beziehung zu 'regions' 🐧

## 12.6 Abfrageeditor

Mit der Abfrageerstellung können Sie eine Untermenge einer Tabelle definieren indem Sie eine SQL-ähnliche WHERE Klausel verwenden und das Ergebnis im Hauptfenster anzeigen lassen. Das Abfrageergebnis kann dann als neuer Vektorlayer gespeichert werden.

### 12.6.1 Abfrage

Öffnen Sie die **Abfrageerstellung** indem Sie die Layereigenschaften öffnen und zum *Allgemein* Menü wechseln. Klicken Sie unter *Objektuntermenge* auf den **[Abfrageerstellung]** Knopf um die *Abfrageerstellung* zu öffnen. Wenn Sie z.B. einen 'regions' Layer mit einem 'TYPE\_2' Feld haben könnten Sie nur die 'regions' des Typs 'borough' in der *Datenlieferanten spezifischer Filterausdruck* Fenster der Abfrageerstellung auswählen. [Figure\\_attributes\\_2](#) zeigt ein Beispiel von der Abfrageerstellung die mit dem `regions.shp` Layer aus dem QGIS Beispieldatensatz. Die Felder, Werte und Operatoren Bereiche helfen Ihnen die SQL-ähnliche Abfrage zu erstellen.

Die **Felder** Liste enthält alle Attributspalten der Attributtabelle die durchsucht werden soll. Um eine Attributspalte dem SQL WHERE Klausel Feld hinzuzufügen machen Sie einen Doppelklick auf den Namen in der Felderliste. Im Allgemeinen können Sie die diversen Felder, Werte und Operatoren verwenden um die Abfrage zu erstellen oder Sie können Sie einfach in die SQL Box eingeben.

Die **Werte** Liste führt die Werte einer Attributtabelle auf. Um alle möglichen Werte eines Attribut aufzulisten wählen Sie das Attribut in der Felderliste und klicken Sie den **[Alle]** Knopf. Um die ersten 25 eindeutigen Werte der Attributspalte aufzulisten wählen Sie die Attributspalte in der Felderliste und klicken Sie den **[Stichprobe]** Knopf. Um einen Wert dem SQL WHERE Klausel Feld hinzuzufügen, machen Sie einen Doppelklick auf den Namen in der Werteliste.

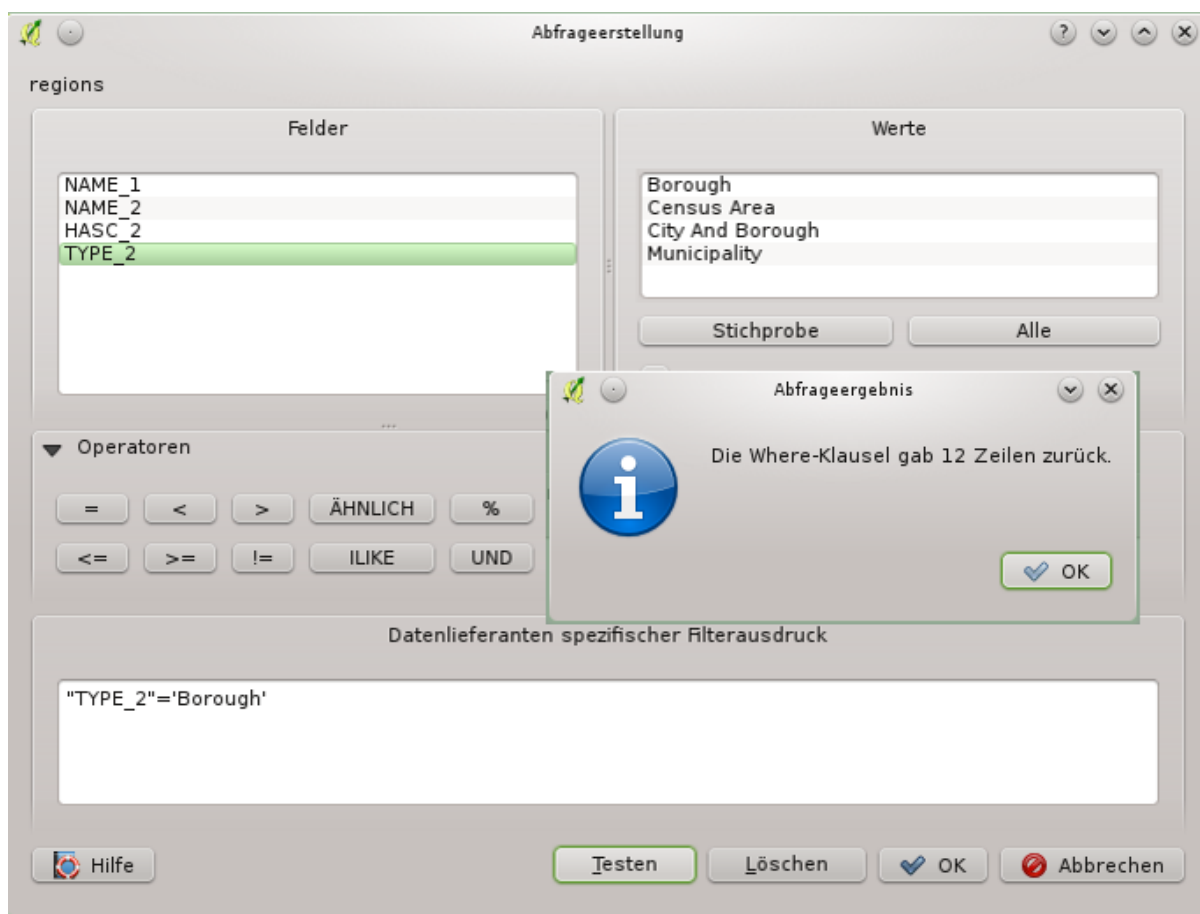



Abbildung 12.49: Abfrageerstellung 




Der **Operatoren** Bereich enthält alle verwendbaren Operatoren. Um einen Operator dem SQL WHERE Klausel Feld hinzuzufügen klicken Sie den entsprechenden Knopf. Es stehen relationale Operatoren ('=', '>', ...) String Vergleichsoperatoren ('LIKE') und logische Operatoren ('AND', 'OR', ...) zur Verfügung.

Der [**Testen**] Knopf zeigt ein Nachrichtenfenster mit der Anzahl von Objekten die auf die aktuelle Abfrage zutreffen, was nützlich im Prozess der Abfrageerstellung ist. Der [**Löschen**] Knopf löscht den Text im SQL WHERE Klausel Feld. Der [**OK**] Knopf schließt das Fenster und wählt die Objekte die auf die Abfrage zutreffen aus. Der [**Abbrechen**] Knopf schließt das Fenster ohne die aktuelle Auswahl zu ändern.

QGIS behandelt die resultierenden Teilmengen als ob Sie der ganze Layer sind. Wenn Sie z.B. den obigen Filter für 'Borough' anwenden, können Sie Ankerage nicht darstellen, abfragen, speichern oder bearbeiten, da dies eine 'Municipality' ist und von daher nicht in der Teilmenge enthalten sind.

Die einzige Ausnahme ist dass solange ihr Layer Teil der Datenbank ist, das Verwenden einer Teilmenge Sie vom Bearbeiten des Layers abhält.

## 12.7 Feldrechner

Mit  **Feldrechner** Knopf in der Attributtabelle können Sie Berechnungen auf Basis von bestehenden Attributwerten oder definierten Funktionen durchführen, z.B. um die Länge oder die Fläche von Geometrieobjekten zu berechnen. Die Ergebnisse können in eine neue Attributspalte geschrieben werden, in ein virtuelles Feld oder Sie können verwendet werden um Werte in einer vorhandenen Spalte zu updaten.

### Tipp: Virtuelle Felder

- Virtuelle Felder sind nicht dauerhaft vorhanden und werden nicht gespeichert.
- Ein Feld kann nur beim Erstellen virtuell gemacht werden.

Der Feldrechner ist jetzt über jeden Layer der Bearbeitung unterstützt erreichbar. Wenn Sie auf das Feldrechner Icon klicken öffnet sich ein Dialog (siehe [figure\\_attributes\\_3](#)). Wenn der Layer nicht im Bearbeitungsmodus ist wird eine Warnung gezeigt und das Verwenden des Feldrechners bewirkt, dass der Layer in den Bearbeitungsmodus gesetzt wird bevor die Berechnung gemacht wird.





Die schnelle Feldberechnungsleiste oben in der Attributtabelle ist nur sichtbar wenn der Layer bearbeitbar ist.

In der schnellen Feldberechnungsleiste wählen Sie erst einen bestehenden Feldnamen aus, öffnen dann den Ausdrucksdialog um Ihren Ausdruck zu erstellen oder schreiben ihn direkt in das Feld und klicken dann den **Alle aktualisieren** Knopf.

Im Feldberechnungsdialg müssen Sie erst auswählen ob sie nur ausgewählte Objekte updaten wollen, eine neues Feld anlegen, in das die Ergebnisse der Berechnung eingefügt werden oder ob Sie ein vorhandenes Feld erneuern wollen.

Wenn Sie sich entschließen ein neues Feld hinzuzufügen, müssen Sie einen Feldnamen, einen Feldtyp (Ganzzahl, Dezimalzahl, Text oder Datum), die Ausgabefeldbreite und die Genauigkeit eingeben (siehe [figure\\_attributes\\_3](#)). Zum Beispiel wenn Sie ein Ausgabefeldbreite von 10 und eine Genauigkeit von 3 wählen, heißt das, dass 6 Einträge vor dem Komma stehen, dann das Komma und dann weitere 3 Einträge für die Genauigkeit.

Ein kurzes Beispiel zeigt wie der Feldrechner arbeitet. Wir wollen die Länge des 'railroads' Layers aus dem QGIS Beispieldatensatz in km berechnen:

1. Laden Sie das Shape `railroads.shp` in QGIS und öffnen Sie die den Dialog  **Attributtabelle öffnen**.
2. Klicken Sie auf  **Bearbeitungsmodus umschalten** und öffnen Sie den  **Feldrechner** Dialog.
3. Wählen Sie das  **Neues Feld anlegen** Kontrollkästchen um die Berechnungen in ein neues Feld zu speichern.

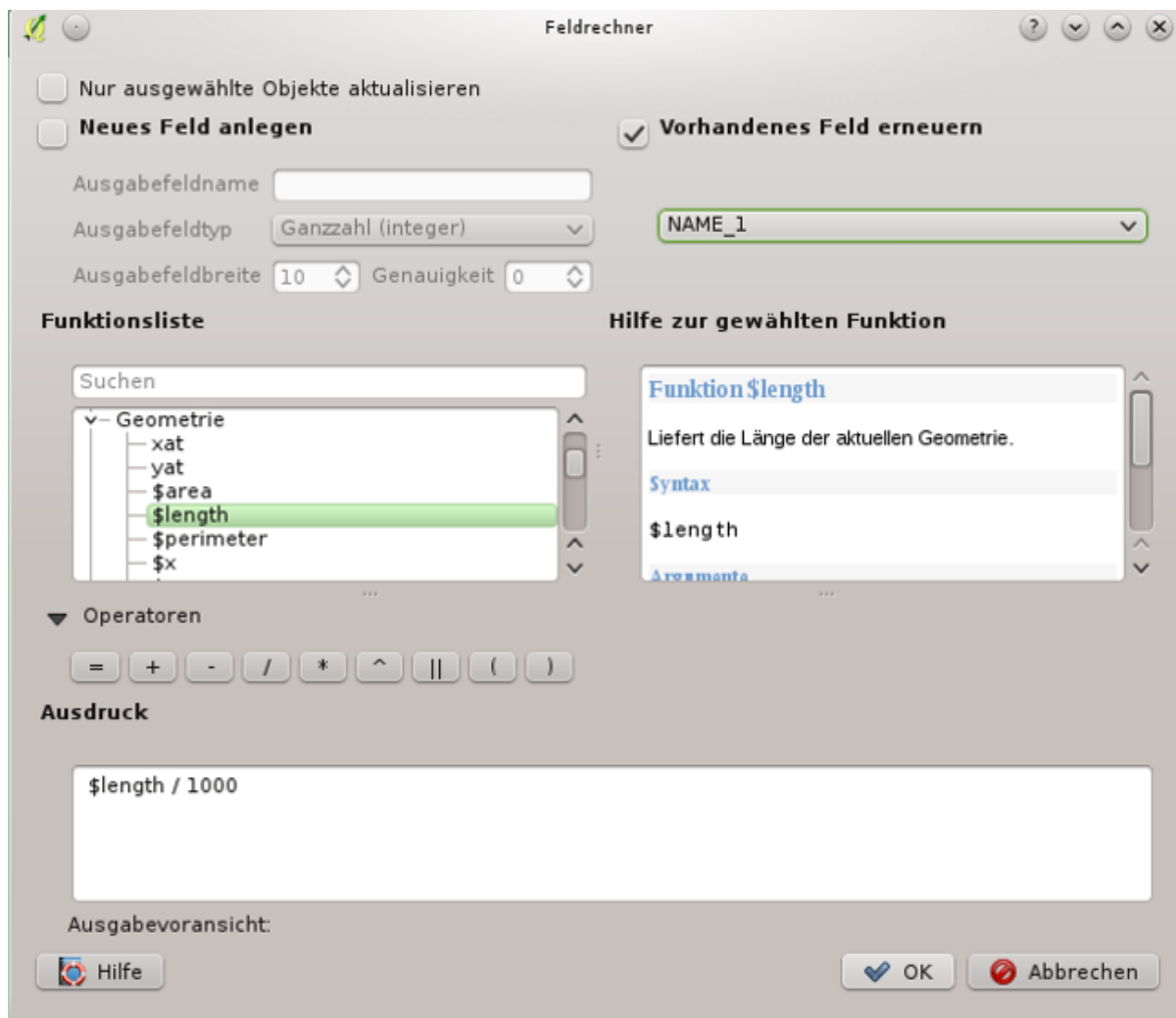



Abbildung 12.50: Feldrechner 

4. Setzen Sie `laenge` als Ausgabefeldname, `real` als Ausgabefeldtyp und definieren Sie die Ausgabefeldbreite mit 10 und die Ausgabefeldgenauigkeit mit 3.
5. Machen Sie jetzt einen Doppelklick auf die Funktion `$length` in der *Geometrie* Gruppe und fügen Sie sie in die Ausdruck Box des Feldrechners ein.
6. Vervollständigen Sie den Ausdruck indem Sie `"/1000"` im Feldrechnerausdruckfenster und klicken Sie **[OK]**.
7. Sie können jetzt eine neue Spalte `laenge` in der Attributtabelle finden.

Die erhältlichen Funktionen sind im Kapitel *Ausdrücke* aufgeführt.



---

## Arbeiten mit Rasterdaten

---

### 13.1 Arbeiten mit Rasterdaten

Dieser Abschnitt beschreibt wie man Rasterlayereigenschaften visualisiert und einstellt. QGIS verwendet die GDAL Bibliothek um Rasterdatenformate zu lesen und zu schreiben. Dies umfasst Arc/Info Binary Grid, Arc/Info ASCII Grid, GeoTIFF, Erdas Imagine und viele weitere Formate. GRASS Rastersupport wird durch einen eigenen QGIS Datenprovider bereitgestellt. Die Rasterdaten können auch im Lesemodus aus zip und gzip Archiven in QGIS geladen werden.

Zum Zeitpunkt der Erstellung dieses Dokumentes werden mehr als 100 Rasterdatenformate von der GDAL-Bibliothek unterstützt (siehe GDAL-SOFTWARE-SUITE *Literatur und Internetreferenzen*). Eine vollständige Liste ist unter [http://www.gdal.org/formats\\_list.html](http://www.gdal.org/formats_list.html) erhältlich.

---

**Bemerkung:** Einige der aufgelisteten Formate können auf ihrem Rechner aus unterschiedlichen Gründen nicht unterstützt werden. Einige brauchen z.B. kommerzielle Bibliotheken oder die GDAL Installation auf Ihrem Rechner, wurde ohne die Unterstützung für das entsprechende Format erstellt. Nur Formate, die getestet wurden, können ausgewählt werden, wenn Sie eine Rasterdatei in QGIS laden. Andere nicht getestete Formate können anhand des [GDAL] Alle Dateien (\*) Filter geladen werden.

---

Das Arbeiten mit GRASS Rasterdaten wird in Kapitel *GRASS GIS Integration* vorgestellt.

#### 13.1.1 Was ist ein Rasterlayer?

Rasterlayer 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 Rasterlayer sind Fernerkungsdaten wie Luft- oder Satellitenbilddaten oder modellierte Daten wie Höhenmodelle.

Im Gegensatz zu Vektorlayern haben Rasterlayer keine Verknüpfung zu einer Attributtabelle mit Werten für jede Zelle. Sie sind anhand der Pixelauflösung und der x/y Koordinate eines Eckpixels des Rasterlayers geocodiert. Dies ermöglicht es QGIS die Daten korrekt im Kartenfenster zu positionieren.

QGIS liest die Projektionsinformationen innerhalb des Rasterlayers (e.g. GeoTiff) oder aus einem passenden Worldfile um die Daten richtig darzustellen.

#### 13.1.2 Rasterlayer in QGIS laden

Rasterlayer werden geladen indem man auf das  Rasterlayer hinzufügen Icon klickt oder indem man die Option Layer →  Rasterlayer hinzufügen ... auswählt. Es können mehrere Layer gleichzeitig geladen werden indem

man `Strg` oder `Umschalt` gedrückt hält und auf mehrere Objekte im *Öffnen einer GDAL-Rasterdatenquelle* Dialog klicken.

Ist ein Rasterlayer in die Legende geladen können Sie auf den Layernamen mit der rechten Maustaste klicken um layerspezifische Eigenschaften auszuwählen oder zu aktivieren oder um einen Dialog zum Einstellen der Rastereigenschaften des Layers zu öffnen.

### Rechte-Maustaste Menü für Rasterlayer

- *Auf die Layerausdehnung zoomen*
- *Auf besten Maßstab zoomen (100%)*
- *Auf aktuelle Ausdehnung strecken*
- *In der Übersicht anzeigen*
- *Entfernen*
- *Kopieren*
- *KBS für Layer setzen*
- *Layer-KBS dem Projekt zuweisen*
- *Speichern als ...*
- *Eigenschaften*
- *Umbenennen*
- *Stil kopieren*
- *Neue Gruppe hinzufügen*
- *Alles ausklappen*
- *Alles zusammenfalten*
- *Zeichenreihenfolge aktualisieren*

## 13.2 Dialogfenster Rasterlayereigenschaften

Um die Eigenschaften eines Rasterlayers zu sehen oder einzustellen doppelklicken Sie auf den Layernamen in der Legende oder rechtsklicken Sie auf den Layernamen und wählen Sie *Eigenschaften* aus dem Kontextmenü. Dies öffnet den *Layer-eigenschaften* Dialog (siehe [figure\\_raster\\_1](#)).

Es gibt mehrere Menüs in diesem Dialog:

- *Allgemein*
- *Stil*
- *Transparenz*
- *Pyramiden*
- *Histogramm*
- *Metadaten*

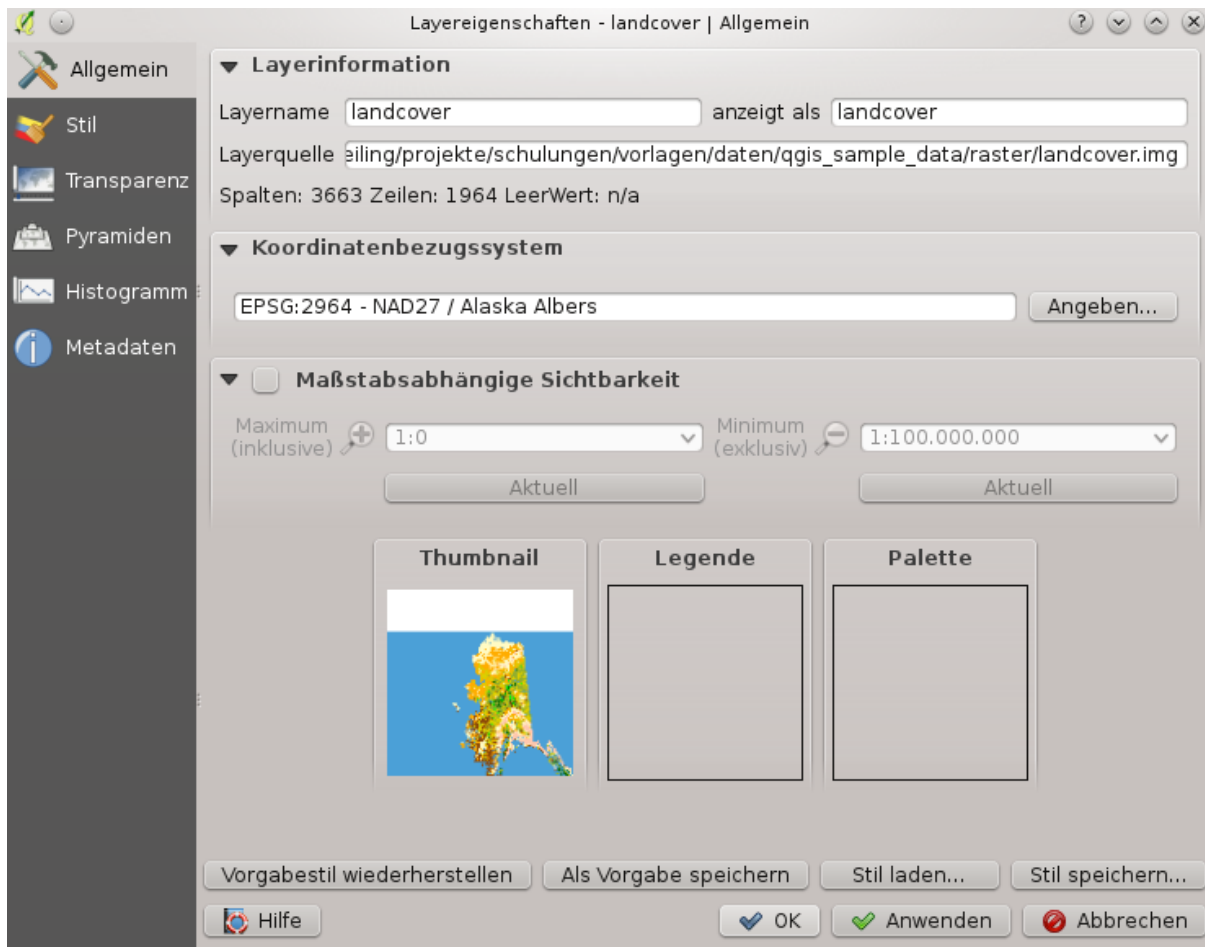


Abbildung 13.1: Dialog Rasterlayereigenschaften 

## 13.2.1 Menü Allgemein

### Layerinformation

Das Menü *Allgemein* stellt grundlegende Informationen über den ausgewählten Rasterlayer dar, einschließlich der Layerquelle, dem Anzeigenamen (der verändert werden kann) und der Anzahl von Spalten, Zeilen und LeerWerten des Rasterlayers.

### Koordinatenbezugssystem

Hier können Sie das Koordinatenbezugssystem (KBS) als PROJ.4 Text ablesen. Wenn diese Einstellung nicht richtig ist können Sie Sie verändern indem Sie den Knopf [**Angeben**] klicken.

### Maßstabsabhängige Sichtbarkeit

Zusätzlich kann eine skalenabhängige Sichtbarkeit eingestellt werden. Dazu muss das Kontrollkästchen aktiviert sein und ein entsprechender Maßstab, indem Ihre Daten im Kartenfenster dargestellt werden, eingetragen werden.

Unten können Sie einen Thumbnail, sein Legendensymbol und die Palette sehen.

## 13.2.2 Menü Stil

### Kanaldarstellung

QGIS bietet vier verschiedene *Darstellungsart*. Die ausgewählte Darstellungsart hängt vom Datentyp ab.

1. Multikanalfarbe - wenn die Datei ein Multiband mit mehreren Kanälen ist (z.B. bei einem Satellitenbild mit mehreren Bändern)
2. Palette - wenn ein Einkanalbild eine indizierte Palette besitzt (z.B. benutzt bei digitalen Topographischen Karten)
3. Einkanalgraustufen - das Bild oder ein Kanal davon wird grau dargestellt; QGIS benutzt diese Darstellung wenn die Datei weder Multibänder, noch eine indizierte Palette noch eine kontinuierliche Palette (z.B. in einer Schummerungskarte benutzt) besitzt.
4. Einkanalpseudofarbe - diese Darstellung ist bei Dateien mit kontinuierlicher Palette oder Farbkarte (z.B. wie sie in Höhenkarten verwendet wird) möglich

### Multikanalfarbe

Mit der Darstellung Multikanalfarbe werden drei ausgewählte Kanäle des Bildes dargestellt, wobei jedes Band die rote, grüne oder blaue Komponente zum Erstellen eines Farbbildes darstellt.

Diese Auswahl bietet Ihnen ein breites Spektrum von Optionen zum Verändern des Erscheinungsbildes des Rasterlayers an. Zuerst müssen Sie den Datenbereich Ihres Bildes abrufen. Dieser kann durch das Wählen von *Ausdehnung* und das Klicken von [**Laden**] ermittelt werden. QGIS kann die *Min* und *Max* Werte der Kanäle  *Schätzen (schneller)* oder die  *Genau (langsamer) Genauigkeit* verwenden.

Jetzt können Sie die Farben mit Hilfe des *Min/Max Werte* Bereichs skalieren. Viele Bilder haben ein paar sehr tiefe und hohe Daten. Diese Ausreißer können anhand der  *Kumulativer Anzahlschnitt* Einstellung beseitigt werden. Der voreingestellte Datenbereich ist auf 2% bis 98% der Datenwerte eingestellt und kann manuell angepasst werden. Mit dieser Einstellung kann das graue Aussehen des Bildes verschwinden. Mit der Skalierungsoption  *Min/max* erstellt QGIS eine Farbtabelle von allen im Originalbild enthaltenen Daten (z.B. erstellt QGIS eine Farbtabelle mit 256 Werten, vorausgesetzt Sie haben 8bit-Kanäle). Sie können Ihre Farbtabelle auch mit der  *Mittlere +/- Standardabweichung*  $x$   berechnen. Dann werden nur Werte innerhalb der Standardabweichung oder innerhalb mehrerer Standardabweichungen verwendet. Dies ist nützlich wenn ein oder zwei Zellen mit anormalen hohen Werten, die einen negativen Effekt auf die Darstellung des Rasterbildes haben, in einem Grid auftauchen.



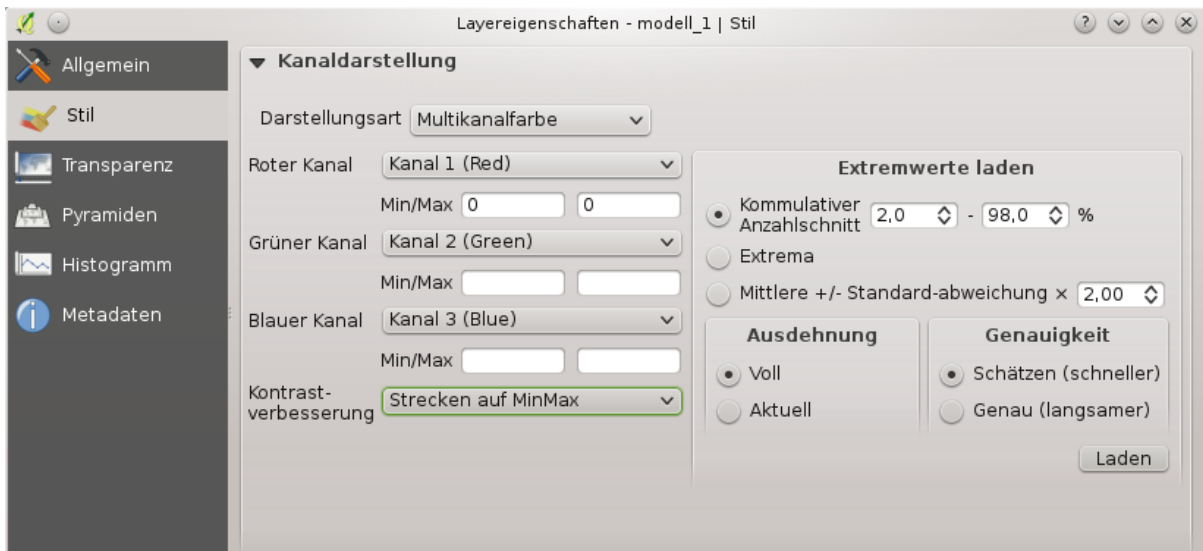


Abbildung 13.2: Rasterdarstellung - Multikanalfarbe 

Alle Berechnungen können auch für die  *Aktuell* Ausdehnung durchgeführt werden.

**Tipp: Einen einzelnen Kanal eines Mehrkanal-Rasterlayers anzeigen**

Wenn Sie sich nur einen einzelnen Kanal eines Multikanalfarbe Bildes (z.B. Rot) ansehen wollen kommen Sie vielleicht auf die Idee den Grün- und Blaukanal auf "Nicht gesetzt" einzustellen. Dies ist nicht der korrekte Weg. Um den Rotkanal darzustellen stellen Sie den Bildtyp auf 'Einkanalgraustufen' ein und wählen Sie dann Rot als Kanal, der für Grau benutzt werden soll, aus

**Palette**

Dies ist die voreingestellt Darstellungsart für Singleband-Dateien die bereits eine Farbtabelle besitzen, wobei jedem Pixel eine bestimmte Farbe zugewiesen wird. In diesem Fall wird die Palette automatisch dargestellt. Wenn Sie die Farben, die einem bestimmten Wert zugewiesen werden, ändern wollen, Doppelklicken Sie auf die Farbe und der *Farbauswahl* Dialog erscheint. Seit QGIS 2.2. ist es möglich dem Farbwert eine Beschriftung zuzuweisen. Die Beschriftung erscheint dann in der Legende des Rasterlayers.

**Kontrastverbesserung**

**Bemerkung:** Wenn GRASS Rasterlayer hinzugefügt werden wird die Option **Kontrastverbesserung** immer automatisch auf **Strecken auf MinMax** eingestellt, ungeachtet der Einstellungen in den QGIS Optionen.

**Einkanalgraustufen**

Mit dieser Darstellungsart können Sie ein Einkanalbild mit einem *Farbverlauf* darstellen: 'Schwarz nach Weiß' oder 'Weiß nach Schwarz'. Sie können einen *Min* und einen *Max* Wert festlegen indem Sie erst *Ausdehnung* auswählen und dann [**Laden**] drücken. QGIS kann die *Min* und *Max* Werte der Kanäle  *Schätzen (schneller)* oder die  *Genau (langsamer)* Genauigkeit verwenden.

Im *Extremwerte laden* Abschnitt ist das Skalieren der Farbtabelle möglich. Ausreißer können anhand der  *Kumulativer Anzahlschnitt* Einstellung beseitigt werden. Der voreingestellte Datenbereich ist von 2% bis 98% der Datenwerte eingestellt und kann manuell angepasst werden. Mit dieser Einstellung kann der graue Charakter des Bildes verschwinden. Weitere Einstellungen können mit  *Extrema* und  *Mittlere +/- Standardabweichung*  $x$   gemacht werden. Während die erste eine Farbtabelle mit allen Daten, die im Originalbild enthalten sind, erstellt, erstellt die zweite eine Farbtabelle die nur Werte innerhalb der Standardabweichung oder mehrerer Standardabweichungen berücksichtigt. Dies ist nützlich wenn Sie eine oder zwei Zellen mit anormal hohen Werten in einem Grid haben die einen negativen Einfluss auf die Darstellung des Rasterlayers haben.

**Einkanalpseudofarbe**

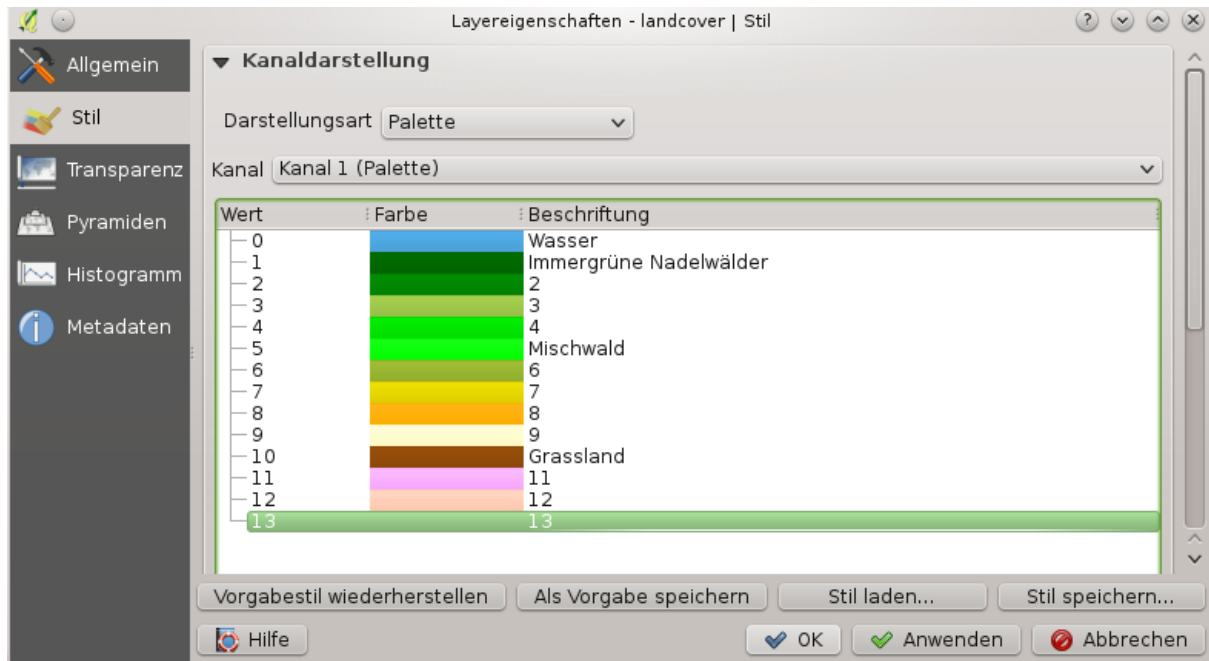


Abbildung 13.3: Rasterdarstellung - Palette 

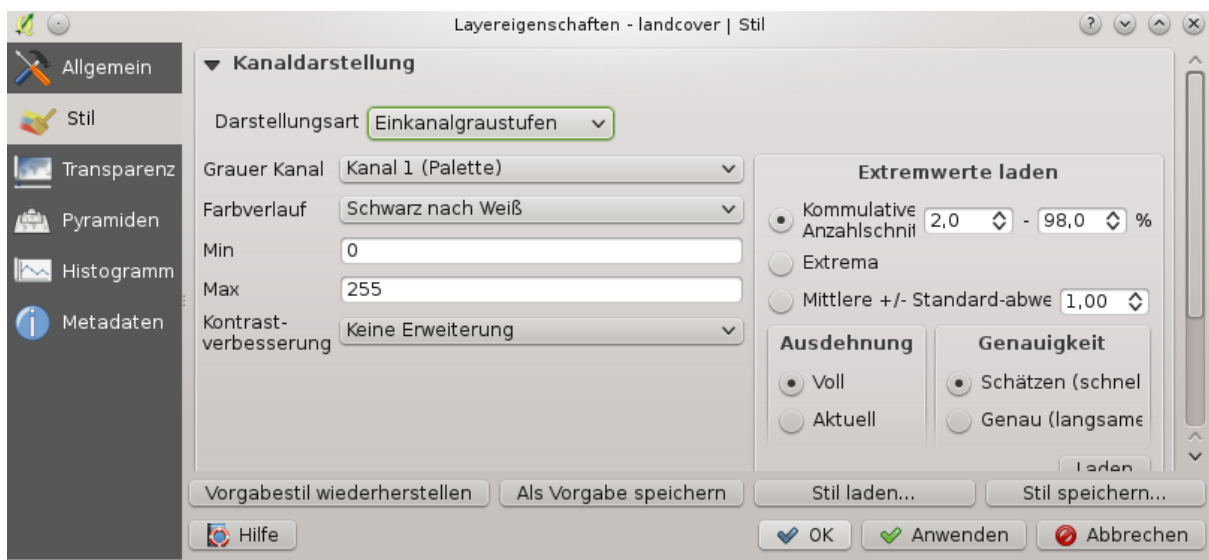


Abbildung 13.4: Rasterdarstellung - Einkanalgrastufen 

Dies ist eine Darstellungsoption für Einkanaldateien die eine kontinuierliche Palette enthalten. Sie können hier auch individuelle Karten für die einzelnen Kanäle erstellen. Es sind drei Typen von Farbinterpolation möglich:

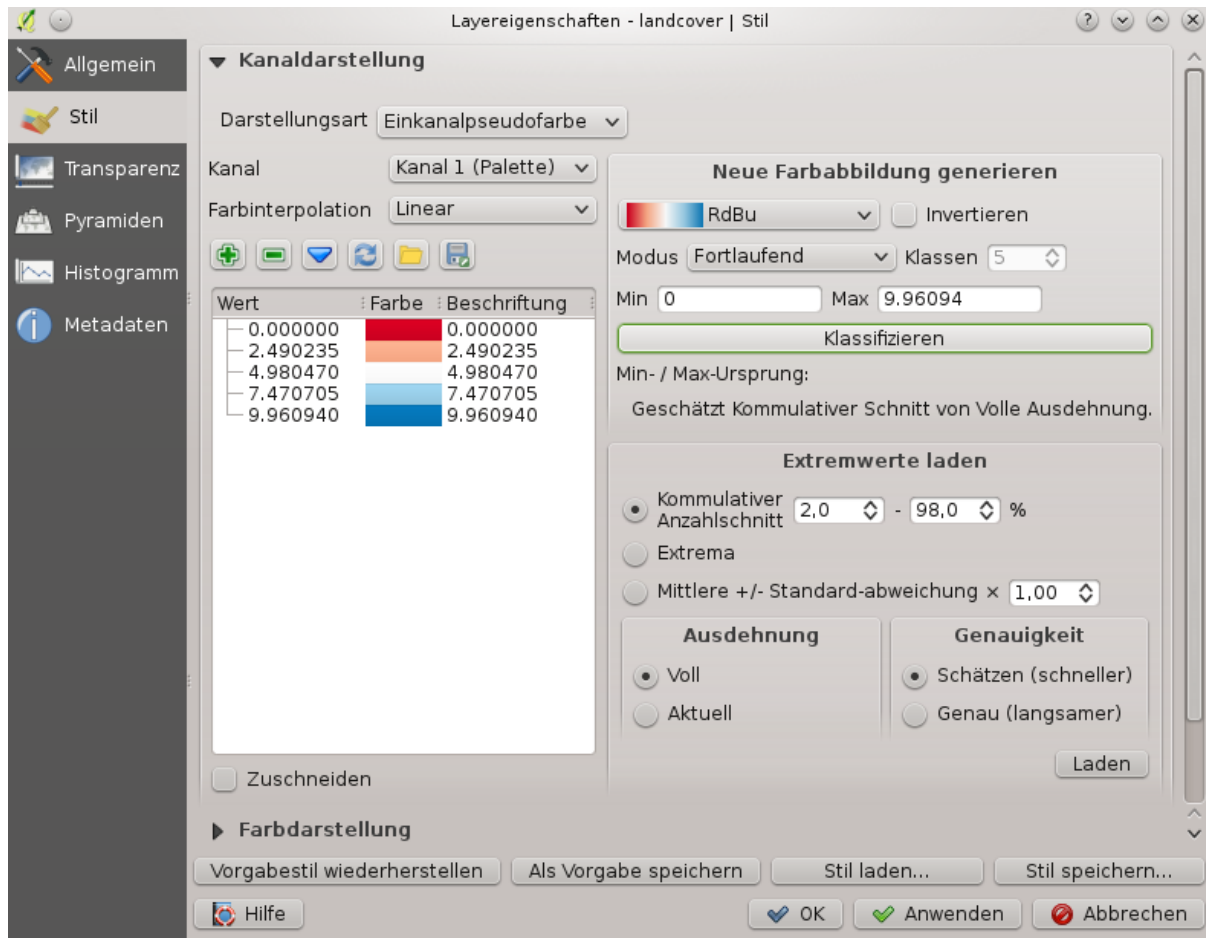









Abbildung 13.5: Rasterdarstellung - Einkanalpseudofarbe 🐧

1. Diskret
2. Linear
3. Genau

Im linken Block fügt der Knopf  Werte manuell hinzufügen einen Wert zu der individuellen Farbtabelle hinzu. Der Knopf  Gewählte Zeile löschen löscht einen Wert aus der individuellen Farbtabelle und der  Farbabbildungselemente Knopf sortiert die Farbtabelle gemäß den Pixelwerten in der Wertespalte. Ein Doppelklick auf die Wertespalte macht das Eingeben eines spezifischen Wertes möglich. Das Doppelklicken auf die Farbspalte öffnet den Dialog *Change color* in dem Sie eine Farbe die auf den Wert angewendet werden soll auswählen können. Des weiteren können Sie auch Beschriftungen für jede Farbe hinzufügen, allerdings wird dieser Wert nicht angezeigt wenn Sie das Objekt abfragen Werkzeug benutzen. Sie können auch auf den Knopf `mActionDraw` Farbabbildung aus Kanal laden klicken welcher versucht die Tabelle aus dem Kanal zu laden (falls dieser eine hat). Und Sie können die Knöpfe  Farbabbildung aus Datei laden oder  Farbabbildung in Datei speichern verwenden um eine bestehende Farbtabelle zu laden oder eine definierte Farbtabelle für andere Sitzungen zu speichern.

Im rechten Block wird Ihnen durch *Neue Farbabbildung generieren* das Erstellen von neuen kategorisierten Farbkarten ermöglicht. Für den *Modus*  'Gleiches Intervall' müssen Sie nur die Anzahl der *Klassen*  auswählen und den Knopf *Klassifizieren* drücken. Sie können die Farben der Farbkarte invertieren indem Sie auf das  *Invertieren* Kontrollkästchen klicken. Im Fall des *Modus*  'Fortlaufend' erstellt QGIS automatisch Klassen abhängig von *Min* und *Max*. Das Definieren von Extremwerten kann mit Hilfe des *Extremwerte laden* Abschnitts durchgeführt werden. Viele Bilder haben einige wenige sehr tiefe und hohe Daten. Diese Ausreißer können

anhand der  *Kumulativer Anzahlschnitt* Einstellung beseitigt werden. Der voreingestellte Datenbereich ist von 2% bis 98% der Datenwerte eingestellt und kann manuell angepasst werden. Anhand dieser Einstellung kann der graue Charakter des Bildes verschwinden. Mit der Skalierungsoption  *Extrema* erstellt QGIS eine Farbtabelle mit allen im Originalbild enthaltenen Daten (z.B. erstellt QGIS eine Farbtabelle mit 256 Werten, vorausgesetzt Sie haben 8bit Kanäle). Sie können Ihre Farbtabelle auch anhand der  *Mittlere +/- Standardabweichung* x  berechnen. Dann werden nur Werte innerhalb der Standardabweichung oder mehrerer Standardabweichungen für die Farbtabelle berücksichtigt.

## Farbdarstellung

Für jede *Kanaldarstellung* ist eine *Farbdarstellung* möglich.


Sie können auch spezielle Darstellungseffekte für Ihre Rasterdatei(en) erreichen indem Sie Mischmodi verwenden (siehe *Vektorlayereigenschaften*).

Weitere Einstellungen können durch das Verändern der *Helligkeit*, der *Sättigung* und des *Kontrast* gemacht werden. Sie können auch eine *Graustufen* Option verwenden bei der Sie zwischen 'Nach Helligkeit', 'Nach Leuchtkraft' und 'Nach Durchschnitt' wählen können. Für einen Farbwert in der Farbtabelle können Sie die 'Stärke' verändern.

## Abtastung

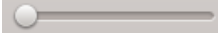
Die *Abtastung* Option kommt zur Erscheinung wenn Sie in ein Bild herein- oder herauszoomen. Abtastungsmodi können die Erscheinung der Karte optimieren. Sie berechnen eine neue Grauwertmatrix anhand einer geometrischen Transformation.



Abbildung 13.6: Rasterdarstellung - Abtastung 

Wenn Sie die 'Nächster Nachbar' Methode anwenden kann die Karte eine pixelige Struktur beim Hineinzoomen haben. Dieses Erscheinungsbild kann verbessert werden indem man die 'Bilinear' oder 'Kubisch' Methode verwendet, die scharfe Objekte verwischt. Der Effekt ist ein weicheres Bild. Diese Methode kann z.B. auf digitale Topographische Karten angewendet werden.


### 13.2.3 Menü Transparenz

QGIS besitzt die Fähigkeit jeden Rasterlayer auf einem unterschiedlichen Transparenzlevel darzustellen. Verwenden Sie den Transparenzschieberegler  um einzugeben bis zu welchem Ausmaß die darunterliegenden Layer (falls es welche gibt) sichtbar durch den aktuellen Rasterlayer sein sollen. Dies ist sehr nützlich wenn Sie mehr als einen Rasterlayer überlagern wollen (z.B. wenn eine Schummerungskarte von einer klassifizierten Rasterkarte überlagert werden soll). So wird die Karte dreidimensionaler aussehen.



Zusätzlich können Sie einen Rasterwert eingeben der als **Leerwert** im *Zusätzlicher Leerwert* Menü behandelt wird.

Die Transparenz kann noch flexibler über die *Transparente Pixelliste* angepasst werden. Die Transparenz jedes Pixels kann hier eingestellt werden.

Als Beispiel wollen wir die Wasserflächen aus dem Rasterlayer `landcover.tif` auf eine Transparenz von 20% setzen. Folgende Schritte sind dazu notwendig:

1. Laden Sie die Rasterdatei `landcover.tif`.
2. Öffnen Sie den Dialog *Layereigenschaften* indem Sie auf den Namen in der Legende doppelklicken, oder im Rechte-Maustaste Menü *Eigenschaften* auswählen.
3. Wählen Sie das Menü *Transparenz*.
4. Wählen Sie 'Keines' aus dem *Transparenzkanal* Menü.
5. Klicken Sie den  Werte manuell hinzufügen Knopf. Eine neue Zeile erscheint in der Pixelliste.
6. Geben Sie den Rasterwert in die 'Von' und 'Nach' Spalte ein (wir benutzen hier 0) und passen Sie die Transparenz auf 20% an.
7. Drücken Sie den Knopf [**Anwenden**] und schauen Sie sich das Ergebnis an.

Sie können Schritte 5 und 6 wiederholen um mehr Werte mit benutzerdefinierter Transparenz einzustellen.

Wie Sie sehen können ist es recht einfach die benutzerdefinierte Transparenz einzustellen, aber es kann ganz schön viel Arbeit sein. Deswegen können Sie den Knopf  In Datei exportieren benutzen um Ihre Transparenzliste in eine Datei zu speichern. Der Knopf  Aus Datei importieren lädt Ihre Transparenzeinstellungen und wendet sie auf den aktuellen Rasterlayer an.

### 13.2.4 Menü Pyramiden

Hochaufgelöste Rasterlayer können das Navigieren in QGIS verlangsamen. Indem Sie kleiner aufgelöste Kopien der Daten (Pyramiden) erstellen kann die Performanz erheblich verbessert werden, da QGIS die am besten geeignete Auflösung für den jeweiligen Zoomlevel auswählt.

Sie brauchen dazu Schreibrecht in dem Ordner, in dem sich sie Originaldaten befinden.

Sie können mehrere Resampling-Methoden zum Berechnen der Pyramiden verwenden:

- Nächster Nachbar
- Durchschnitt
- Gauß
- Kubisch
- Modus
- Keine

Wenn Sie 'Intern (wenn möglich)' aus dem *Übersichtsformat* Menü wählen, versucht QGIS Pyramiden intern zu erstellen. Sie können auch 'Extern' und 'Extern (Erdas Imagine)' wählen.

Bitte beachten Sie dass das Erstellen von Pyramiden die Originaldatei verändern kann und sind sie erstmal erstellt können Sie nicht entfernt werden. Wenn Sie eine 'nichtpyramidierte' Version Ihres Rasters erhalten wollen, machen Sie eine Backupkopie vor dem Erstellen von Pyramiden.

### 13.2.5 Menü Histogramm

Das *Histogramm* Menü ermöglicht es Ihnen die Verteilung der Kanäle oder Farben in Ihrem Rasterlayer anzuzeigen. Das Histogramm wird automatisch erzeugt wenn Sie das *Histogramm* Menü öffnen. Alle bestehenden

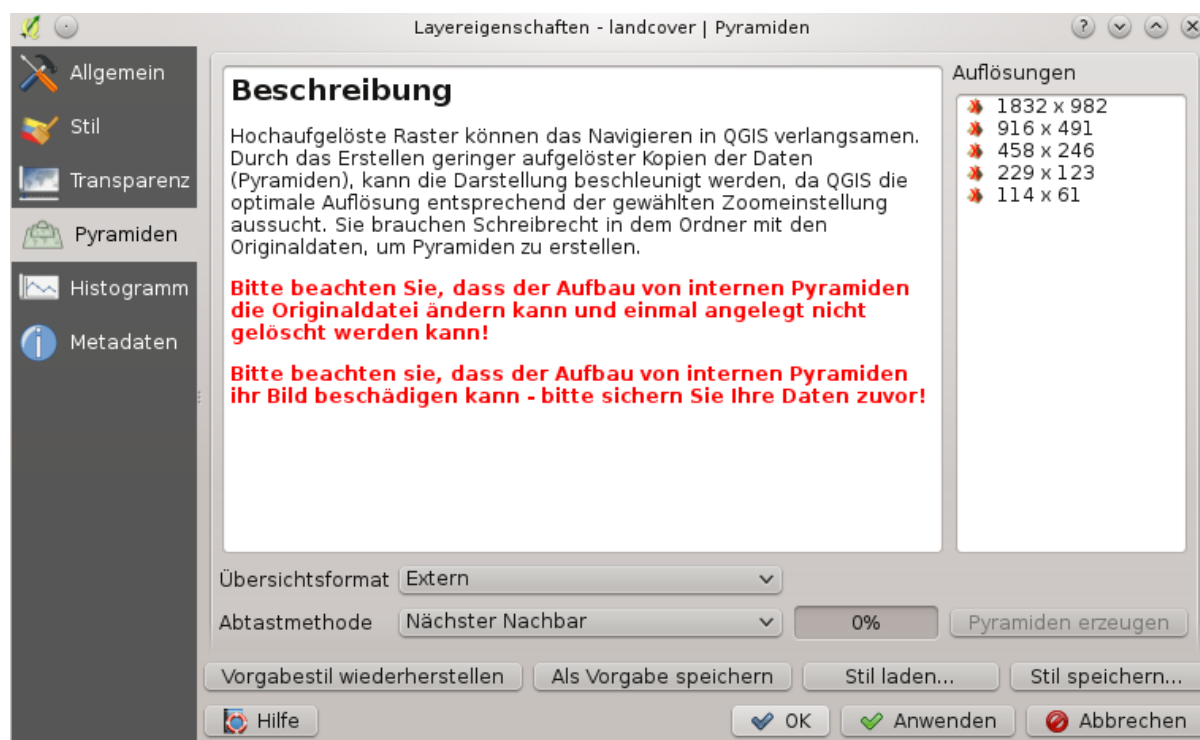




Abbildung 13.7: Das Menü Pyramiden 

Kanäle werden zusammen dargestellt. Sie können das Histogramm als Bild mit dem  Knopf speichern. Mit der *Sichtbarkeit* Option im  *Prüfs/Aktionen* Menü können Sie Histogramme der individuellen Kanäle darstellen. Sie werden die Option  *Gewählten Kanal anzeigen* auswählen müssen. Die *Extrema-Optionen* ermöglichen Ihnen ‘Extrema-Markierungen immer anzeigen’, ‘Zu Extremen zoomen’ und ‘Stile auf Extrema setzen’. Mit der *Aktionen* Option können Sie ‘Zurücksetzen’ und ‘Histogramm Neuberechnen’ nachdem Sie die *Extrema* Optionen gewählt haben.

### 13.2.6 Menü Metadaten

Das *Metadaten* Menü stellt eine Fülle von Informationen über den Rasterlayer dar, einschließlich Statistiken über jeden Kanal im aktuellen Rasterlayer. In diesem Menü können Einträge für *Beschreibung*, *Beschreibung*, *Metadaten-URL* und *Eigenschaften* gemacht werden. In *Eigenschaften* werden Statistiken nach dem Prinzip ‘was brauche ich’ erstellt, so dass es gut sein kann dass für einen Rasterlayer noch keine Statistik erstellt oder gesammelt wurde.

## 13.3 Rasterrechner

Der *Rasterrechner* im *Raster* Menü ermöglicht es Ihnen Berechnungen auf Basis von bestehenden Rasterpixelwerten durchzuführen (siehe [figure\\_raster\\_10](#)). Die Ergebnisse werden in einen neuen Rasterlayer in einem GDAL-unterstützten Format geschrieben.

Die **Rasterkanäle** Liste enthält alle geladenen Rasterlayer die benutzt werden können. Um einen Rasterlayer dem Rasterrechnerausdruck Feld hinzuzufügen, machen Sie einen Doppelklick auf seinen Namen in der Felder Liste. Sie können dann die Operatoren zum konstruieren von Berechnungsausdrücken benutzen oder Sie geben Sie einfach in die Box ein.

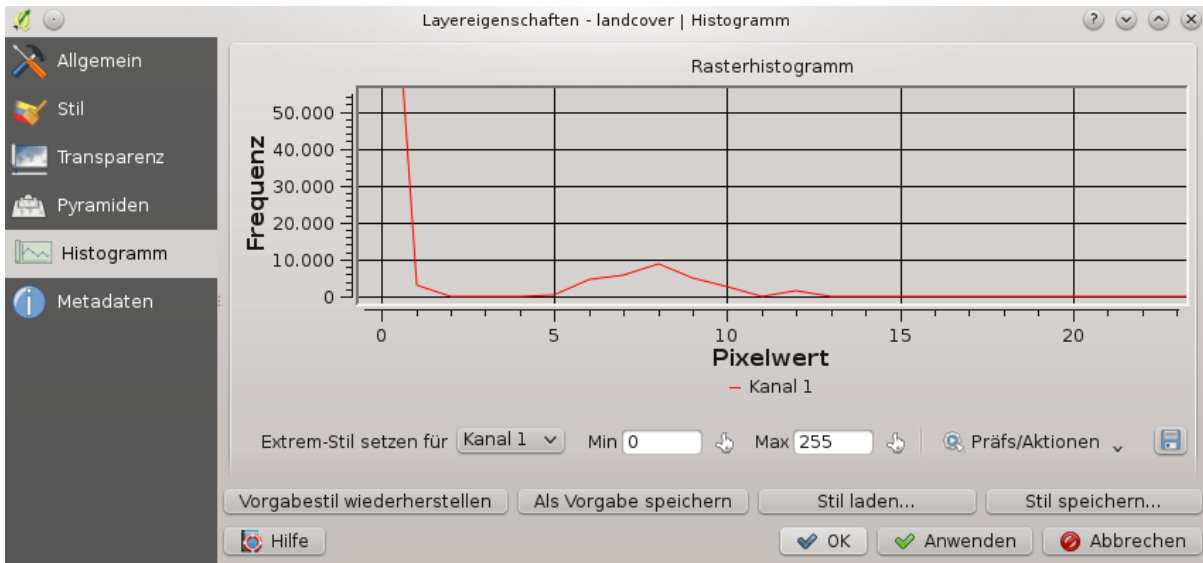



Abbildung 13.8: Rasterlayer Histogramm 

Abbildung 13.9: Rasterlayer Metadaten 



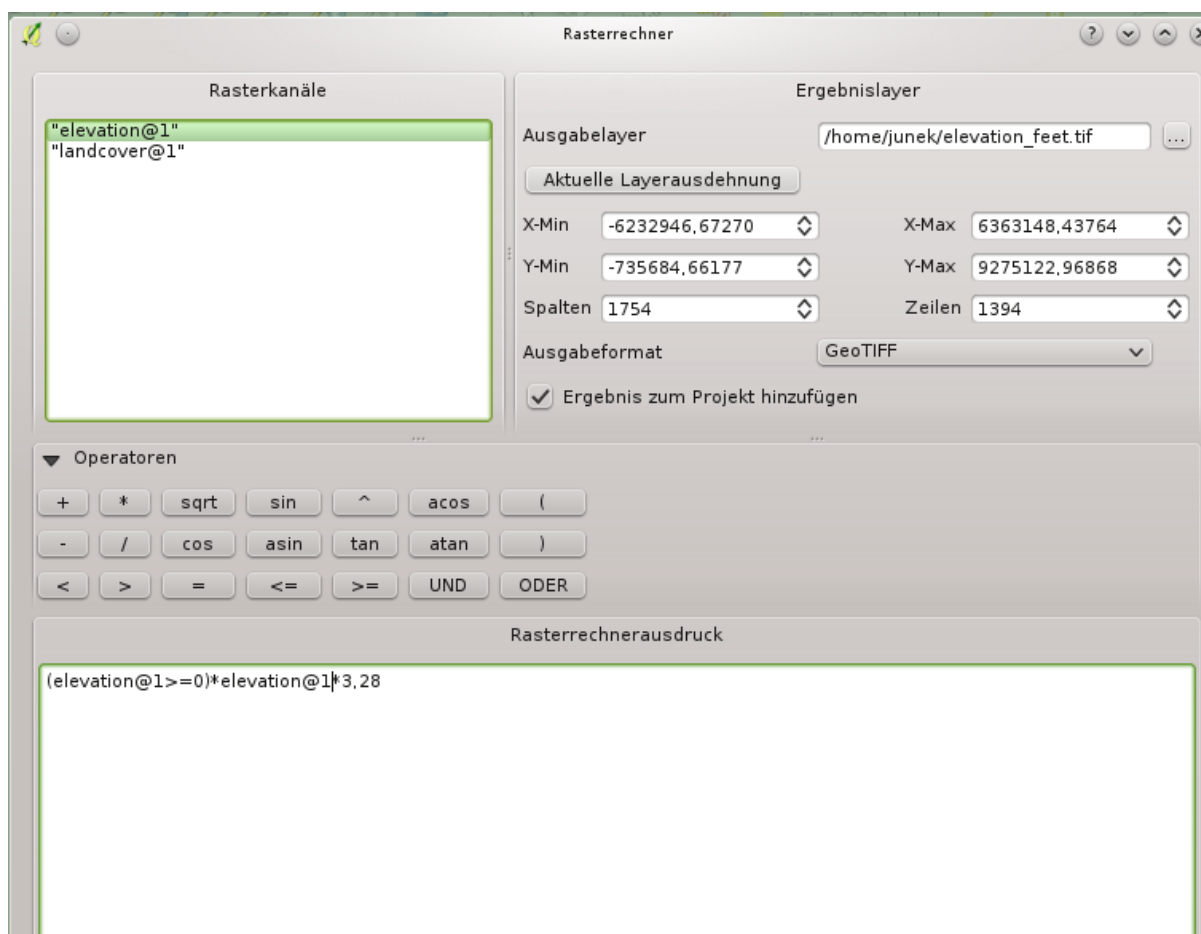




Abbildung 13.10: Rasterrechner 



Im Abschnitt **Ergebnislayer** müssen Sie einen Ausgabebereich definieren. Sie können dann den Analysebereich auf Grundlage eines Eingaberasters oder basierend auf Min/Max X und Y-Koordinaten bzw. mittels Spalten und Zeilen angeben, um die Auflösung des Ausgabebereich festzulegen. Wenn die Eingabebereich eine abweichende Auflösung besitzen, werden die Werte auf Basis des nearest neighbor Algorithmus resampelt.

Der Bereich **Operatoren** stellt Operatoren für die Berechnungen bereit. Um einen Operator auszuwählen, klicken Sie auf die entsprechende Schaltfläche. Es stehen mathematische Berechnungen (+, -, \*, ...) und trigonometrische Funktionen (sin, cos, tan, ...) zur Verfügung. Die Anzahl wird mit den nächsten Versionen sicherlich noch wachsen!

Mit dem Aktivieren des Kontrollkästchens  *Ergebnis zum Projekt hinzufügen* wird der Ausgabebereich automatisch der Legende hinzugefügt und kann somit visualisiert werden.

### 13.3.1 Beispiele

#### Höhenwerte von Meter zu Fuß konvertieren

Für das Erstellen eines Rasterlayers in Fuß aus einem Rasterlayer in Metern müssen Sie den Konvertierungsfaktor von Metern zu Fuß benutzen: 3.28. Der Ausdruck lautet:

```
"elevation@1" * 3.28
```

#### Eine Maske verwenden

Wenn Sie Teile des Rasterlayers ausmaskieren wollen, weil Sie nur an Höhenwerten über 0 Metern interessiert sind, können Sie den folgenden Ausdruck zum Erstellen einer Maske und zum gleichzeitigen Anwenden auf den Rasterlayer verwenden.

```
("elevation@1" >= 0) * "elevation@1"
```

Jede Zelle die größer oder gleich 0 ist wird auf den Wert 1 gesetzt, ansonsten lautet er 0. Dies erstellt die Maske 'on-the-fly'.

Wenn Sie ein Raster klassifizieren wollen, sagen wir beispielsweise in zwei Höhenklassen, können Sie die folgenden Ausdrücke verwenden um ein Raster mit zwei Werten 1 und 2 in einem Schritt erstellen.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

In anderen Worten stelle den Wert für jede Zelle mit einem Wert kleiner 50 auf 1 ein. Für jede Zelle mit einem Wert größer als oder gleich 50 stelle den Wert 2 ein.



---

## Arbeiten mit OGC Daten

---

### 14.1 QGIS als OGC Datenclient

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

Zur Beschreibung von geographischen Objekten in einem einfachen Datenmodell wurden von OGC 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 von QGIS unterstützte OGC Spezifikationen sind:

- **WMS** — Web Map Service (*WMS/WMTS Client*)
- **WMTS** — Web Map Tile Service (*WMS/WMTS Client*)
- **WFS** — Web Feature Service (*WFS und WFS-T Klient*)
- **WFS-T** — Web Feature Service - Transactional (*WFS und WFS-T Klient*)
- **WCS** — Web Coverage Service (*WCS Client*)
- **SFS** — Simple Features for SQL (*PostGIS Layer laden*)
- **GML** — Geography Markup Language

OGC-Dienste werden vermehrt zum Austausch von geographischen Daten zwischen unterschiedlichen GIS-Systemen und -implementierungen verwendet. QGIS unterstützt mittlerweile die oben genannten Spezifikationen als Client in Form von **SFS** (durch den Postgresql/PostGIS Datenprovider, vgl. *PostGIS Layer laden*).

#### 14.1.1 WMS/WMTS Client

##### 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.

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 mehr auf Übertragung ausgelegter Dienst (REST = Representational State Transfer) als ein voll funktionsfähiger Web Service. Daraus resultiert die Tatsache, dass die von QGIS generierte URL für das Bild auch in einem Browser eingesetzt werden kann. Das ist besonders hilfreich, wenn es beim Einsatz von WMS Probleme geben sollte. Da es sehr viele unterschiedliche 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.

## Überblick über die WMTS Unterstützung

QGIS kann auch als WMTS Client fungieren. WMTS ist ein OGC Standard zum Bereitstellen von Tile Sets von räumlichen Daten. Dies ist ein schnellerer und effizienterer Weg als die Bereitstellung über einen WMS weil bei WMTS die Tile Sets vorgeneriert werden und der Client nur die Übermittlung von Tiles, nicht Ihre Erstellung abfragt. Ein WMS beinhaltet typischerweise sowohl die Erstellung als auch die Übermittlung der Daten. Ein bekanntes Beispiel eines nicht-OGC Standards zum Darstellen von gekachelten räumlichen Daten ist Google Maps.

Um die Daten in einer Vielzahl von Maßstäben je nach Anforderung darzustellen werden die WMTS Tile Sets in mehreren verschiedenen Maßstäben erstellt und dem GIS Client zur Abfrage bereitgestellt.

Das Diagramm veranschaulicht das Konzept der Tile Sets:

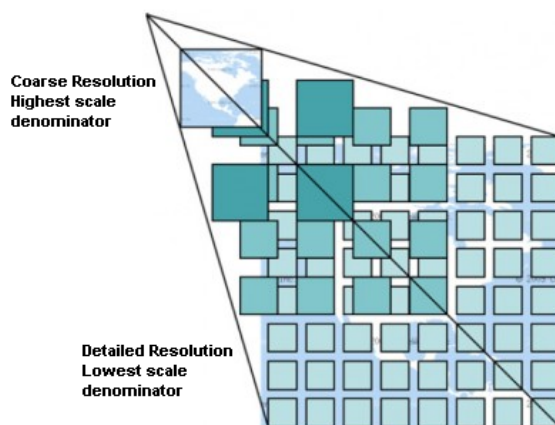


Abbildung 14.1: Konzept der WMTS Tile Sets

Die zwei Typen von WMTS Schnittstellen die QGIS unterstützt sind die über Key-Value-Pairs (KVP) und RESTful. Diese zwei Schnittstellen sind unterschiedlich und Sie müssen diese für QGIS unterschiedlich spezifizieren.

1) In order to access a **WMTS KVP** service, a QGIS user must open the WMS/WMTS interface and add the following string to the URL of the WMTS tile service:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

Ein Beispiel für diesen Typ von Adresse ist

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?\  
service=WMTS&request=GetCapabilities
```

Zu Testzwecken funktioniert der topo2 Layer in diesem WMTS gut. Indem man diese Zeichenfolge hinzufügt gibt man an dass ein WMTS Web Service anstatt eines WMS Service benutzt werden soll.

2. Der **RESTful WMTS** Service erfordert eine andere Form, eine einfache URL. Das von der OGC empfohlene Format ist:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```


Dieses Format hilft Ihnen zu erkennen dass dies eine RESTful Adresse ist. Eine RESTful WMTS Verbindung wird in QGIS einfach durch das Verlinken seiner Adresse im WMS Setup im URL Feld des Formulars erstellt. Ein Beispiel für diesen Adresstyp im Fall der Österreichischen Grundkarte ist <http://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml>.

**Bemerkung:** Sie können immer noch alte Services die WMS-C genannt werden finden. Diese Services sind denen von WMTS ähnlich (z.B. der gleiche Zweck der dann ein bisschen anders funktioniert). Sie können Sie genauso wie WMTS Services verwalten. Geben Sie einfach `?tiled=true` am Ende der URL ein. Schlagen Sie unter [http://wiki.osgeo.org/wiki/Tile\\_Map\\_Service\\_Specification](http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification) für weitere Informationen zu diese Spezifikation nach.

Wenn Sie WMTS lesen können Sie auch an WMS-C denken.

## WMS/WMTS Server auswählen


Wenn Sie das WMS Feature das erste Mal in QGIS verwenden sind keine Server definiert.

Fangen Sie an indem Sie den  WMS/WMTS-Layer hinzufügen Knopf in der Werkzeugleiste klicken oder indem Sie *Layer* → *WMS/WMTS-Layer hinzufügen ...* wählen.

Der Dialog *Layer eines WM(T)S-Servers hinzufügen* erscheint dann. Sie können zum Ausprobieren aber einige vordefinierte Server hinzufügen, indem Sie auf den Knopf **[Vorgegebene Server ergänzen]** klicken. Dadurch werden Ihnen zwei WMS Demo Server zur Verfügung gestellt: die WMS Server der DM Solutions Group und von Lizardtech. Um im Reiter *Layer* einen neuen WMS Server zu definieren wählen Sie **[Neu]**. Geben Sie dann die Parameter die zur Verbindung benötigt werden, ein, so wie es in [table\\_OGC\\_1](#) aufgelistet ist:

Name	Ein Name für diese Verbindung. Dieser Name wird in der Serververbindungs Dropdown Box verwendet und kann diese dann von anderen WMS Servern unterscheiden.
URL	URL des Servers der die Daten bereitstellt. Dies muss ein auflösbarer Hostname sein- das gleiche Format das Sie auch benutzen würden um eine Telnetverbindung zu öffnen oder einen Ping an einen Host zu senden.
Benutzername	Benutzername um einen abgesicherten WMS Server anzubinden. Dieser Parameter ist optional.
Passwort	Passwort für einen grundlegend authentifizierten WMS Server. Dieser Parameter ist optional.
Gemeldete GetMap-URI aus Diensteigenschaften ignorieren	<input checked="" type="checkbox"/> <i>Gemeldete GetMap/GetTile-URI aus Diensteigenschaften ignorieren.</i> Verwenden Sie die vorgegebene URI aus dem URL Feld oben.
Gemeldete GetFeatureInfo-URI ignorieren	<input checked="" type="checkbox"/> <i>Gemeldete GetFeatureInfo-URI ignorieren.</i> Verwenden Sie die vorgegebene URI aus dem URL Feld oben.

Tabelle OGC 1: WMS Verbindungs-Parameter

Wenn Sie einen Proxyserver, um WMS Services aus dem Internet empfangen zu können, aufsetzen müssen können Sie Ihren Proxyserver in den Optionen hinzufügen. Wählen Sie *Einstellungen* → *Optionen* und klicken Sie auf das Menü *Netzwerk*. Dort können sie Ihre Proxyeinstellungen hinzufügen und diese aktivieren indem Sie das Kontrollkästchen  *Proxy für Webzugriff benutzen* aktivieren. Vergewissern Sie sich dass Sie den richtigen Proxytyp aus dem *Proxytyp*  Dropdownmenü ausgewählt haben

Ist erst einmal eine WMS Serververbindung erstellt worden bleibt dieser für kommende QGIS Sitzungen erhalten.

### Tipp: WMS-Server-URLs

Vergewissern Sie sich beim Eingeben der WMS Server URL dass Sie nur die einfache URL eingeben. Zum Beispiel sollten keine Fragmente wie `request=GetCapabilities` oder `version=1.0.0` enthalten sein.

## WMS/WMTS Layer laden

Nachdem Sie erfolgreich Ihre Parameter eingetragen haben können Sie den [**Verbinden**] Knopf zum abrufen der Fähigkeiten des ausgewählten Servers verwenden. Dies beinhaltet die Bildkodierung, Layerstile und Projektionen. Da dies eine Netzwerkoperation ist hängt die Schnelligkeit der Antwort von der Qualität Ihrer Netzwerkverbindung zum WMS Server ab. Während Daten vom WMS Server heruntergeladen werden wird der Downloadfortschritt links unten im WMS Dialog visualisiert.

Ihr Bildschirm sollte in etwa so wie der in Abbildung [figure\\_OGR\\_1](#) aussehen wie die Ausgabe durch den DM Solutions Group Server.

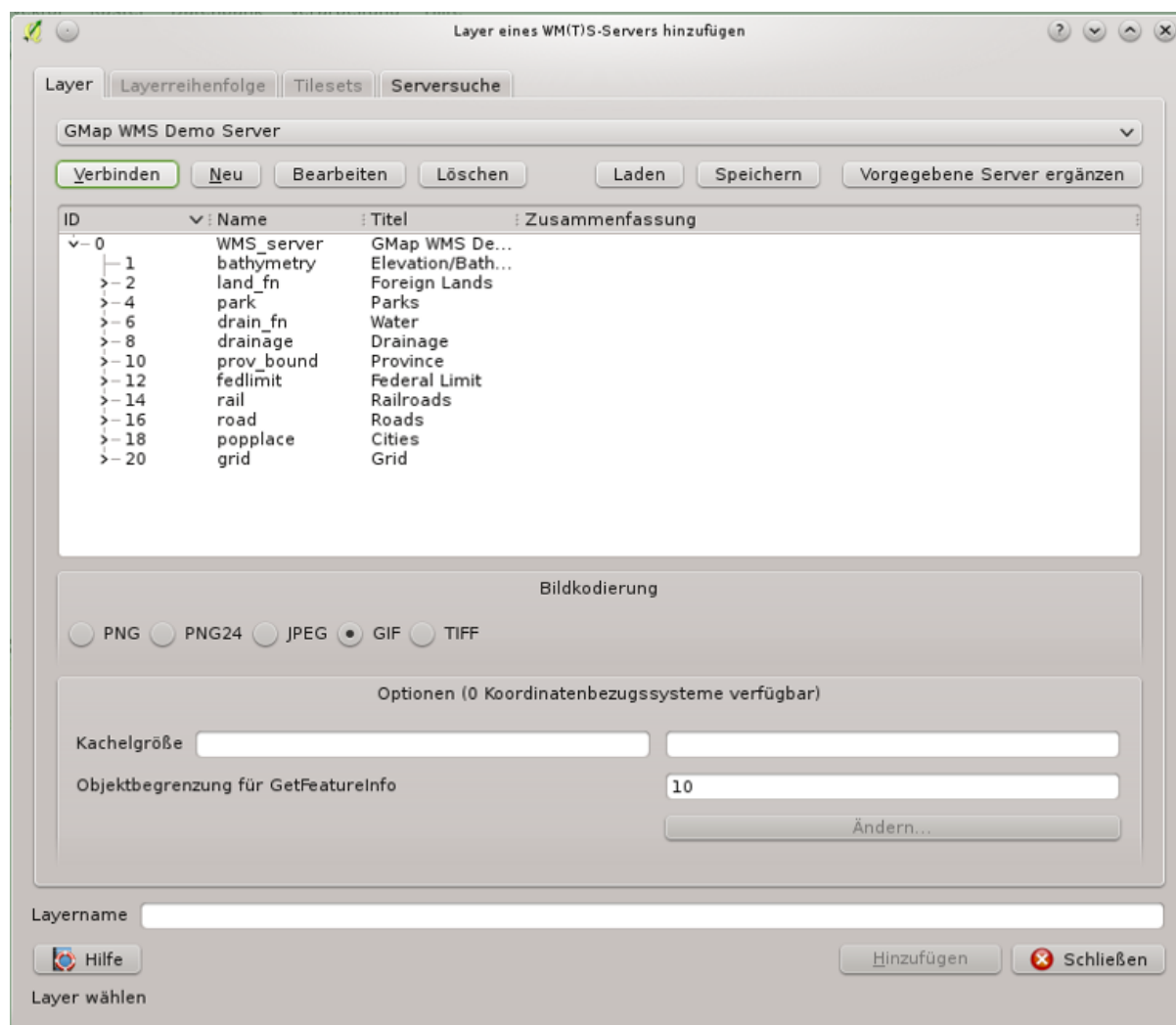


Abbildung 14.2: Dialog zum Hinzufügen eines WMS Servers bei dem die erhältlichen Layer gezeigt werden 

### Bildkodierung

Der *Bildkodierung* Bereich listet die Formate die sowohl vom Client als auch vom Server unterstützt werden auf. Wählen Sie eines abhängig von Ihren Anforderungen an die Bildqualität aus.

#### Tipp: 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.

Verwenden Sie JPEG wenn Sie damit rechnen das die WMS Daten photographischen Charakter haben und/oder Sie ein Verlust an Bildqualität nicht stört. Dieser Kompromiss reduziert typischerweise die Datentransferbedingungen um das fünffache verglichen mit PNG.

Verwenden Sie PNG wenn Sie eine genaue Wiedergabe der Originaldaten erzielen wollen und die erhöhten Datentransferbedingungen Sie nicht stören.

### Optionen

Der Optionen Bereich des Dialogs stellt ein Textfeld zur Verfügung in das Sie einen *Layernamen* für den WMS Layer hinzufügen können. Dieser Name wird nach dem Laden in der Legende erscheinen.

Unter dem Layernamen können Sie wenn Sie den WMS Request in mehrere Requests aufsplitten wollen die *Kachelgröße* (z.B. 256x256) definieren.

Die *Objektbegrenzung für GetFeatureInfo* legt fest welche Attributspalten vom Server abgefragt werden.

Wenn Sie einen WMS aus der Liste wählen erscheint ein Feld mit der Standardprojektion, die vom Mapserver bereitgestellt wird. Ist der [Ändern ...] Knopf aktiv können Sie darauf klicken und die Standardprojektion des WMS in ein anderes vom WMS Server bereitgestelltes KBS ändern.

### Layerreihenfolge

Der Reiter *Layerreihenfolge* listet die vom gerade verbundenen WMS Server ausgewählten Layer auf. Sie stellen vielleicht fest dass einige Layer ausklappbar sind. Das bedeutet dass der Layer in einer Auswahl von Bildstilen dargestellt werden kann.

Sie können mehrere Layer auf einmal auswählen aber nur einen Bildstil pro Layer. Wenn mehrere Layer ausgewählt sind werden Sie am WMS Server kombiniert und in einem Rutsch an QGIS weitergegeben.

### Tipp: WMS Layer anordnen

Von einem Server dargestellte WMS Layer werden in der Reihenfolge aus dem Abschnitt Layers von oben bis unten überlagert. Wenn Sie die Layerreihenfolge ändern wollen können Sie den Reiter *Layerreihenfolge* benutzen.

### Transparenz

In dieser Version von QGIS ist die Einstellung *Globale Transparenz* aus dem Menü *Layerereigenschaften* immer gemäß den Layereigenschaften eingestellt.

### Tipp: Transparenz von WMS-Layern

Die WMS Bildtransparenz steht Ihnen abhängig von der Bildkodierung zur Verfügung: PNG und GIF unterstützen Transparenz währenddessen JPEG keine Unterstützung bietet.

### Koordinatenbezugssystem

Koordinatenbezugssystem (KBS) ist die Bezeichnung des OGC für eine Projektion in QGIS.

Jeder WMS Layer kann abhängig von den Fähigkeiten des WMS in mehreren KBS dargestellt werden.

Um ein KBS auszuwählen wählen Sie [Ändern ...] und ein Dialog ähnlich wie Figure Projection 3 aus *Arbeiten mit Projektionen* erscheint. Der hauptsächliche Unterschied der WMS-Version ist dass nur die KBS, die vom WMS Server unterstützt werden, gezeigt werden.

### Serversuche

In QGIS können Sie WMS-Server suchen. [Figure\\_OGC\\_2](#) zeigt den Reiter *Serversuche* mit dem Dialog *Layer eines WM(T)S-Servers hinzufügen*.

Wie Sie sehen können ist es möglich ein Suchwort in das Textfeld einzugeben und den [Suchen] Knopf zu betätigen. Nach kurzer Zeit werden dann die Ergebnisse aufgelistet. Gehen Sie die Ergebnisliste durch und überprüfen Sie die Suchergebnisse in der Tabelle. Um ein Ergebnis zu visualisieren suchen Sie einen Tabelleneintrag aus, drücken Sie auf den [Gewählte Zeile der WMS-Liste hinzufügen] Knopf und gehen Sie zurück zum Reiter *Layer*. QGIS hat Ihre Serverliste automatisch aktualisiert und das ausgewählte Suchergebnis ist bereits ausführbar in der Liste der gespeicherten WMS-Server im Reiter *Layer* gemacht. Sie brauchen die Layerliste nur mit dem Knopf [Verbinden] abfragen. Diese Option ist sehr praktisch wenn Sie Karten anhand bestimmter Schlüsselwörter suchen wollen.

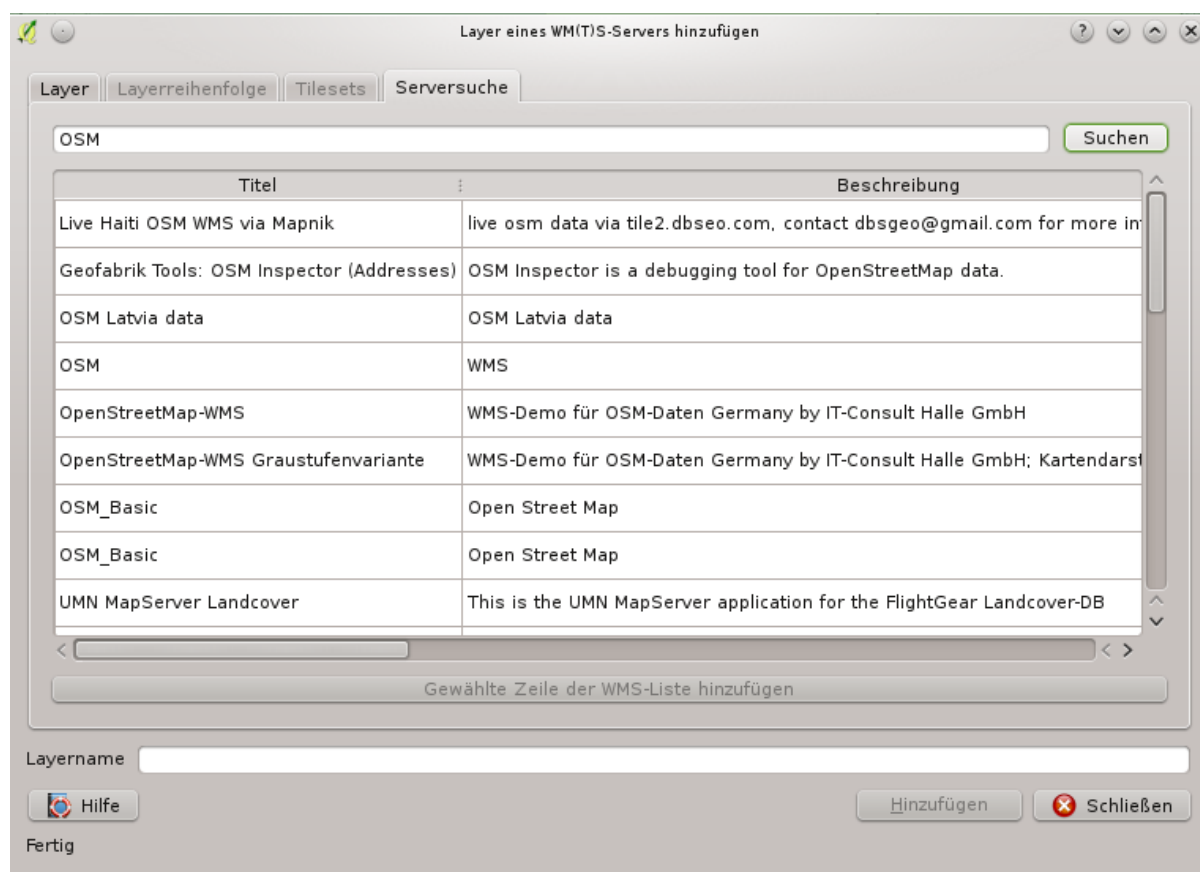



Abbildung 14.3: Dialog zum Suchen von WMS Servern nach einigen Stichwörtern 

Diese Suchfunktion ist ein Frontend zur API von <http://geopole.org>.


## Tilesets

Wenn Sie WMTS (gecachte WMS) Dienste verwenden wie

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?
service=WMTS&request=GetCapabilities
```


sind Sie in der Lage den *Tilesets* Reiter, der vom Server bereitgestellt wird, zu durchsuchen. Zusätzliche Informationen wie Kachelgröße, Formate und unterstützte KBS werden in der Tabelle aufgelistet. In Kombination mit diesen Funktionalitäten können Sie den Kachelmaßstabschieberegler verwenden indem Sie *Einstellungen* → *Bedienfelder* (KDE und Windows) oder *Ansicht* → *Bedienfelder* (Gnome und MacOSX) auswählen und dann 'Kachelmaßstab' aussuchen. Dadurch werden die erhältlichen Maßstäbe vom Kachelserver in einem netten ange-dockten Schieberegler zugänglich gemacht.

## Das Objekte abfragen Werkzeug

Nachdem Sie einen Layer von einem WMS-Server geladen haben, können Sie die Layer mit dem Werkzeug  *Objekte Abfragen* abfragen, sofern der WMS-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. **Formatauswahl**

Wenn durch den Server mehrere Formate unterstützt werden wird dem Objekte abfragen Dialog automatisch eine Kombobox mit den unterstützten Formaten hinzugefügt und das ausgewählte Format kann im Projekt für den Layer gespeichert werden. **GML Formatunterstützung**



Das  **Objekte abfragen** Werkzeug unterstützt WMS Server Response (GetFeatureInfo) im GML Format (es wird Feature in der QGIS GUI in diesem Zusammenhang genannt). Wenn das "Feature" Format vom Server unterstützt wird und ausgewählt ist, sind die Ergebnisse des Objekte abfragen Werkzeugs Vektorobjekte, wie bei einem normalen Vektorlayer. Wenn ein einzelnes Objekt im Baum ausgewählt wird wird es in der Karte hervorgehoben und kann in die Zwischenablage kopiert werden und in einen anderen Vektorlayer eingefügt werden. Sehen Sie sich für die Unterstützung von GetFeatureInfo im GML Format das Beispielsetup des UMN Mapservers unten an.

```
# in layer METADATA add which fields should be included and define geometry (example):

"gml_include_items"    "all"
"ows_geometries"      "mygeom"
"ows_mygeom_type"     "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is send as multipart and contains XSD)
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
  NAME "OGRGML"
  MIMETYPE "ogr/gml"
  DRIVER "OGR/GML"
  FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

## Eigenschaften

Nachdem Sie einen WMS Server hinzugefügt haben können Sie sich seine Eigenschaften mit einem Rechtsklick in der Legende und dem Auswählen von *Eigenschaften* ansehen. **Reiter Metadaten**

Der Reiter *Metadaten* im Kontextmenü zeigt eine Vielzahl von Informationen über den WMS-Server. Diese Infos sind dem Capabilities-Dokument des Servers entnommen. Viele Definitionen können reduziert werden indem man den WMS Standard liest (siehe OPEN-GEOSPATIAL-CONSORTIUM *Literatur und Internetreferenzen*), hier sind dazu einige praktische Definitionen:

- **Servereigenschaften**

- **WMS Version**– Die WMS-Version, die vom Server unterstützt wird.
- **Bildformate** - Die Liste MIME-Typen mit denen der Server reagieren kann wenn er die Karte zeichnet. QGIS unterstützt die Formate mit denen die Qt Bibliotheken gebaut worden sind, was typischerweise wenigstens image/png und image/jpeg sind.
- **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 herangezoomt werden** - Ob der Layer durch den Server herangezoomt werden kann oder nicht. Diese Version von QGIS nimmt an, dass bei allen WMS Layern Ja eingestellt ist. Mangelhafte Layer

werden seltsam dargestellt.

- **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.
- **Feste Breite, Fest Höhe** - Ob dieser Layer festgelegte Quellpixeldimensionen besitzt. Diese Version von QGIS nimmt an dass bei allen WMS Layern nichts eingestellt ist. Mangelhafte Layer werden seltsam dargestellt.
- **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), so dass 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.

### Zeigen der WMS Legenden grafik im Inhaltsverzeichnis und in der Druckzusammenstellung

Der QGIS WMS Datenprovider ist in der Lage eine Legenden grafik in der Inhaltsverzeichnisliste und in der Druckzusammenstellung darzustellen. Die WMS Legende wird nur gezeigt wenn der WMS Server eine `GetLegendGraphic` Capability hat und für den Layer die `getCapability` URL angegeben wurde, also müssen Sie zusätzlich einen Stil für den Layer aussuchen.

Wenn eine `legendGraphic` verfügbar ist, wird diese unter dem Layer angezeigt. Sie ist klein und Sie müssen darauf klicken um Sie in Ihrer tatsächlichen Größe (gemäß den `QgsLegendInterface` Architekturlimitationen) zu öffnen. Das Klicken auf die Legende des Layers öffnet einen Rahmen mit einer Legende in voller Auflösung.


In der Druckzusammenstellung wird die Legende in ihren (gedownloadeten) Originalmaßen integriert. Die Auflösung der Legenden grafik kann in dem Elementeigenschaften unter Legende -> `WMS LegendGraphic` eingestellt werden um Ihren Druckanforderungen zu entsprechen

Die Legende wird Kontextinformationen basiert auf dem aktuellen Maßstab darstellen. Die WMS Legende wird nur gezeigt wenn der WMS Server eine `GetLegendGraphic` Capability hat und für dem Layer eine `getCapability` URL angegeben wurde, also müssen Sie einen Stil auswählen.

### Einschränkungen des WMS-Klienten

Nicht alle mögliche WMS Client Funktionalitäten sind in diese Version von QGIS integriert worden. Einige der bemerkenswerteren Ausnahmen folgen noch.

#### WMS-Layereigenschaften ändern

Nachdem Sie die  WMS/WMTS-Layer hinzufügen Prozedur beendet haben gibt es keine Möglichkeit mehr die Einstellungen zu ändern. Sie können das Problem umgehen indem Sie den Layer komplett entfernen und neu beginnen.

#### WMS-Server, die eine Authentifizierung benötigen

Derzeit werden öffentlich zugängliche und gesicherte WMS Services unterstützt. Die gesicherten WMS Server können mit öffentlicher Authentifizierung angebunden werden. Sie können die (optionalen) Anmeldeinformationen hinzufügen wenn Sie einen WMS Server hinzufügen. Schlagen Sie unter Abschnitt [WMS/WMTS Server auswählen](#) Details nach.

---

#### Tipp: Zugriff auf abgesicherte OGC-Layer

Wenn Sie Zugriff auf OGC-Layer benötigen, die anders als durch einfache, öffentliche Authentifizierung abgesichert sind, können Sie `InteProxy` als transparenten Proxy verwenden. Dieser unterstützt verschiedene

Methoden der Authentifizierung. Weitere Informationen zu diesem Thema finden Sie auf der Webseite <http://inteproxy.wald.intevation.org>.

---

**Tipp: QGIS WMS Mapserver**

Seit Version 1.7.0 besitzt QGIS seine eigene Implementierung eines WMS 1.3.0 Mapservers. Lesen mehr darüber in Kapitel *QGIS als OGC Datenserver*.

---

## 14.1.2 WCS Client



Ein Web Coverage Service (WCS) stellt eine Anbindung zu Rasterdaten in Formaten die nützlich für die client-seitige Darstellung, als Input für wissenschaftliche Modelle und für andere Clients zur Verfügung. Der WCS ist vergleichbar zu WFS und WMS. Als WMS und WFS Service Instanz erlaubt der WCS den Clients Teile von Serverinformationsbeständen basierend auf räumlichen Einschränkungen und Abfragekriterien auszuwählen.

QGIS hat einen nativen WCS Provider und unterstützt sowohl Version 1.0 und 1.1 (welche deutliche Unterschiede aufweisen), aktuell jedoch wird 1.0 vorgezogen da es mit 1.1 viele Probleme gibt (z.B. implementiert jeder Server es auf eine andere Art und Weise mit zahlreichen Besonderheiten).

Der native WCS Provider kümmert sich um alle Netzwerkanfragen und verwendet alle voreingestellten QGIS Netzwerkanfragen (insbesondere proxy). Es ist auch möglich einen Cache auszuwählen ('Immer cachen', 'Cache vorziehen', 'Netzwerk vorziehen', 'Immer Netzwerk') und der Provider unterstützt ebenfalls das Auswählen der Zeitposition wenn Temporal Domain vom Server angeboten wird.



## 14.1.3 WFS und WFS-T Klient

In QGIS verhält sich ein WFS Layer ziemlich genau so wie jeder andere Vektorlayer. Sie können Objekte abfragen und auswählen und sich die Attributtabelle anschauen.. Seit QGIS 1.6 wird auch das Bearbeiten von WFS-T unterstützt.

Im Allgemeinen verhält sich das Hinzufügen eines WFS Layers sehr ähnlich wie die Vorgehensweise die beim WMS verwendet wird. Der Unterschied besteht darin dass keine voreingestellten Server definiert sind, also müssen wir eigene hinzufügen.

### Einen WFS-Layer laden

In diesem Beispiel verwenden wir den WFS-Server der Firma DMSolutions und laden einen Layer. Die URL ist: [http://www2.dmsolutions.ca/cgi-bin/mswfs\\_gmap](http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap)

1. Klicken Sie auf das  WFS-Layer hinzufügen Werkzeug in der Werkzeugeiste. Der Dialog *WFS-Layer des Servers hinzufügen* erscheint.
2. Klicken Sie auf **[Neu]**.
3. Geben Sie 'DM Solutions' als Namen ein.
4. Geben Sie die URL ein (siehe oben).
5. Klicken Sie **[OK]**.
6. Wählen Sie 'DM Solutions' aus der *Serververbindungen*  Dropdownliste.
7. Klicken Sie **[Verbinden]**.
8. Warten Sie bis die Layerliste aufgefüllt ist.
9. Wählen Sie den *Parks* Layer aus der Liste.
10. Klicken Sie **[Hinzufügen]** um den Layer zur Karte hinzuzufügen.

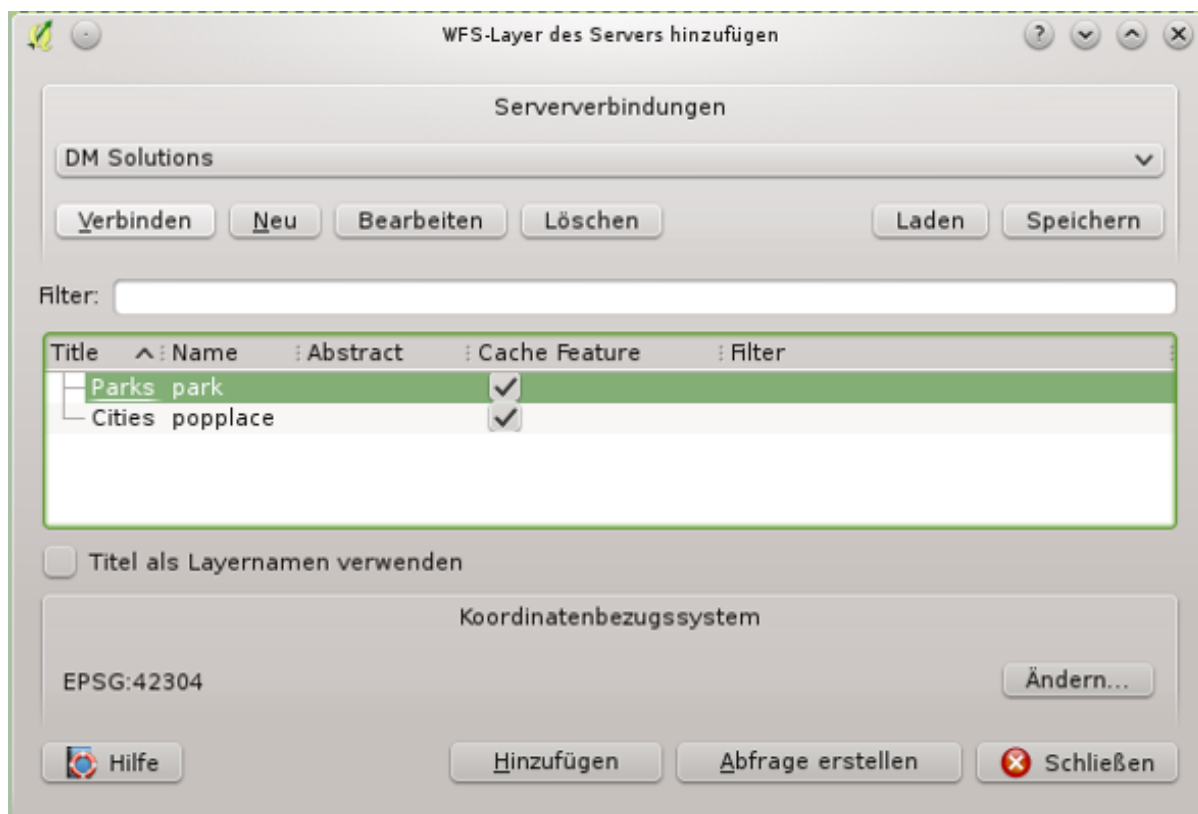


Abbildung 14.4: Einen WFS Layer hinzufügen 

Beachten Sie dass auch jede Proxyeinstellung die Sie in Ihren eingestellt haben berücksichtigt wird.

Sie werden feststellen dass der Downloadprozess links unten im QGIS Hauptfenster visualisiert wird. Nachdem der Layer geladen ist können Sie eine oder zwei Provinzen abfragen und auswählen und sich die Attributtabelle anschauen.

Es wird nur WFS 1.0.0 unterstützt. Bis zu diesem Zeitpunkt wurde noch nicht sehr umfangreich die Anbindung mit anderen WFS Servern getestet. Wenn Sie Probleme feststellen, zögern Sie bitte nicht, eine Email an das QGIS Projekt zu schicken oder einen Fehlerreport zu schreiben. Sie finden eine Liste möglicher Kontakte in Kapitel *Hilfe und Support*.

---

#### **Tipp: WFS-Server finden**

Sie können weitere 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.

---

## 14.2 QGIS als OGC Datenserver

QGIS Server ist eine open Source WMS 1.3, WFS 1.0.0 und WCS 1.1.1 Umsetzung die zusätzlich erweiterte kartografische Funktionen für thematische Karten zur Verfügung stellt. Der QGIS Server ist eine FastCGI/CGI (Common Gateway Interface)-Anwendung, ist in C++ geschrieben und arbeitet mit einem Webserver zusammen (z.B. Apache oder Lighttpd). Es wurde mit finanziellen Mitteln des EU Projekts Orchestra, Sany und der Stadt Uster aus der Schweiz entwickelt.

QGIS Server verwendet QGIS im Hintergrund für die GIS-Logik und für das Darstellen der Karte. Des weiteren wird die Qt-Bibliothek für Grafiken und für die plattformunabhängige C++-Programmierung verwendet.

Im Gegensatz zu anderer WMS-Software verwendet der QGIS Server kartographische Regeln als Konfigurationssprache, sowohl für die Server Konfiguration als auch für die benutzerdefinierten kartographischen Regeln.

Da QGIS Desktop und QGIS Server die gleichen Visualisierungsbibliotheken verwenden sehen die Karten, die im Internet veröffentlicht werden genauso wie im Desktop GIS aus.

In einer der folgenden Anleitungen werden wir eine Beispielkonfiguration zum Aufsetzen eines QGIS Servers anbieten. Vorerst empfehlen wir eine der folgenden URLs zu lesen um mehr Informationen zu bekommen:

- [http://karlinapp.ethz.ch/qgis\\_wms/](http://karlinapp.ethz.ch/qgis_wms/)
- [http://hub.qgis.org/projects/quantum-gis/wiki/QGIS\\_Server\\_Tutorial](http://hub.qgis.org/projects/quantum-gis/wiki/QGIS_Server_Tutorial)
- <http://linfiniti.com/2010/08/qgis-mapserver-a-wms-server-for-the-masses/>

## 14.2.1 Beispielinstallation unter Debian Squeeze

An dieser Stelle zeigen wir Ihnen eine kurze und einfache Beispielinstallation für Debian Squeeze. Auch viele andere Betriebssysteme bieten Pakete für QGIS Server. Wenn Sie alles aus dem Quellcode bauen müssen, sehen in den oben angegebenen URLs nach.

Abgesehen von QGIS Desktop und QGIS Server benötigen Sie einen Webserver, in unserem Fall apache2. Sie können alle Pakete mit 'aptitude' oder 'apt-get install' zusammen installieren. Alle weiteren notwendigen Pakete werden dann mitinstalliert. Nach der Installation sollten Sie testen, ob der Webserver und der QGIS Server wie erwartet funktionieren. Vergewissern Sie sich, dass der Apache Webserver läuft. Ansonsten starten Sie ihn mit '/etc/init.d/apache2 start'. Öffnen Sie nun einen Web-Browser und geben Sie die URL <http://localhost> ein. Wenn alles in Ordnung ist, sollten Sie die Meldung 'It works!' sehen.


Jetzt testen wir die QGIS Server Installation. Die Datei `qgis_mapserv.fcgi` finden Sie unter `/usr/lib/cgi-bin/qgis_mapserv.fcgi` und stellt einen Test-WMS bereit, der die Grenzen von Alaska zeigt. Laden Sie den WMS mit der URL [http://localhost/cgi-bin/qgis\\_mapserv.fcgi](http://localhost/cgi-bin/qgis_mapserv.fcgi) wie in Abschnitt *WMS/WMTS Layer laden* beschrieben.

## 14.2.2 Erstellen eines WMS/WFS/WCS aus einem QGIS Projekt

Um einen neuen QGIS WMS, WFS oder WCS Server zur Verfügung zu stellen, müssen wir eine QGIS Projektdatei mit einigen Daten erstellen. Hier benutzen wir die 'Alaska' Shapedatei aus dem QGIS Beispieldatensatz. Definieren Sie die Farben und den Stil der Layer in QGIS und auch das Projekt KBS, falls die nicht schon definiert ist.

Gehen Sie dann zu dem *OWS Server Menü* des *Einstellung* → *Projekteigenschaften* Dialog und geben Sie einige Informationen über den OWS in den Feldern unter *Diensteigenschaften* an. Diese werden dann in der GetCapabilities Antwort des WMS, WFS oder WCS auftauchen. Wenn Sie  *Diensteigenschaften* nicht aktivieren wird der QGIS Server die Informationen aus der `wms_metadata.xml` Datei, die im `cgi-bin` Ordner gespeichert ist, verwenden.

### WMS-Capabilities

Im *WMS-Capabilities* Bereich können Sie die von der WMS GetCapabilities Response angezeigten Ausmaße definieren indem Sie die minimalen und maximalen X und Y Werte in die Felder unter *Angezeigte Ausmasse* eingeben. Das Klicken von *Aktuelle Anzeigegrenzen übernehmen* stellt diese Werte auf die Ausmaße die gerade im QGIS Kartenfenster dargestellt werden ein. Wenn Sie  *KBS-Beschränkungen* aktivieren können Sie einschränken in welchem Koordinatenbezugssystem (KBS) QGIS Server Karten zum Darstellen anbieten wird. Verwenden Sie den  Knopf um diese KBS aus der Koordinatenbezugssystem-Auswahl auszuwählen oder klicken Sie *Benutzte* um das im QGIS Projekt benutzte KBS der Liste hinzuzufügen.

Wenn Sie Druckzusammenstellungen in Ihrem Projekt definiert haben werden diese in der GetCapabilities Response aufgelistet und sie können vom GetPrint Request verwendet werden um einen Druck zu erstellen, wobei eines der Druckzusammenstellungslayouts als Vorlage verwendet wird. Dies ist eine QGIS-spezifische Erweiterung zur WMS 1.3.0 Spezifikation. Wenn Sie die Veröffentlichung jedweder Druckzusammenstellung durch

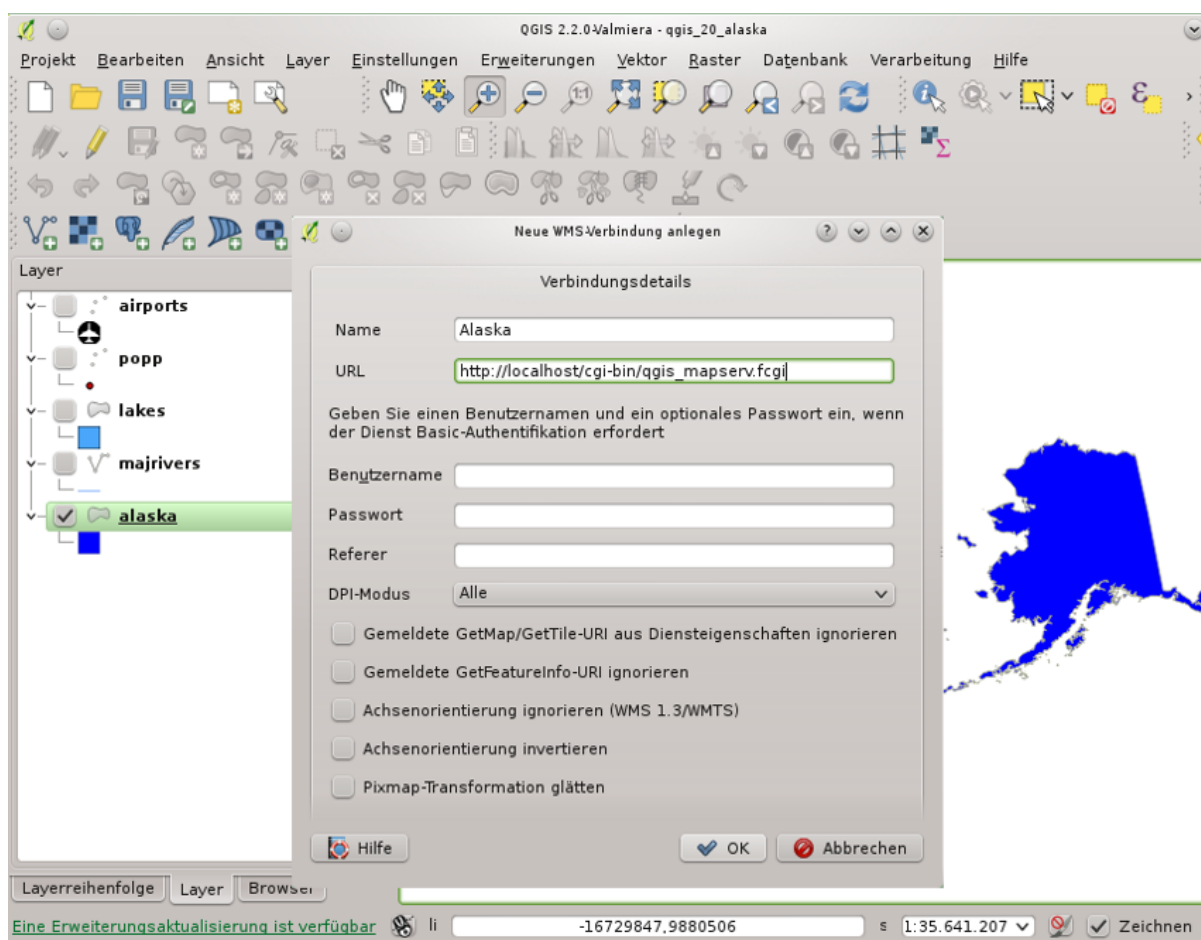


Abbildung 14.5: Standard WMS mit USA boundaries im QGIS Server (KDE) 

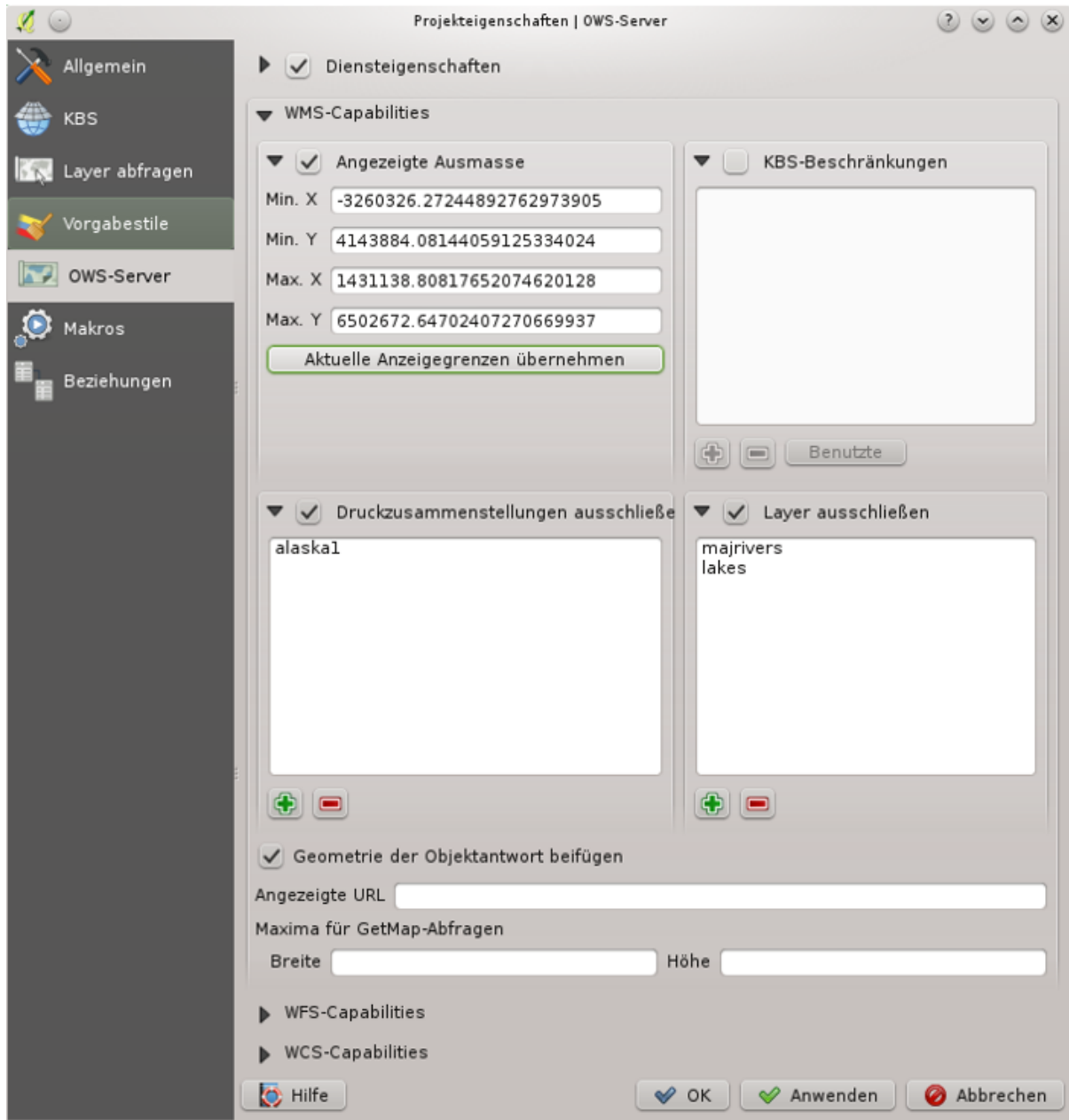






Abbildung 14.6: Definitionen für ein QGIS WMS/WFS/WCS Server Projekt (KDE)



den WMS ausschließen wollen aktivieren Sie  *Druckzusammenstellungen ausschließen* und klicken Sie unten den  Knopf. Wählen Sie dann eine Druckzusammenstellung aus dem *Druckzusammenstellung wählen* Dialog um diese der Ausnahmeliste für die Zusammenstellungen hinzuzufügen.

Wenn Sie einen Layer oder eine Layergruppe davon ausschließen wollen vom WMS veröffentlicht zu werden aktivieren Sie  *Layer ausschließen* und klicken Sie auf den *!ActionSignPlus!* Knopf darunter. Dies öffnet den *:guilabel: 'Eingeschränkte Layer und Gruppen wählen* Dialog, der es Ihnen ermöglicht die Layer und Gruppen die nicht veröffentlicht werden sollen auszusuchen. Verwenden Sie den *Umschalt* oder *Strg* Knopf wenn Sie mehrere Einträge auf einmal auswählen wollen.

Sie können requestete GetFeatureInfo als Klartext, XML und GML empfangen. Standard ist XML, das Text- oder GML-Format hängt vom verwendeten Ausgabeformat für den GetFeatureInfo Request ab.

Wenn Sie es wünschen können Sie das  *Geometrie der Objektantwort beifügen* Kontrollkästchen aktivieren. Dies wird die Geometrien der Objekte der GetFeatureInfo Response im Textformat hinzufügen. Wenn Sie wollen dass QGIS Server spezifische Request URLs in der WMS GetCapabilities Response angibt, geben sie die entsprechende URL im *Angezeigte URL* Feld an. Darüberhinaus können Sie die Maximalgröße der Karten, die vom GetMap Request wiedergegeben wird, durch das Eingeben der Maximalen Breite und Höhe in die entsprechenden Felder unter *Maxima für GetMap-Abfragen* einschränken.

Wenn einer Ihrer Layer Kartentippanzeige verwendet (z.B. um Text anhand von Ausdrücken zu zeigen) wird dies innerhalb der GetFeatureInfo Ausgabe aufgelistet. Wenn der Layer eine Wertabbildung für eines seiner Attribute verwendet wird diese Information ebenfalls in der GetFeatureInfo Ausgabe gezeigt.

### WFS-Capabilities

Im *WFS-Capabilities* Bereich können Sie die Layer auswählen die Sie als WFS veröffentlichen wollen und festlegen ob Sie die Update, Insert und Delete Operationen erlauben. Wenn Sie eine URL in das *Angezeigte URL* Feld des *WFS-Capabilities* Bereichs eingeben, wird QGIS Server diese bestimmte URL in der WFS GetCapabilities Response angeben.

### WCS-Capabilities

Im *WCS-Capabilities* Bereich können Sie die Layer auswählen die Sie als WCS veröffentlichen wollen. Wenn Sie eine URL in das *Angezeigte URL* Feld des *WFS-Capabilities* Bereichs eingeben, wird QGIS Server diese bestimmte URL in der WCS GetCapabilities Response angeben.

Speichern Sie die Sitzung jetzt in eine Projektdatei `alaska.qgs`. Um das Projekt als WMS/WFS bereitzustellen erstellen wir einen neuen Ordner `/usr/lib/cgi-bin/project` mit Verwaltungsrechten und fügen eine Projektdatei `alaska.qgs` und eine Kopie der `qgis_mapserv.fcgi` Datei hinzu - das ist alles.

Jetzt testen wir unser WMS, WFS und WCS Projekt. Fügen Sie QGIS den WMS, WFS und WCS wie in *WMS/WMTS Layer laden*, *WFS und WFS-T Klient* und *WCS Client* beschrieben zu und laden Sie die Daten. Die URL ist:

```
http://localhost/cgi-bin/project/qgis_mapserv.fcgi
```

### Den OWS feinabstimmen

Bei Vektorlayern ermöglicht Ihnen das *Felder* Menü des *Layer* → *Eigenschaften* Dialogs für jedes Attribut zu definieren ob es veröffentlicht wird oder nicht. Standardmäßig werden alle Attribute von Ihrem WMS und WFS veröffentlicht. Wenn Sie nicht wollen dass ein Attribut veröffentlicht wird deaktivieren Sie das entsprechende Kontrollkästchen in der *WMS* oder *WFS* Spalte.

Sie können Wasserzeichen über die Karten legen, die von Ihrem WMS produziert werden, indem Sie Beschriftungstexte oder SVG-Anmerkungen hinzufügen. Siehe Beschriftungstools unter *Allgemeine Werkzeuge* für Anleitungen zum Erstellen von Anmerkungen. Für Anmerkungen die als Wasserzeichen in der WMS Ausgabe dargestellt werden sollen muss das *Feste Kartenposition* Kontrollkästchen im *Anmerkungstext* Dialog deaktiviert sein. Dies können Sie erreichen indem Sie auf die Anmerkung doppelklicken während eines der Beschriftungstools aktiv ist. Bei SVG-Anmerkungen werden Sie entweder das Projekt einstellen müssen um absolute Pfade zu speichern (im *Allgemein* Menü des *Projekt* → *Projekteigenschaften* Dialogs) oder den Pfad zum SVG-Bild manuell so verändern dass es einen gültigen relativen Pfad darstellt.



## Zusätzliche Paramter die vom WMS GetMap Request unterstützt werden.

Im WMS GetMap Request akzeptiert der QGIS Server ein paar zusätzliche Paramter zusätzlich zu den Standardparametern gemäß der OGC WMS 1.3.0 Spezifikation:

- **MAP** Parameter: Ähnlich wie MapServer kann der MAP Parameter zum Angeben des Pfads zur QGIS Projektdatei verwendet werden. Sie können einen absoluten Pfad oder einen Pfad relativ zum Speicherort der ausführbaren Serverdatei (`qgis_mapserv.fcgi`) angeben. Wenn nichts angegeben ist sucht QGIS Server nach .qgs Dateien in dem Verzeichnis in dem die ausführbare Serverdatei gespeichert ist.

Beispiel:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\
REQUEST=GetMap&MAP=/home/qgis/mymap.qgs&...
```

- **DPI** Parameter: Der DPI Parameter kann verwendet werden um die angefragte Ausgabeauflösung anzugeben.

Beispiel:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?REQUEST=GetMap&DPI=300&...
```

- **OPACITIES** Parameter: Die Deckkraft kann auf Layer- oder Gruppenebene eingestellt werden. Zugelassene Werte bewegen sich von 0 (ganz transparent) bis 255 (volle Abdeckung).

Beispiel:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\
REQUEST=GetMap&LAYERS=mylayer1,mylayer2&OPACITIES=125,200&...
```

## QGIS Server Logging

Um Requests zu loggen, die zum Server gesendet werden, stellen Sie die folgenden Umgebungsvariablen ein:

- **QGIS\_SERVER\_LOG\_FILE**: Legen sie Pfad und Dateiname fest. Vergewissern Sie sich dass der Server die richtigen Rechte zum Schreiben in eine Datei hat. Die Datei sollte automatisch erstellt werden, senden Sie einfach ein paar Requests an den Server. Wenn sie nicht auftaucht überprüfen Sie die Rechte.
- **QGIS\_SERVER\_LOG\_LEVEL**: Bestimmen Sie den gewünschten Loglevel. Mögliche Werte sind:
  - 0 INFO (alle Requests loggen),
  - 1 WARNING,
  - 2 CRITICAL (loggt nur kritische Fehler, geeignet für Produktionszwecke).

Beispiel:

```
SetEnv QGIS_SERVER_LOG_FILE /var/tmp/qgislog.txt
SetEnv QGIS_SERVER_LOG_LEVEL 0
```

### Beachten Sie

- Wenn Sie das Fcgid Modul benutzen verwenden Sie FcgidInitialEnv anstelle von SetEnv!
- Das Server Logging ist auch aktiviert wenn die ausführbaren Dateien im Release Modus kompiliert sind.

## Umgebungsvariablen

- **QGIS\_OPTIONS\_PATH**: Die Variable legt den Pfad zum Verzeichnis mit Einstellungen fest. Es funktioniert auf die gleiche Weise wie die QGIS Application `-optionspath` Option. Sie schaut nach einer Einstellungsdatei in `<QGIS_OPTIONS_PATH>/QGIS/QGIS2.ini`. Um z.B. QGIS Server auf einem Apache einzurichten um die `/path/to/config/QGIS/QGIS2.ini` Einstellungsdatei zu verwenden, fügen Sie Apache config folgendes hinzu:

```
SetEnv QGIS_OPTIONS_PATH "/path/to/config/"
```

---

## Arbeiten mit GPS Daten

---


### 15.1 GPS Plugin



#### 15.1.1 Was ist GPS?

GPS, das Global Positioning System, ist ein satellitenbasiertes System das es jedem mit einem GPS Empfänger ermöglicht seine genaue Position überall auf der Welt zu finden. GPS wird als Navigationshilfe in Flugzeugen, in Schiffen und von Wanderern benutzt. Der GPS Empfänger verwendet die Signale von den Satelliten um seine Länge, Breite und (manchmal) Höhe zu berechnen. Die meisten Empfänger haben auch die Fähigkeit Orte (bekannt als **Wegpunkte**), Sequenzen von Wegpunkten, die zusammen eine geplante **Route** ergeben und Tracklogs oder **Spuren** der zeitlichen Bewegung des Empfängers zu speichern. Wegpunkte, Routen und Spuren sind die drei Grundtypen in GPS Daten. QGIS stellt Wegpunkte in Punktlayern dar, wohingegen Routen und Spuren in Vektorlinien dargestellt werden.


#### 15.1.2 GPS-Daten aus einer Datei laden

Es gibt dutzende von verschiedenen Dateiformaten zum Speichern von GPS Daten. Das Format das QGIS benutzt wird GPX (GPS eXchange format) genannt, welches ein Standardaustauschformat ist welches eine beliebige Anzahl von Wegpunkten, Routen und Spuren in einer Datei enthalten kann.

Um eine GPX Datei zu laden müssen Sie zuerst das Plugin laden. *Erweiterungen* → *ImActionShowPluginManager!* *menuselection: 'Erweiterungen verwalten und installieren ...'* öffnet den Pluginmanagerdialog. Aktivieren Sie das  *GPS Werkzeuge* Kontrollkästchen. Wenn dieses Plugin geladen ist erscheinen zwei Knöpfe mit einem kleinen tragbaren GPS Gerät in der Werkzeugleiste:

-  Erstelle neuen GPX Layer
-  GPS-Werkzeuge

Zum Arbeiten mit GPS Daten stellen wir Ihnen eine Beispiel GPX Datei, die im QGIS Beispieldatensatz zu finden ist, zur Verfügung: `qgis_sample_data/gps/national_monuments.gpx`. Siehe Abschnitt *Beispieldaten* für weitere Informationen über die Beispieldaten.

1. Wählen Sie *Vektor* → *GPS* → *GPS Werkzeuge* oder klicken Sie das  *GPS-Werkzeuge* Icon in der Werkzeugleiste und öffnen Sie den *GPX-Datei laden* Reiter (siehe [figure\\_GPS\\_1](#)).
2. Suchen Sie den Ordner `qgis_sample_data/gps/`, wählen Sie die GPX-Datei `national_monuments.gpx` und klicken Sie [**Öffnen**].

Verwenden Sie den [**Suchen...**] Knopf um die GPX-Datei auszuwählen, verwenden Sie dann die Kontrollkästchen um die Objekttypen die aus der GPX-Datei geladen werden sollen auszuwählen. Jeder Objekttyp wird in einen



Abbildung 15.1: Das *GPS Werkzeuge* Dialogfenster 

separaten Layer geladen wenn Sie **[OK]** klicken. Die Datei `national_monuments.gpx` enthält nur Wegpunkte.

---

**Bemerkung:** GPS Einheiten ermöglichen es Ihnen Daten in verschiedenen Koordinatensystemen zu speichern. Wenn Sie eine GPX Datei herunterladen (von Ihrer GPS Einheit oder einer Webseite) und Sie dann in QGIS laden versichern Sie sich das die Daten die in Ihrer GPX-Datei gespeichert sind WGS 84 (Länge/Breite) verwenden. QGIS erwartet dieses und es ist die offizielle GPX Spezifikation. Siehe <http://www.topografix.com/GPX/1/1/>.

---

### 15.1.3 GPSBabel

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 GPSBabel 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 GPSBabel auch dazu, daher ist es ratsam, es zu installieren. Wenn Sie aber lediglich GPX-Dateien in QGIS laden möchten, geht dies auch ohne GPSBabel. Version 1.2.3 funktioniert auf jeden Fall mit QGIS, neuere aber wahrscheinlich auch.

### 15.1.4 GPS-Daten importieren

Um GPS-Daten aus einer Datei, die nicht im GPX-Format vorliegt zu importieren, benutzen Sie den Reiter *Aus anderer Datei importieren*. Wählen Sie dann die Datei (und den Dateityp), die importiert werden soll aus, von welchem Datenformat Sie importieren möchten und wo die konvertierte GPX-Datei unter welchem Namen abgelegt werden soll. Beachten Sie, dass nicht für alle Datenformate die drei GPS-Datentypen Wegpunkte, Routen und Spuren unterstützt werden. Manchmal sind es nur ein oder zwei.

### 15.1.5 GPS-Daten von einem Empfänger herunterladen

QGIS kann mit dem Programm GPSBabel GPS-Daten direkt von einem GPS-Empfänger in einen Vektorlayer laden. Dazu können Sie den Reiter *Vom GPS herunterladen* verwenden (siehe [Figure\\_GPS\\_2](#)), indem Sie den Typ Ihres GPS-Empfängers angeben, den Verbindungsport (oder USB wenn Ihr GPS dies unterstützt), den Namen der GPX-Datei, in welche 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 GPSBabel mit dem Gerät kommuniziert.

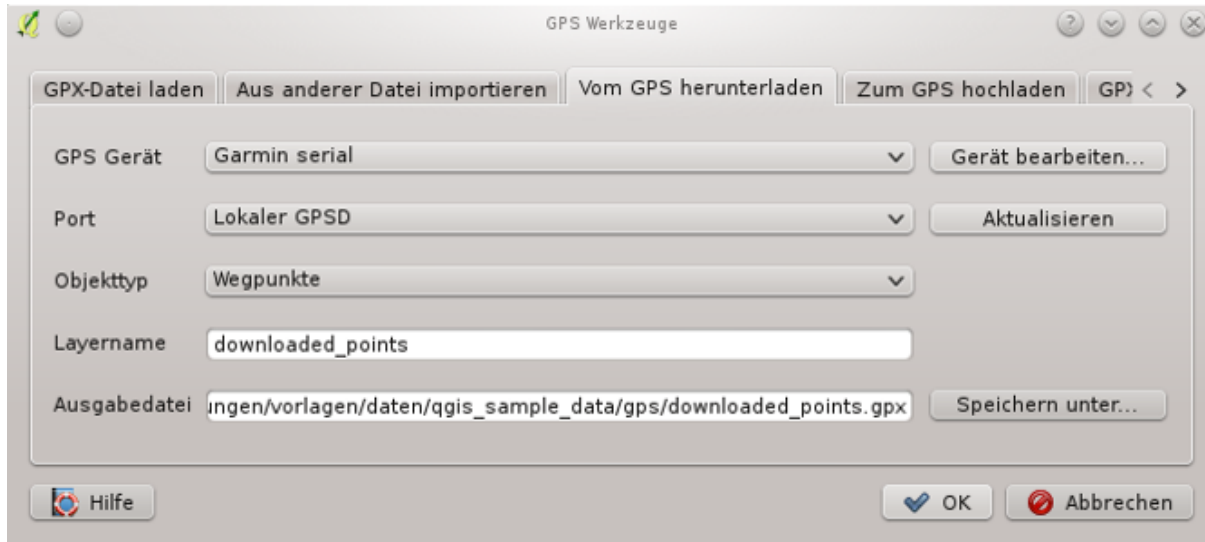




Abbildung 15.2: Das Downloadwerkzeug

Wenn kein vorhandener Typ mit Ihrem Empfänger funktioniert, können Sie einen eigenen, neuen Gerätetyp erstellen (vgl. Abschnitt *Neues GPS-Gerät definieren*).

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. Es kann auch einfach USB sein, wenn dies von dem GPS-Gerät unterstützt wird.

-  Unter Linux ist dies etwas wie `/dev/ttyS0` oder `/dev/ttyS1`.
-  Unter Windows ist dies `COM1` oder `COM2`.

Wenn Sie **[OK]** klicken werden die Daten vom GPS-Empfänger heruntergeladen und in QGIS als Vektorlayer dargestellt.

### 15.1.6 GPS-Daten auf einen Empfänger hochladen

Sie können auch einen Vektorlayer aus QGIS auf einen GPS-Empfänger hochladen, indem Sie den Reiter *Zum 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 (oder USB). Genau wie beim Reiter *Vom 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 in Kombination mit den Vektorbearbeitungsfunktionen von QGIS. Es ermöglicht Ihnen eine Karte zu laden, Wegpunkte und Routen zu erstellen und diese dann auf Ihr GPS Gerät hochzuladen und zu verwenden.

### 15.1.7 Neues GPS-Gerät 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, die in *Vom GPS herunterladen* und *Zum GPS hochladen* aufgelistet sind, funktioniert, können Sie selbst einen erstellen. Dazu klicken Sie auf den Knopf **[Editiere Geräte]** im Hochladen Reiter oder Herunterladen Reiter.

Um ein neues Gerät zu definieren klicken Sie einfach den Knopf **[Neu]**, geben einen Namen, einen Downloadbefehl und einen Uploadbefehl für Ihr Gerät ein und klicken den **[Aktualisieren]** Knopf. Danach steht der Name des Gerätes in den Reitern *Zum GPS hochladen* und *Vom GPS herunterladen* zur Verfügung und kann in Form jeder Zeichenfolge eingegeben werden. Der Downloadbefehl ist der Befehl der zum Downloaden der Daten von einem Gerät in eine GPX-Datei verwendet wird. Dies ist wahrscheinlich ein GPSSbabelbefehl, man kann aber auch jedes

andere Kommandozeilenprogramm das eine GPX-Datei erstellen kann benutzen. 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 Schlüsselwörter und startet das Kommando `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

Das Kommando hinaufladen wird benutzt, um die Daten auf Ihren GPS-Empfänger zu transferieren. Es werden dazu die gleichen Schlüsselwörter benutzt, nur dass `%in` durch den Namen der hochzuladenden GPX-Datei und `%out` durch den Namen des Verbindungsports ersetzt wird.

Sie können mehr über GPSBabel und seine Funktionen unter der URL <http://www.gpsbabel.org> erlernen.

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

### 15.1.8 Downloaden von Punkten/Spuren von GPS Einheiten

Wie in vorigen Abschnitten beschrieben verwendet QGIS GPSBabel um Punkte/Spuren direkt ins Projekt herunterzuladen. QGIS wird mit einem vordefinierten Profil zum Downloaden von Garmingeräten zur Verfügung gestellt. Unglücklicherweise gibt es dort einen [bug](#) der es unmöglich macht andere Profile zu erstellen, also ist das direkte Downloaden in QGIS mithilfe der GPS Werkzeuge im Moment auf Garmin UBS Einheiten begrenzt.

#### Garmin GPSMAP 60cs

##### MS Windows

Installieren Sie Garmin USB Treiber von [http://www8.garmin.com/support/download\\_details.jsp?id=591](http://www8.garmin.com/support/download_details.jsp?id=591)

Verbinden Sie die Einheit. Öffnen Sie GPS Werkzeuge und verwenden Sie `type=garmin serial` und `port=usb:.`. Füllen Sie die Felder *Layername* und *Ausgabedatei* aus. Manchmal scheint es Probleme beim Speichern in einen bestimmten Ordner zu geben, wenn Sie etwas wie `c:\temp` verwenden funktioniert es für gewöhnlich.

##### Ubuntu/Mint GNU/Linux

Es wird zuerst ein Eintrag über die Rechte des Gerätes benötigt, wie beschrieben auf [https://wiki.openstreetmap.org/wiki/USB\\_Garmin\\_on\\_GNU/Linux](https://wiki.openstreetmap.org/wiki/USB_Garmin_on_GNU/Linux). Sie können versuchen eine Datei `/etc/udev/rules.d/51-garmin.rules` zu erstellen, die diese Regel enthält.

```
ATTRS{idVendor}=="091e", ATTRS{idProduct}=="0003", MODE="666"
```

Danach ist es nötig sicher zu gehen das das `garmin_gps` Kernelmodul nicht geladen ist

```
rmmod garmin_gps
```

und dann können Sie die GPS Werkzeuge verwenden. Leider scheint es einen [bug](#) zu geben und QGIS friert mehrere Male ein bevor die Operation gut funktioniert.

#### BTGP-38KM Datenlogger (nur Bluetooth)

##### MS Windows

Der bereits erwähnte Bug lässt es nicht zu, dass Daten innerhalb von QGIS heruntergeladen werden, also müssen Sie GPSBabel aus der Kommandozeile heraus oder mit Hilfe seiner Schnittstelle verwenden.

```
gpsbabel -t -i skytraq,baud=9600,initbaud=9600 -f COM9 -o gpx -F C:/GPX/aaa.gpx
```

### Ubuntu/Mint GNU/Linux

Verwenden Sie den gleichen Befehl (oder Einstellungen wenn Sie die GPSBabel GUI verwenden) wie in Windows. Unter Linux ist vielleicht üblich eine Nachricht wie folgt zu erhalten

```
skytraq: Too many read errors on serial port
```

es ist einfach eine Frage des aus- und anschalten des Dataloggers und es erneut zu versuchen.

## BlueMax GPS-4044 Datalogger (sowohl BT als auch USB)

### MS Windows

**Bemerkung:** Es muss seine Treiber installieren bevor man es unter Windows 7 verwendet. Sehen Sie in den Herstellerseiten für den richtigen Download nach.

Downloaden mit GPSBabel, mit USB und BR, gibt immer einen Fehler heraus wie

```
gpsbabel -t -i mtk -f COM12 -o gpx -F C:/temp/test.gpx
mtk_logger: Can't create temporary file data.bin
Error running gpsbabel: Process exited unsuccessfully with code 1
```

### Ubuntu/Mint GNU/Linux

#### Mit USB

Nachdem Sie das Kabel angeschlossen haben verwenden Sie den `dmesg` Befehl um zu verstehen welcher Port benutzt wird, zum Beispiel `/dev/ttyACM3`. Benutzen Sie dann wie immer GPSBabel aus der Kommandozeile oder der GUI.

```
gpsbabel -t -i mtk -f /dev/ttyACM3 -o gpx -F /home/user/bluemax.gpx
```


#### Mit Bluetooth

Verwenden Sie Blueman Device Manager um das Gerät zu verbinden und machen Sie es über einen Systemport zugänglich, starten Sie dann GPSBabel.





```
gpsbabel -t -i mtk -f /dev/rfcomm0 -o gpx -F /home/user/bluemax_bt.gpx
```

.

## 15.2 Live GPS tracking

Um das Live GPS tracking in QGIS zu aktivieren müssen Sie *Einstellungen* → *Bedienfelder* →  *GPS-Information* wählen. Es erscheint ein neues Dockfenster auf der linken Seite der Oberfläche.


Es sind 4 Bildschirme im GPS Tracking Fenster möglich:

-  GPS Positionskoordinaten und eine Schnittstelle für manuelles Eintragen von Stützpunkten und Objekten.
-  GPS Signalstärke von Satellitenverbindungen
-  GPS Polarbildschirm der die Anzahl und die polare Position von Satelliten zeigt
-  GPS Optionsansicht (siehe [figure\\_gps\\_options](#))

Mit einem angeschlossenen GPS-Empfänger (muss auch von Ihrem Betriebssystem unterstützt werden) reicht ein einfacher Klick auf den Knopf **[Verbinden]**. Ein zweiter Klick (jetzt auf **[Trennen]**) trennt die Verbindung zu dem GPS-Empfänger wieder. Für GNU/Linux wurde `gpsd` Support integriert. Dadurch ist eine einfache Verbindung zu den meisten GPS Geräten gewährleistet. Sie müssen `gpsd` im Vorfeld konfigurieren, damit QGIS sich damit verbinden kann.

**Warnung:** Wenn Sie Ihre Position in die Oberfläche aufnehmen wollen müssen Sie erst einen neuen Vektorlayer erstellen und diesen in den Bearbeitungsmodus bringen um Ihren Track aufnehmen zu können.

## 15.2.1 Positionskoordinaten

 Wenn das GPS Signale von Satelliten empfängt werden Sie Ihre Position als Länge, Breite und Höhe zusammen mit zusätzlichen Attributen sehen.

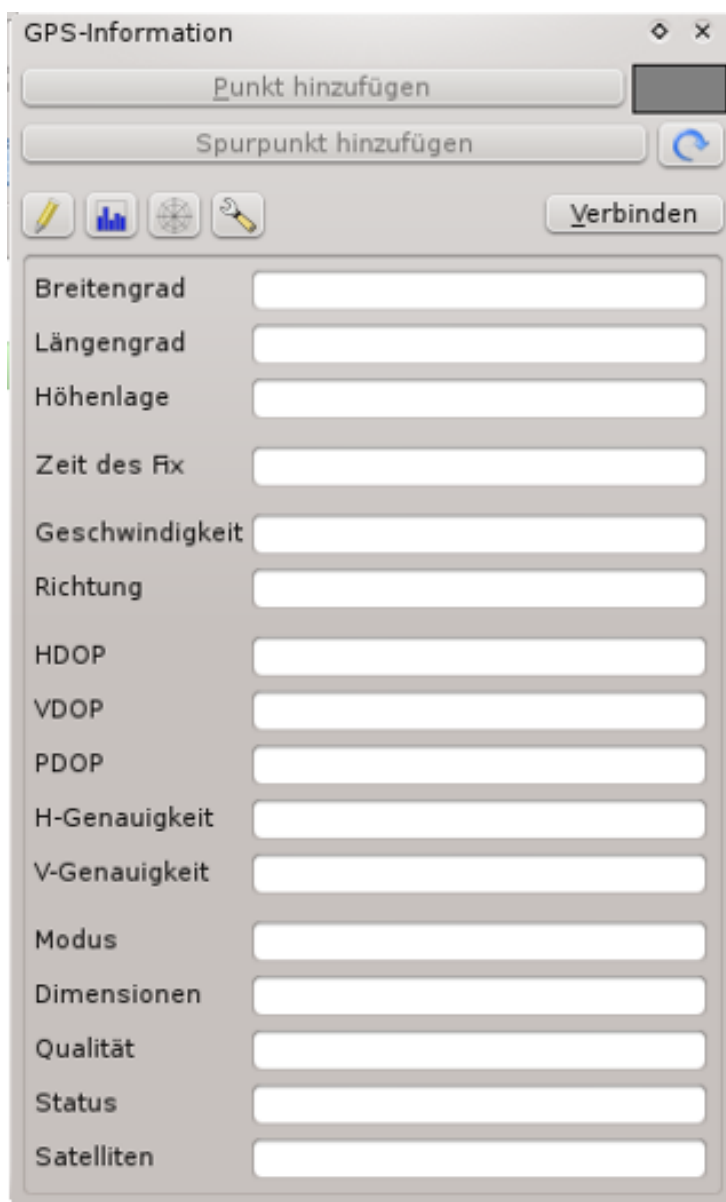


Abbildung 15.3: GPS Tracking Position und zusätzliche Attribute 



### 15.2.2 GPS Signalstärke

 Hier können Sie die Signalstärke der Satelliten von denen Sie Signale empfangen sehen.

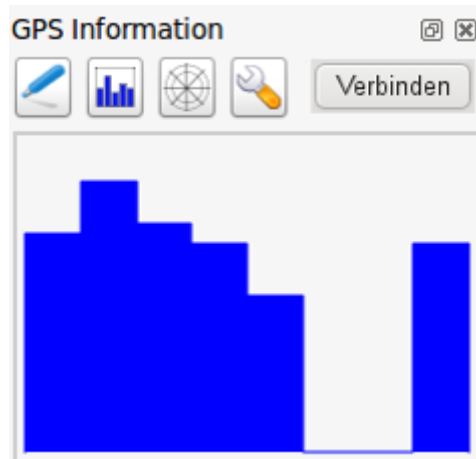



Abbildung 15.4: GPS Tracking Signalstärke 

### 15.2.3 GPS Polar-Bildschirm

 Wenn Sie wissen wollen, wo sich die Satelliten befinden, mit denen Sie gerade verbunden sind, wechseln Sie zum GPS Polarbildschirm. Sie können auch die IDs der Satelliten sehen, von denen Sie Signale empfangen.

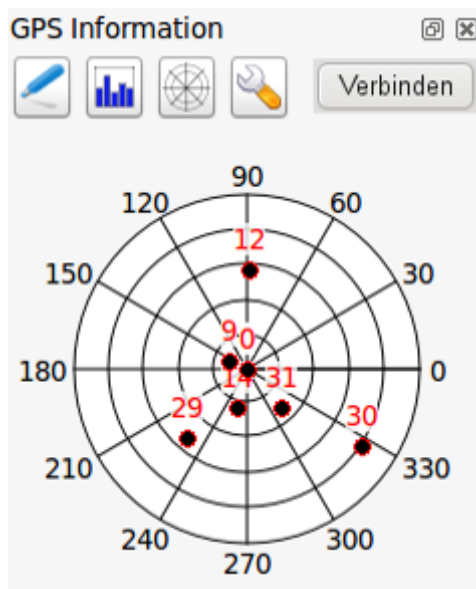



Abbildung 15.5: GPS Tracking Polarbildschirm 

### 15.2.4 GPS Optionen


 Wenn es Probleme bei der Verbindung zum GPS-Gerät geben sollte können Sie innerhalb dieser Einstellungen wechseln:

- *Automatisch feststellen*

- *Intern*
- *Serielles Gerät*
- *gpsd* (wählt den Host, Port und das Gerät mit dem Ihr GPS verbunden ist)


Ein wiederholter Klick auf [**Verbinden**] stellt die Verbindung zum GPS-Gerät wieder her.

Sie können  *Hinzugefügte Objekte automatisch speichern* aktivieren wenn Sie sich im Bearbeitungsmodus befinden. Oder Sie können  *Punkte automatisch hinzufügen* aktivieren um Punkte mit einer bestimmten Größe und Farbe der Kartenansicht hinzuzufügen.

Indem Sie das Kontrollkästchen  *Cursor* aktivieren, können Sie den Schieberegler  verwenden, um den Positionscursor im Kartenfenster kleiner oder größer zu machen.

Das Aktivieren des Radioknopfes  *Karte zentrieren* ermöglicht es auszuwählen, wie das Kartenfenster aktualisiert werden soll. Dies enthält 'immer beim Verlassen', wenn die aufgenommenen Koordinaten den Bereich des Kartenfensters verlassen oder 'niemals', um die Kartenausschnitt beizubehalten.

Schliesslich können Sie das Kontrollkästchen  *Logdatei* aktivieren und einen Pfad angeben, wo die Logdateien über die GPS-Messung abgelegt werden.

Wenn Sie ein Objekt manuell angeben wollen, müssen Sie zurück zu  *Position* gehen und dann auf [**Punkt hinzufügen**] oder [**Wegpunkt hinzufügen**] klicken.

### 15.2.5 Ein Bluetooth GPS fürs Live Tracking anbinden


Mit QGIS können Sie ein Bluetooth GPS für das Aufnehmen von Felddaten anbinden. Um dies durchzuführen benötigen Sie ein Bluetooth Gerät und einen Bluetooth Empfänger auf Ihrem Computer.

Als erstes müssen Sie Ihr GPS Gerät erkennen lassen und mit dem Computer verbinden. Stellen Sie das GPS an, gehen Sie zum Bluetooth Icon in Ihrem Infobereich und suchen Sie nach einem Neuen Gerät.

Vergewissern Sie sich dass auf der rechten Seite Ihrer Geräteauswahlmaske alle Geräte ausgewählt sind so dass Ihre GPS Einheit darunter in Erscheinung tritt. Im nächsten Schritt sollte ein serieller Verbindungsservice erhältlich sein, wählen Sie ihn aus und klicken Sie auf den [**Verbinden**] Knopf.

Denken Sie daran dass die an die GPS Verbindung angebotenen COM Ports aus den Bluetooth Eigenschaften resultieren.

Machen Sie die Kopplung für die Verbindung nachdem das GPS erkannt wurde. Normalerweise ist der Authorisationskode 0000.

Öffnen Sie jetzt das *GPS-Information* Bedienfeld und wechseln Sie zur  *GPS Optionsansicht*. Wählen Sie den zu der GPS Verbindung zugewiesenen COM Port und klicken Sie [**Verbinden**]. Nach kurzer Zeit sollte ein Cursor, der Ihre Position anzeigt erscheinen.

Wenn QGIS keine GPS Daten empfangen kann sollten Sie Ihr GPS Gerät neustarten, 5-10 Sekunden warten und dann wieder eine Verbindung versuchen. Normalerweise funktioniert diese Lösung. Wenn Sie wieder einen Verbindungsfehler erhalten vergewissern Sie sich dass kein anderer Bluetoothempfänger, der an die gleiche GPS Einheit gekoppelt ist, in Ihrer Nähe ist.

### 15.2.6 GPSPMAP 60cs verwenden

#### MS Windows

Der einfachste Weg, damit es funktioniert ist eine Middleware (Freeware, nicht offen) genannt *GPSSGate* zu verwenden.

Starten Sie das Programm, lassen Sie es nach GPS Geräten suchen (funktioniert sowohl mit USB als Bluetooth Geräten), und klicken Sie in Sie in QGIS einfach [**Verbinden**] wobei Sie im Live Tracking Bedienfeld den  *Automatisch feststellen* Modus verwenden.

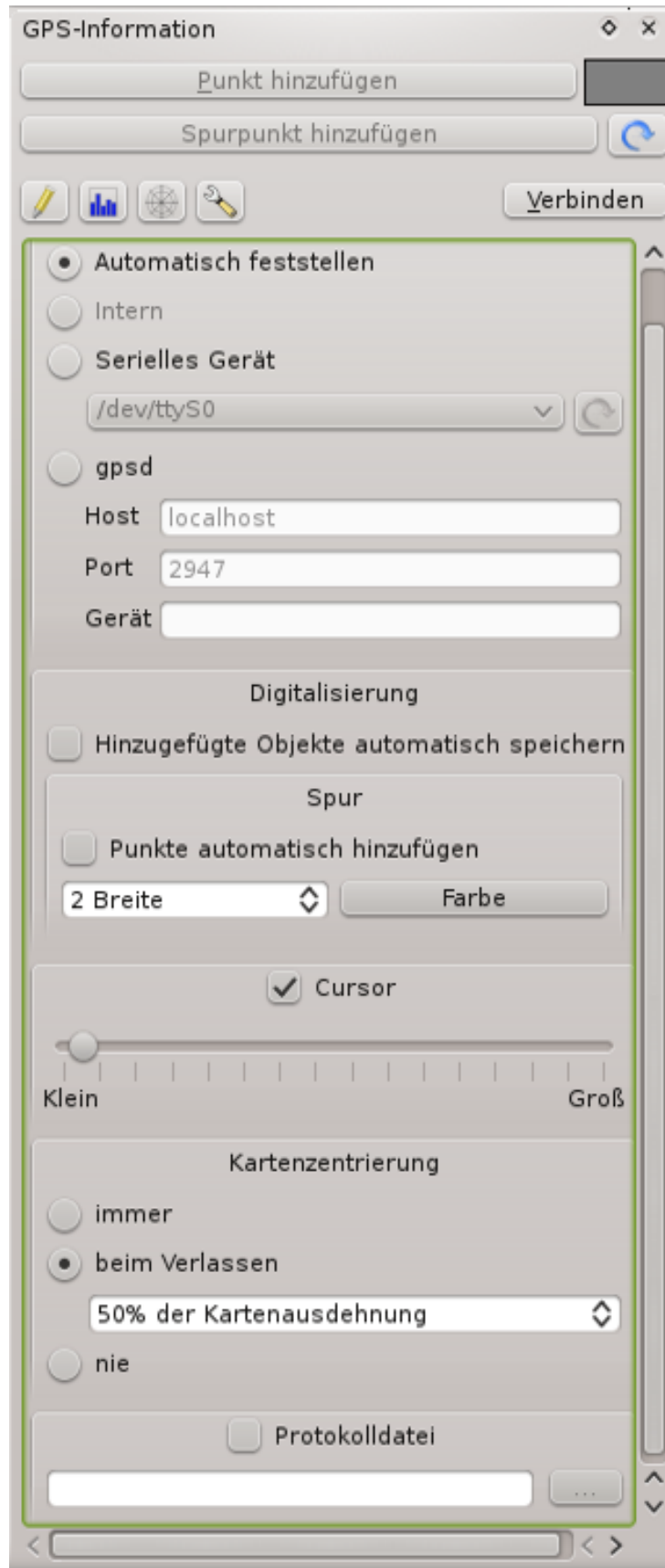


Abbildung 15.6: GPS Tracking Optionsansicht 

### Ubuntu/Mint GNU/Linux

Wie unter Windows ist der einfachste Weg einen Server, in diesem Fall GPSD, dazwischen zu benutzen, also

```
sudo apt-get install gpsd
```

Laden Sie dann das `garmin_gps` Kernelmodul

```
sudo modprobe garmin_gps
```

Und verbinden Sie dann die Einheit. Überprüfen Sie dann mit `dmesg` die aktuelle von dem Gerät verwendete Einheit, zum Beispiel `/dev/ttyUSB0`. Starten Sie jetzt `gpsd`

```
gpsd /dev/ttyUSB0
```


Und verbinden Sie sich zuletzt mit dem QGIS Live Tracking Tool.

### 15.2.7 BTGP-38KM Datenlogger verwenden (nur Bluetooth)

Sie können GPSD (unter Linux) oder GPSGate (unter Windows) mühelos verwenden.

### 15.2.8 BlueMax GPS-4044 Datenlogger verwenden (sowohl BT als auch USB)

#### MS Windows

Das Live Tracking funktioniert mit USB und BT Modus, mit oder ohne GPSGate, benutzen Sie einfach den  *Automatisch feststellen* Modus oder stellen Sie das Tool auf den richtigen Port ein.

### Ubuntu/Mint GNU/Linux

#### Für USB

Das Live Tracking funktioniert sowohl mit GPSD

```
gpsd /dev/ttyACM3
```

oder ohne es, indem man das QGIS Live Tracking Tool direkt mit dem Gerät verbindet (zum Beispiel `/dev/ttyACM3`).

#### Für Bluetooth

Das Live Tracking funktioniert sowohl mit GPSD

```
gpsd /dev/rfcomm0
```

oder ohne es, indem man das QGIS Live Tracking Tool direkt mit dem Gerät verbindet (zum Beispiel `/dev/rfcomm0`).











---

## GRASS GIS Integration


---

Das GRASS Plugin bietet den Zugriff zu GRASS GIS Datenbanken und Funktionalitäten (siehe GRASS-PROJECT in *Literatur und Internetreferenzen*). Dies beinhaltet das Visualisieren von GRASS Raster- und Vektorlayern, das Bearbeiten von Vektorattributen, das Erstellen von neuen Vektorlayern und das Analysieren von 2-D und 3-D Daten mit mehr als 400 GRASS Modulen.

In diesem Kapitel bekommen Sie eine Einführung in die Plugin Funktionalitäten und zeigen Ihnen einige Beispiele wie man mit GRASS Daten arbeitet und diese verwaltet. Folgende Funktionen stehen über das GRASS Plugin zur Verfügung, wenn Sie es wie in Abschnitt *GRASS Plugin starten* beschrieben laden:

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







### 16.1 GRASS Plugin starten

Um GRASS Funktionalitäten zu benutzen und/oder GRASS Vektor- und Rasterlayer in QGIS zu visualisieren müssen Sie das GRASS Plugin mit dem Pluginmanager auswählen und laden. Gehen Sie deswegen zum Menü *Erweiterungen* →  *Erweiterungen verwalten und installieren ...*, wählen Sie  *GRASS* und klicken Sie **[OK]**.

Sie können wie in Abschnitt *GRASS Layer visualisieren* beschrieben, direkt Raster- und Vektorlayer aus einer existierenden GRASS-Datenbank LOCATION laden (siehe Abschnitt *sec\_load\_grassdata*), oder Sie können eine neue GRASS LOCATION mit QGIS erstellen (siehe Abschnitt *Eine neue GRASS LOCATION erstellen*), einige Raster- und Vektordaten in diese neue Location importieren (siehe Abschnitt *import\_loc\_data*) und über die GRASS-Werkzeuge weitere Analysen durchführen (siehe Abschnitt *Die GRASS Werkzeugkiste*).

## 16.2 GRASS Layer visualisieren

Wenn das GRASS-Plugin geladen ist, können Sie GRASS Vektor- und Rasterlayer mit den entsprechenden Knöpfen in der Werkzeugleiste laden. Als Beispiel benutzen wir den QGIS Alaska Beispieldatensatz (siehe Kapitel *Beispieldaten*). Dieser enthält eine kleine GRASS LOCATION mit 3 Vektor- und einem Rasterlayer mit Höheninformationen.

1. Erstellen Sie einen neuen Ordner `grassdata`, laden Sie den QGIS 'Alaska' Datensatz `qgis_sample_data.zip` von <http://download.osgeo.org/qgis/data/> herunter und entpacken Sie die Datei nach `grassdata`.
2. Starten Sie QGIS.
3. Falls noch nicht in einer vorhergehenden QGIS Sitzung geschehen, laden Sie das GRASS Plugin indem Sie *Erweiterungen* →  *Erweiterungen verwalten und installieren ...* klicken und  GRASS aktivieren. Die GRASS Werkzeugleiste erscheint im QGIS Hauptfenster.
4. Klicken Sie in der GRASS Werkzeugleiste das  *Mapset öffnen* Icon um den *Wählen Sie ein GRASS Mapset* Assistenten zu öffnen.
5. Als `gisdbase` suchen und wählen Sie bitte den Pfad zum Ordner `grassdata`.
6. Sie sollten nun als *LOCATION*  `alaska` und als *MAPSET*  `demo` auswählen können.
7. Klicken Sie auf **[OK]**. Einige weitere, zuvor grau hinterlegte GRASS-Werkzeuge sind nun aktiv.
8. Klicken Sie auf  *GRASS-Rasterlayer hinzufügen*, wählen Sie den Layer `gtopo30` und drücken Sie auf **[OK]**. Die Höhendaten werden nun dargestellt.
9. Klicken Sie auf  *GRASS-Vektorlayer hinzufügen*, wählen Sie den Kartennamen `alaska` aus und klicken Sie **[OK]**. Der Alaska Grenzvektorlayer wird über die `gtopo30` Karte gelegt. Sie können jetzt die Layereigenschaften wie in Kapitel *Vektorlayereigenschaften* beschrieben anpassen (z.B. Ändern der Deckkraft, Füll- und Umrandungsfarbe).
10. Laden Sie auch noch die weiteren Vektorlayer `rivers` und `airports` hinzu und passen Sie deren Layereigenschaften an.

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. Weitere Beispiel GRASS-Datenbanken finden Sie auf der GRASS Homepage unter: <http://grass.osgeo.org/download/sample-data/>.

---

### **Typ:** Probleme beim Laden von GRASS-Layern

Sollten Sie Probleme beim Laden von GRASS-Layern haben oder sich QGIS unvorhergesehen beendet, dann überprüfen Sie bitte, ob das GRASS-Plugin korrekt geladen ist wie in Abschnitt *GRASS Plugin starten* beschrieben.

---

## 16.3 Information zur GRASS-Datenbank

GRASS Daten werden in einem Ordner gespeichert, der als `GISDBASE` bezeichnet wird. Standardmäßig wird der Ordner `grassdata` genannt und er muss erstellt worden sein, bevor man beginnt, mit dem GRASS Plugin in QGIS zu arbeiten. Innerhalb dieses Ordners sind die GRASS Daten als Projekte in Unterordnern genannt `LOCATION`'s organisiert. Jede `:file:'LOCATION` ist durch ein Koordinatenbezugssystem, Kartenprojektion und eine geographische Grenze definiert und kann darüberhinaus weitere Unterordner `MAPSETS` besitzen, um die Layer der `LOCATION` weiter z.B. thematisch oder räumlich zu unterteilen (Neteler & Mitasova 2008 *literature\_and\_web*). Um Raster- und Vektorlayer mit den GRASS Modulen zu analysieren, müssen diese zuerst in eine passende GRASS `LOCATION` importiert werden. (Dies ist nicht ganz korrekt. Mit den GRASS Modulen `r.external` und `v.external` können Sie eine 'read-only' Verknüpfung zu externen durch

GDAL/OGR-unterstützte Layer erstellen, ohne die Daten importieren zu müssen. Da dies aber nicht der normale Weg für GRASS Anfänger ist, wird auf diese Möglichkeit nicht näher eingegangen.)

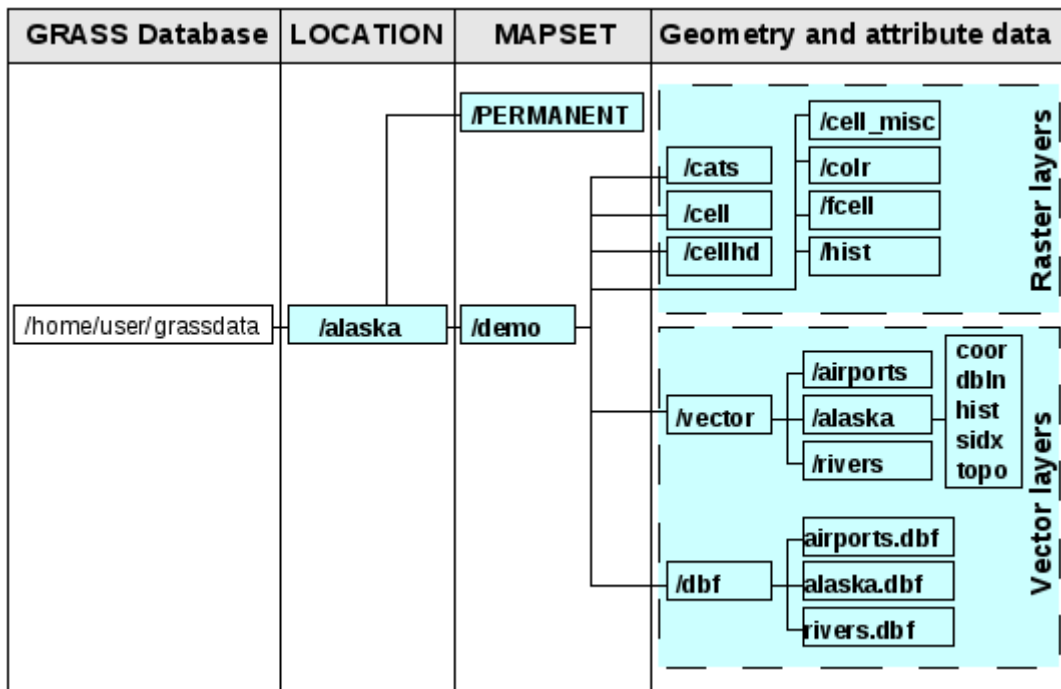




Abbildung 16.1: GRASS Daten in der Alaska LOCATION

### 16.3.1 Eine neue GRASS LOCATION erstellen

Als ein Beispiel möchten wir Ihnen zeigen, wie die GRASS LOCATION des Alaska Beispieldatensatzes erstellt wurde. Das Koordinatenbezugssystem ist Albers Equal Area mit der Einheit 'feet'. Diese GRASS Location `alaska` wird für alle GRASS GIS Beispiele verwendet. Es ist also sinnvoll, sich diesen Datensatz zu installieren (siehe Abschnitt *Beispieldaten*).

1. Starten Sie QGIS und stellen Sie sicher dass das GRASS Plugin geladen ist.
2. Visualisieren Sie die `alaska.shp` Shapedatei (siehe Abschnitt *Shape Layer laden*) aus dem Alaska Datensatz (siehe *Beispieldaten*).
3. Klicken Sie in der GRASS Werkzeugleiste auf das  `Neues Mapset` Icon um den *Neues Mapset* Assistenten zu öffnen.
4. Wählen Sie einen bestehende GRASS Datenbankordner (GISDBASE) `grassdata` oder erstellen Sie einen für die neue LOCATION anhand des Dateimanagers auf Ihrem Computer. Klicken Sie dann **[Weiter]**.
5. Wir können diesen Assistenten verwenden um eine neue MAPSET innerhalb einer bestehenden LOCATION zu erstellen (siehe Abschnitt *Eine neue GRASS MAPSET erstellen*) oder um dabei auch eine neue LOCATION zu erstellen. Wählen Sie  *Erstelle neue Location* (see *figure\_grass\_location\_2*).
6. Geben Sie einen Namen für die LOCATION - wir haben 'alaska' verwendet - und klicken Sie **[Weiter]**.
7. Definieren Sie die Projektion indem Sie den Radiobutton  *Projektion* zum Aktivieren der Projektionsliste klicken.
8. Wir benutzen Albers Equal Area Alaska (Fuß) Projektion. Da wir zufällig wissen dass es durch den EPSG ID 2964 repräsentiert wird geben wir dies in das Suchfenster ein. (Bemerkung: Wenn Sie diesen Ablauf für eine andere LOCATION und Projektion wiederholen wollen und Sie die EPSG ID nicht gespeichert haben,

klicken Sie auf das  KBS-Status Icon in der unteren rechten Ecke der Statusleiste (siehe Abschnitt *Arbeiten mit Projektionen*)).

9. Geben Sie bei *Filter* 2964 ein um die Projektion auszuwählen.
10. Klicken Sie auf [**Weiter**].
11. Um die Standardregion zu definieren müssen wir die LOCATION Ausdehnung in Nord-, Süd-, Ost- und Westrichtung eingeben. Hier klicken wir einfach auf den Knopf [**Setze aktuelle QGIS-Ausdehnung**] um die Ausdehnung des geladenen Layers `alaska.shp` als voreingestellte GRASS Region Ausmaße geltend zu machen.
12. Klicken Sie auf [**Weiter**].
13. Wir müssen auch eine MAPSET innerhalb unserer neuen LOCATION definieren (dies ist nötig wenn Sie eine neue LOCATION erstellen). Sie können Sie wie Sie wollen nennen - wir haben 'demo' verwendet. GRASS erstellt automatisch eine spezielle MAPSET genannt PERMANENT die dazu erstellt wird um Kerndaten für das Projekt, seine Standardausmaße und Koordinatensystemdefinitionen zu speichern (siehe Neteler & Mitasova 2008 in *Literatur und Internetreferenzen*).
14. Prüfen Sie alles, damit alles wie gewünscht ist. Klicken Sie auf [**Abschliessen**].
15. Die neue LOCATION, 'alaska', und zwei MAPSETS, 'demo' und 'PERMANENT' werden erstellt. Die gerade geöffnete Arbeitsumgebung ist 'demo', wie Sie es definiert haben.
16. Beachten Sie, dass einige Werkzeuge des GRASS Plugins grau hinterlegt waren und nun auch zur Verfügung stehen.



Abbildung 16.2: Erstellen einer neuen GRASS LOCATION oder einer neuen MAPSET in QGIS


Wenn das wie eine Menge von Schritten schien ist es doch nicht so schlimm und ein schneller Weg eine LOCATION zu erstellen. Die LOCATION 'alaska' ist jetzt bereit für den Datenimport (siehe Abschnitt *Daten in eine GRASS LOCATION importieren*). Sie können auch die bereits bestehenden Vektor- und Rasterdaten in der Beispiel GRASS LOCATION 'alaska' verwenden, die im QGIS 'Alaska' Datensatz *Beispieldaten* enthalten ist und weiter zu Abschnitt *Das GRASS Vektormodell* gehen.



### 16.3.2 Eine neue GRASS MAPSET erstellen



Ein Benutzer hat nur Schreibzugriff auf eine MAPSET die er oder sie erstellt hat. Das heißt dass Sie neben dem Zugriff zu Ihrer eigenen MAPSET Sie Karten in MAPSETs anderer Benutzer lesen können (und Sie können Ihre lesen) aber Sie nur die Karten in Ihrer eigenen MAPSET verändern und löschen können.

Alle MAPSETs beinhalten eine WIND Datei die die aktuellen Grenzkordinatenwerte und die aktuell ausgewählte Rasterauflösung speichert (siehe Neteler & Mitasova 2008 in *Literatur und Internetreferenzen*, und Abschnitt *Einstellung der GRASS Region*).

1. Starten Sie QGIS und stellen Sie sicher dass das GRASS Plugin geladen ist.
2. Klicken Sie in der GRASS Werkzeugleiste auf das  Neues Mapset Icon um den *Neues Mapset* Assistenten zu öffnen.
3. Wählen Sie den GRASS Datenbankordner (GISDBASE) `grassdata` mit der LOCATION 'alaska' aus wo Sie eine weitere MAPSET genannt 'test' hinzufügen wollen.
4. Klicken Sie auf **[Weiter]**.
5. Wir können diesen Assistenten verwenden um eine neue MAPSET innerhalb einer bestehenden LOCATION zu erstellen oder um dabei auch eine neue LOCATION zu erstellen. Wählen Sie  *Erstelle neue Location* (see [figure\\_grass\\_location\\_2](#)) und klicken Sie **[Weiter]**.
6. Geben Sie den Namen 'test' für die neue `:file:'MAPSET` ein. Unter dem Assistenten sehen Sie eine List bestehender MAPSETs und die entsprechenden Besitzer.
7. Klicken Sie auf **[Weiter]**, prüfen Sie die Einstellungen und drücken Sie dann auf **[Abschließen]**.

### 16.4 Daten in eine GRASS LOCATION importieren

Dieser Abschnitt zeigt ein Beispiel wie man Raster- und Vektordaten in die 'alaska' GRASS LOCATION aus dem QGIS 'Alaska' Datensatz importiert. Daher verwenden wir die Landcover Rasterkarte `landcover.img` und die GML Vektorkarte `lakes.gml` aus dem QGIS 'Alaska' Datensatz (siehe *Beispieldaten*).

1. Starten Sie QGIS und stellen Sie sicher dass das GRASS Plugin geladen ist.
2. Klicken Sie in der GRASS Werkzeugleiste das  Mapset öffnen Icon um den *Wählen Sie ein GRASS Mapset* Assistenten zu starten.
3. Wählen Sie als GRASS Datenbank den Ordner `grassdata` aus dem QGIS Alaskadatensatz, als LOCATION 'alaska', als MAPSET 'demo' aus und klicken Sie **[OK]**.
4. Nun klicken Sie auf das Icon  GRASS-Werkzeugkiste öffnen, damit die GRASS Werkzeuge (siehe Abschnitt *Die GRASS Werkzeugkiste*) zur Verfügung stehen.
5. Um die Rasterkarte `landcover.img` zu importieren, drücken Sie auf das Modul `r.in.gdal` im *Modulbaum* Reiter. Diese GRASS Module ermöglicht es, GDAL-unterstützte Rasterdateien in eine GRASS LOCATION zu importieren.
6. Browsen Sie zum Ordner `raster` im QGIS 'Alaska' Datensatz und wählen Sie die Datei `landcover.img`.
7. Definieren Sie als Rasterausgabename `landcover_grass` und klicken Sie **[Starten]**. Im Reiter *Ergebnis* können Sie den gerade laufenden GRASS Befehl `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass` sehen.
8. Wenn **Erfolgreich beendet** erscheint klicken Sie **[Ergebnis visualisieren]**. Der `landcover_grass` Rasterlayer ist jetzt in GRASS importiert und wird in der QGIS Kartenansicht visualisiert.
9. Um die GML Vektordatei `lakes.gml` zu importieren klicken Sie das Modul `v.in.ogr` im *Modulbaum* Reiter. Dieses GRASS Modul ermöglicht den Import von OGR-unterstützten Vektordateien in eine GRASS LOCATION. Der Moduldialog für `v.in.ogr` erscheint.

10. Browsen Sie zum Ordner `gml` im |qg| 'Alaska' Datensatz und wählen Sie die Datei `:file:'lakes.gml` als OGR Datei aus.
11. Definieren Sie `lakes_grass` als Vektorausgabennamen und klicken Sie **[Starten]**. Sie brauchen sich in diesem Beispiel nicht um andere Optionen kümmern. Im Reiter *Ergebnis* sehen Sie den aktuell laufenden GRASS Befehl `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes\_grass`.
12. Wenn **Erfolgreich beendet** erscheint klicken Sie **[Ergebnis visualisieren]**. Der `lakes_grass` Vektorlayer ist jetzt in GRASS importiert und wird in der QGIS Kartenansicht visualisiert.

## 16.5 Das GRASS Vektormodell

Es ist wichtig das GRASS Vektordatenmodell vor dem Digitalisieren zu verstehen.

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 Geometrielemente können innerhalb ein und dem selben Datensatz enthalten sein. Sie werden in unterschiedlichen 'Ebenen' innerhalb von QGIS dargestellt. Auch wenn es möglich ist, Geometrielemente zu mischen, so ist es eigentlich unüblich und wird normalerweise auch in GRASS GIS nur selten verwendet. Etwa bei Netzwerkanalysen. Im Normalfall sollten Sie versuchen, unterschiedliche Geometrietyten in unterschiedlichen Datensätzen (Layern) zu speichern.

Es ist auch möglich, unterschiedliche Inhalte des gleichen Geometrietyps in verschiedenen Ebenen eines Vektorlayers 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 Attributtabelle, die über ihre Ebene angesprochen wird. 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.

Die 'Ebene' eines jeden Objektes wird in GRASS intern als 'layer' bezeichnet. 'Layer' ist die Nummer die definiert ob es mehr als einen Layer innerhalb des Datensatzes gibt (z.B. ob die Geometrie Wald oder See ist). Vorerst kann dies nur eine Nummer sein. In Zukunft wird GRASS auch Namen als Felder in der Benutzeroberfläche unterstützen.

Attribute können innerhalb der GRASS LOCATION als dBase oder SQLite3 oder in externen Datenbanktabellen abgelegt werden, z.B. 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.


---

### Tipp: Das GRASS Vektormodell verstehen

Der beste Weg, etwas über das GRASS Vektordatenmodell und seine Fähigkeiten zu erfahren, ist das Studieren eines der vielen verfügbaren GRASS-Tutorien, wo dieses Thema vertieft behandelt wird. Unter <http://grass.osgeo.org/documentation/manuals/> sind eine Vielzahl weiterer Informationsquellen, Bücher und Tutorien in unterschiedlichen Sprachen vorhanden.

---

## 16.6 Einen neuen GRASS Vektorlayer erstellen

Um einen neuen GRASS Vektorlayer mit dem GRASS plugin zu erstellen klicken Sie das  Neuen GRASS-Vektorlayer anlegen Werkzeugleistenicon. Geben Sie einen Namen in das Textfenster ein und Sie können mit dem Digitalisieren von Punkt-, Linien- oder Polygoneometrien beginnen indem Sie wie in Abschnitt *Digitalisieren und Editieren eines GRASS Vektorlayers* verfahren.

GRASS GIS erlaubt es aufgrund des topologischen Datenmodells, die verschiedenen Geometrietypen (Punkt, Linie und Fläche) in einem Vektorlayer abzuspeichern. Aus diesem Grund ist es nicht notwendig im Vorfeld einen Geometrietyp festzulegen. Dies unterscheidet sich von der Erstellung eines neuen Shapefile in QGIS, denn Shapefiles verwenden das Simple Feature Vektormodell (siehe Abschnitt *Neue Vektorlayer erstellen*).


---

**Tipp: Erstellen einer Attributtabelle für einen neuen GRASS Vektorlayer**

Wenn Sie für ihre digitalisierten Geometrien auch eine Attributtabelle erstellen wollen, müssen Sie nach dem Anlegen des GRASS-Vektorlayers eine Attributtabelle mit entsprechenden Spalten erstellen, bevor Sie mit dem Digitalisieren beginnen (siehe [figure\\_grass\\_digitizing\\_5](#)).

---

## 16.7 Digitalisieren und Editieren eines GRASS Vektorlayers

Die Digitalisierwerkzeuge für GRASS-Vektorlayer werden über den Knopf  GRASS-Vektorlayer bearbeiten in der GRASS-Werkzeugleiste gestartet. Dazu müssen Sie den zu bearbeitenden Layer in der Legende auswählen, bevor Sie auf den Editier-Knopf drücken. Abbildung [figure\\_grass\\_digitizing\\_2](#) zeigt den Dialog für die GRASS Digitalisierung. Die einzelnen Werkzeuge werden im folgenden Kapitel beschrieben.

---

**Tipp: Polygone in GRASS digitalisieren**

Wenn Sie ein Polygon innerhalb eines GRASS Vektorlayers erstellen wollen, digitalisieren Sie zuerst die Boundary der Fläche mit der Moduseinstellung 'Keine Kategorie'. Danach fügen Sie einen Zentroid (Labelpunkt) mit der Moduseinstellung 'Nächst folgender Kategoriewert' hinzu. Der Grund ist, dass in topologischen Layern die Attributinformationen einer Fläche immer mit dem Zentroiden und nicht mit der Grenze verknüpft werden.

---

**Werkzeugleiste**

In [figure\\_grass\\_digitizing\\_1](#) sehen Sie die GRASS Digitalisieren Werkzeugleisten Icons, die vom GRASS Plugin zur Verfügung gestellt werden. Tabelle [table\\_grass\\_digitizing\\_1](#) erklärt die verfügbaren Funktionalitäten.



Abbildung 16.3: GRASS Digitalisieren Werkzeugleiste

Icon	Werkzeug	Zweck
	Neuer Punkt	Digitalisiert neuen Punkt
	Neue Linie	Digitalisiert neue Linie
	Neue Grenze	Digitalisiert neue Grenze (zum Beenden ein neues Werkzeug wählen)
	Neuer Zentroid	Digitalisiert neuen Zentroiden (Labelpunkt für eine existierende Fläche)
	Verschiebe Vertex	Wählt einen Stützpunkt einer existierenden Linie oder Umrandung und setzt diesen an eine neue Position
	Vertex hinzufügen	Fügt einen Stützpunkt zu einer existierenden Linie hinzu
	Lösche Vertex	Löscht einen Stützpunkt von einer existierenden Linie (Bestätigung der Auswahl durch einen weiteren Klick nötig)
	Verschiebe Element	Wählt existierende Grenze, Linie oder Zentroiden und verschiebt sie an eine neue Position
	Unterteile Linie	Unterteile eine bestehende Linie in zwei Teile
	Element löschen	Löscht eine existierende Geometrie inklusive des Eintrags in der Attributtabelle (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)
	Schließen	Beendet die Bearbeitung (und aktualisiert die Topologie anschließend)

Tabelle GRASS Digitizing 1: GRASS Digitalisierwerkzeuge

### Reiter Kategorie

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

- **Modus:** Der Kategoriewert der auf neuen Geometrieelemente angewendet wird.
  - Nächst folgender Kategoriewert - Der nächste noch nicht verwendete Kategoriewert wird auf das Geometrieelement angewendet
  - Manueller Eintrag - Definieren Sie den Kategoriewert für das Geometrieelement manuell im 'Kategorie' Eingabefeld.
  - Keine Kategorie - Wendet keinen Kategoriewert auf das Geometrieelement an. Dies wird z.B. für Flächengrenzen verwendet, da die Kategoriewerte über den Zentroiden angebunden werden.
- **Kategorie** - Die Nummer (ID) die an jedes digitalisierte Geometrieelement angehängt wird. Sie wird verwendet um jedes Geometrieelement mit seinen Attributen zu verbinden.
- **Feld (Layer)** - Jedes Geometrieelement kann mit mehreren Attributtabellen, die verschiedene GRASS Geometrielayer verwenden, verknüpft werden. Die voreingestellte Layernummer ist 1.

---

### Tipp: Einen zusätzlichen GRASS 'layer' mit lqgl erstellen

Wenn Sie gerne mehr Layer zu Ihrem Datensatz hinzufügen wollen tragen Sie einfach eine neue Nummer im 'Feld (Layer)' Eingabefeld ein und drücken Sie Eingabe. Im Reiter Tabelle können Sie Ihre neue Tabelle, die dann an den neuen Layer angebunden ist, erstellen.

---

### Reiter Einstellungen

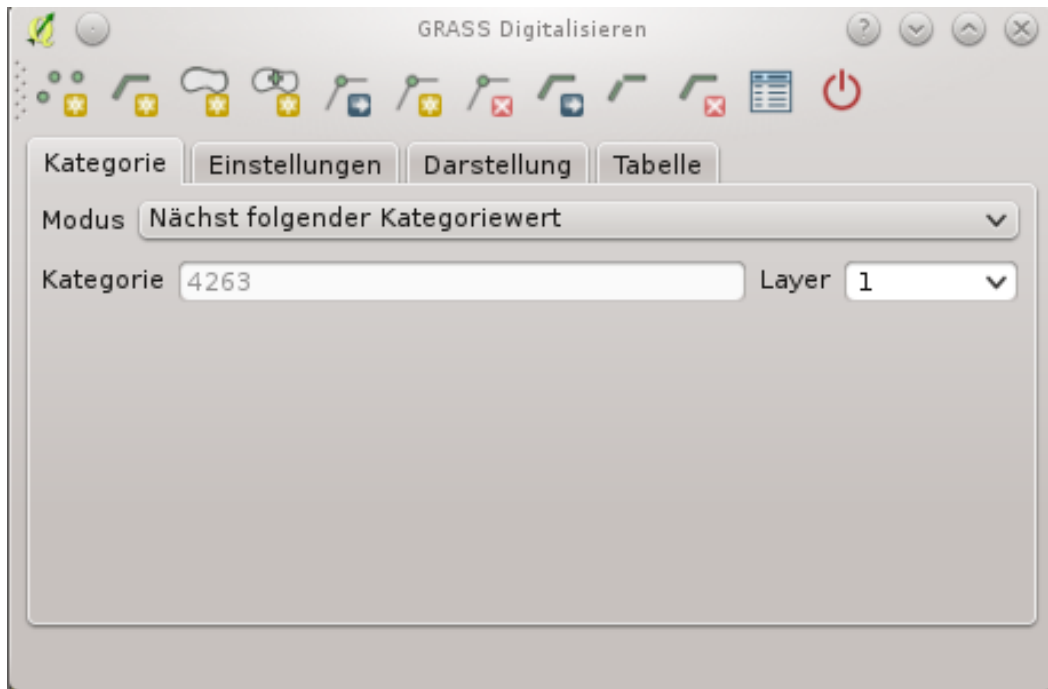


Abbildung 16.4: GRASS Digitalisieren Kategorie Reiter

Der Reiter *Einstellungen* erlaubt das Setzen der Fangtoleranz in Bildschirmpixeln. Dies ist der Schwellenwert in Pixeln, innerhalb dessen neu digitalisierte Knotenpunkte an vorhandene Knoten gesnappt werden. Dies hilft, Lücken oder Überlagerungen zwischen Objekten zu vermeiden. Der Standardwert ist auf 10 Pixel eingestellt.

#### Darstellung Reiter

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

#### Reiter Tabelle

Der Reiter *Tabelle* stellt Informationen über die Datenbanktabelle für einen gegebenen 'layer' bereit. Hier können Sie neue Spalten in eine bestehende Attributtabelle einfügen oder eine neue Datenbanktabelle für einen neuen GRASS Vektorlayer erstellen (siehe Abschnitt *Einen neuen GRASS Vektorlayer erstellen*).

---


#### Tipp: GRASS Schreibberechtigung

Sie müssen der Besitzer der GRASS MAPSET, die Sie bearbeiten wollen, sein. Es ist unmöglich Datenlayer in einer MAPSET die Ihnen nicht gehört zu bearbeiten, auch wenn Sie Schreibrechte darauf haben.

---

## 16.8 Einstellung der GRASS Region

Die Region Einstellung (Einstellung eines räumlichen Arbeitsfensters) in GRASS ist wichtig für das Arbeiten mit Rasterlayern. Vektoranalysen sind standardmäßig nicht auf eine definierte Region Definition begrenzt. Aber alle neu erstellten Raster werden die räumliche Ausdehnung und Auflösung der gerade definierten GRASS Region haben, unabhängig von Ihrer ursprünglichen Ausdehnung und Auflösung. Die aktuelle GRASS Region ist in der \$LOCATION/\$MAPSET/WIND Datei gespeichert und Sie definiert die Nord-, Süd-, Ost- und Westgrenze, die Anzahl von Spalten und Reihen sowie die horizontale und vertikale räumliche Auflösung.

Es ist möglich die Visualisierung der GRASS Region in der QGIS Kartenansicht ein- und auszuschalten indem Sie den  Knopf verwenden.


Mit dem  Icon können Sie einen Dialog zum Ändern der aktuellen Region und der



Abbildung 16.5: GRASS Digitalisieren Einstellungen Reiter

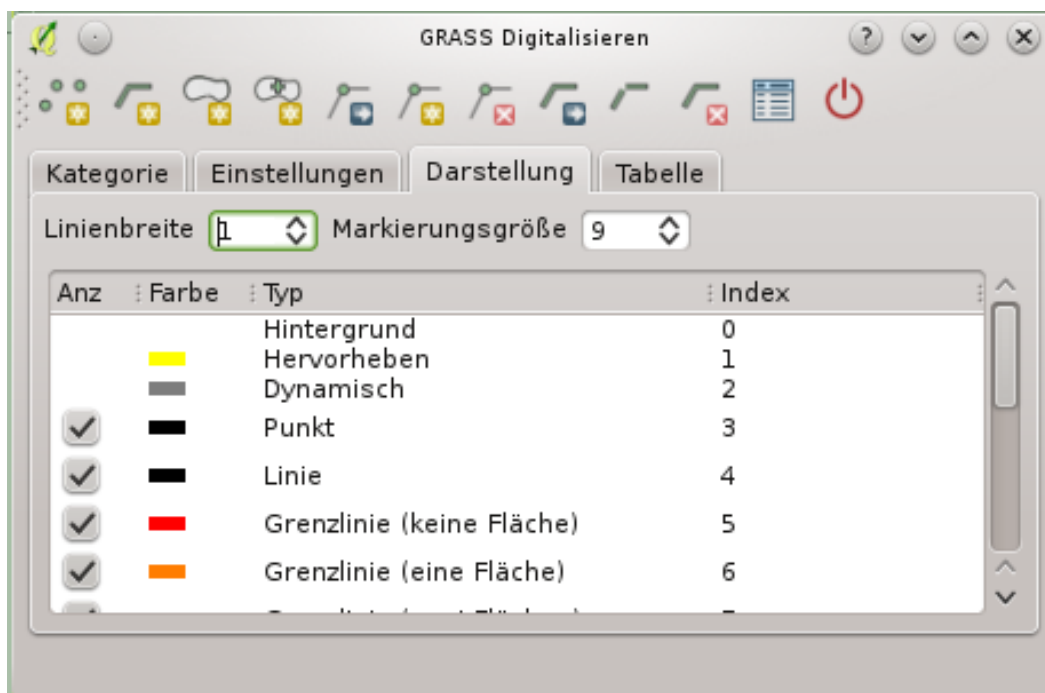


Abbildung 16.6: GRASS Digitalisieren Darstellung Reiter




Abbildung 16.7: GRASS Digitalisieren Tabelle Reiter

Darstellung des GRASS-Region Rechtecks im QGIS Kartenfenster öffnen. Geben Sie die neuen Region Ausmaße und Auflösung eine und klicken Sie **[OK]**. Der Dialog ermöglicht Ihnen auch aktiv eine neue Region mit Ihrer Maus im QGIS Kartenfenster auszuwählen. Klicken Sie zu diesem Zweck mit der linken Maustaste in das QGIS Kartenfenster, öffnen Sie ein Rechteck, schließen Sie es mit Hilfe der linken Maustaste und klicken Sie **[OK]**.

Das GRASS Modul `g.region` bietet viele weitere Optionen zur Einstellung der passenden Ausdehnung und Auflösung der Region für die Rasteranalyse. Das Modul `g.region` können Sie über die GRASS Werkzeugkiste nutzen, wie in Kapitel *Die GRASS Werkzeugkiste* beschrieben.

## 16.9 Die GRASS Werkzeugkiste

Die  GRASS-Werkzeugkiste ermöglicht es, GRASS Module auf Daten innerhalb einer ausgewählten GRASS LOCATION und MAPSET anzuwenden. Dazu muss im Vorfeld eine GRASS LOCATION und MAPSET geöffnet werden, in der Sie Schreibrechte besitzen. Dies ist normalerweise garantiert, wenn Sie die MAPSET selbst erstellt haben und notwendig, damit die Ergebniskarten der Raster- und Vektoranalysen in der ausgewählten LOCATION und `:file:MAPSET` gespeichert werden können.

### 16.9.1 Arbeiten mit GRASS Modulen

Die GRASS Shell der Werkzeugkiste bietet Zugriff auf fast alle (mehr als 300) GRASS Module über die Kommandozeile. Um eine benutzfreundlichere Umgebung zu bieten, sind davon etwa 200 Module graphisch auswählbar und bieten einen Dialog in Form eines zusätzlichen Reiters in der Werkzeugkiste.

Eine vollständige Liste von GRASS Modulen die in der grafischen Werkzeugkiste der QGIS Version 2.6 verfügbar sind können Sie im GRASS Wiki unter [http://grass.osgeo.org/wiki/GRASS-QGIS\\_relevant\\_module\\_list](http://grass.osgeo.org/wiki/GRASS-QGIS_relevant_module_list) finden.

Es ist außerdem möglich, die GRASS Werkzeugkiste anzupassen und weitere Module zu integrieren. Die Herangehensweise ist in Abschnitt *Anpassen der Module* beschrieben.

Wie in Abbildung [figure\\_grass\\_toolbox\\_1](#) zu sehen, können Sie nach dem passenden GRASS Modul in dem Reiter *Modulbaum* nachschauen oder im Reiter *Modulliste* suchen.



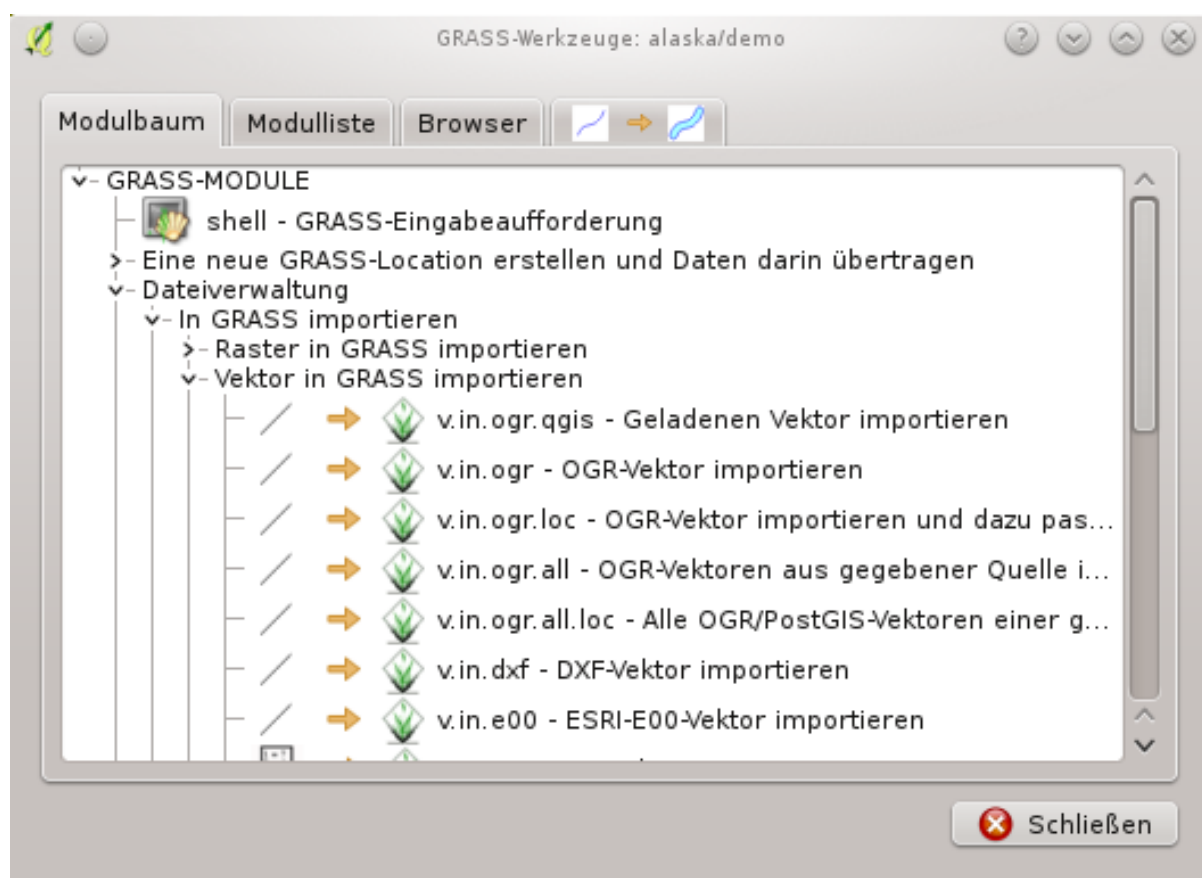


Abbildung 16.8: GRASS Werkzeuge und Modulbaum 



Wenn Sie auf das grafische Icon eines Modules klicken, öffnet sich ein neuer Moduldialog mit drei Reitern *Optionen*, *Ergebnis* und *Handbuch*.

### Optionen

Der Reiter *Optionen* stellt Ihnen einen Moduldialog in vereinfachter Form, in dem sie gewöhnlicherweise einen Raster- oder Vektorlayer, der in der QGIS Kartenansicht visualisiert ist, auswählen können und weitere modul-spezifische Parameter, um die Module zum Laufen zu bringen, eingeben können.

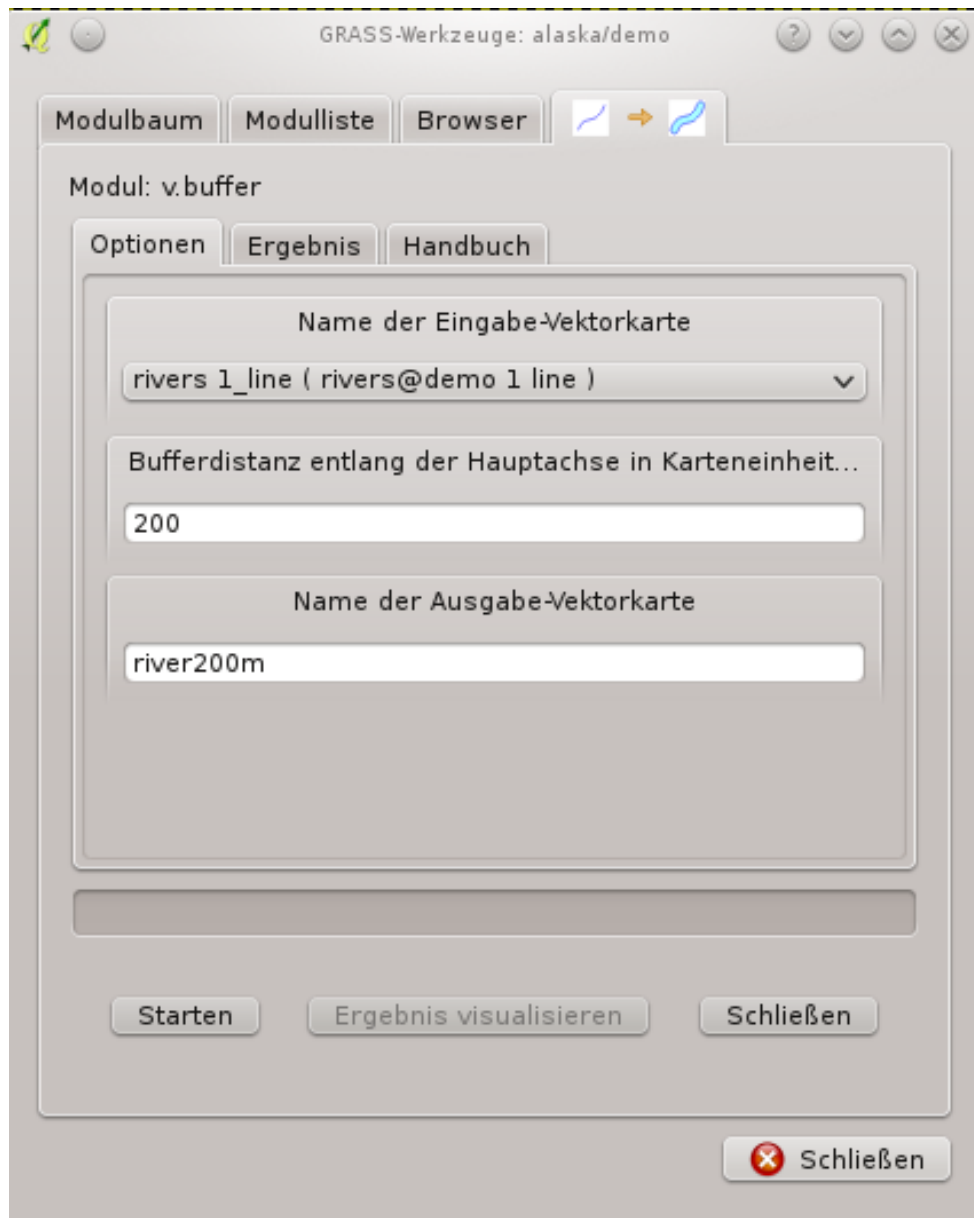


Abbildung 16.9: GRASS Werkzeuge Modul Optionen 

Die zur Verfügung gestellten Modulparameter sind oftmals nicht vollständig um den Dialog einfach zu halten. Wenn Sie weitergehende Modulparameter und -flags verwenden wollen müssen Sie die GRASS Shell starten und das Programm in der Kommandozeile benutzen.

Eine neue Funktion seit der QGIS Version 1.8.0 ist der Knopf *Fortgeschrittene Optionen einblenden* unterhalb der vereinfachten Moduldialoge im Reiter *Optionen*. Momentan ist diese Funktion nur für das Modul *v.in.ascii* umgesetzt. Es ist aber zukünftig für weitere Module vorgesehen und wird so die kompletten Funktionen der GRASS Module grafisch bereitstellen, ohne die GRASS Shell und somit die Kommandozeile benutzen zu müssen.

### Ergebnis

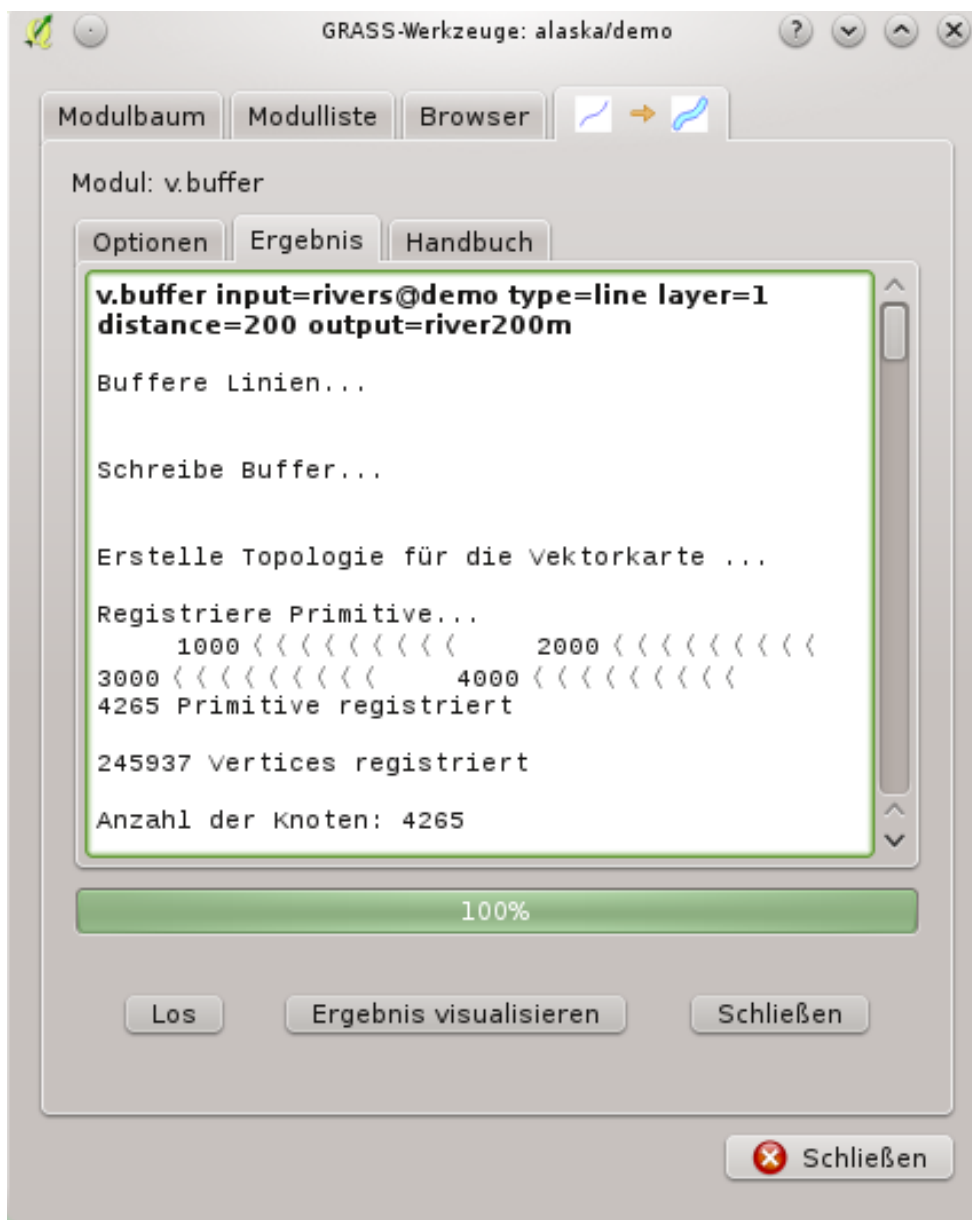


Abbildung 16.10: GRASS Werkzeuge Modul Ergebnis 

Der Reiter *Ergebnis* stellt die Ausgabe des Moduls zur Laufzeit 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 Ausgabertext `Erfolgreich beendet`.

**Handbuch**

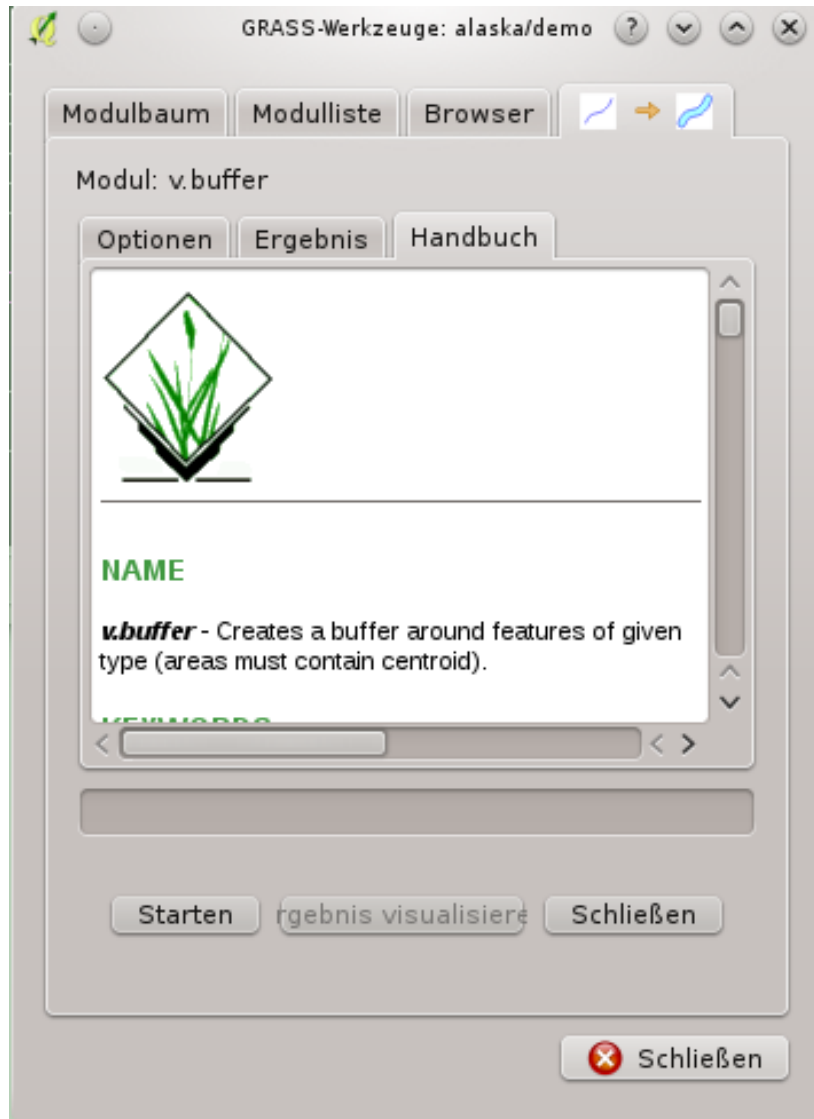


Abbildung 16.11: GRASS Werkzeuge Modul Handbuch 🐧

Der Reiter *Handbuch* zeigt die HTML Hilfeseite der GRASS Module. Sie können es zum Überprüfen weitergehender Modulparameter oder -flags oder um vertiefere Kenntnisse über den Anwendungszweck der Module zu erwerben benutzen. Am Ende jeder Modul Handbuchseite sehen Sie weiterführende Links zum `Main Help index`, dem `Thematic index` und dem `Full index`. Diese Links stellen die gleiche Information wie das Modul `g.manual` zur Verfügung.

**Tipp: Ergebnisse direkt anzeigen**




Wollen Sie Ihre Ergebnisse direkt in der Kartenansicht ansehen, nutzen Sie den Knopf 'Ergebnis visualisieren' im unteren Bereich des jeweiligen Modulreiters

## 16.9.2 GRASS Beispielanwendung

Die folgenden Beispiele sollen die Anwendung verschiedener GRASS Module demonstrieren.

### Höhenlinien aus einem DGM erstellen

Im ersten Beispiel wird eine Höhenlinienkarte aus einem Höhenraster (DEM) erstellt. Hier wird angenommen dass Sie die Alaska LOCATION wie in Abschnitt *Daten in eine GRASS LOCATION importieren* eingerichtet haben.

- Als erstes öffnen Sie die Location, indem Sie auf das  Mapset öffnen Icon klicken und dann die Alaska Location auswählen.
- Nun laden Sie den `gtopo30` Rasterlayer, indem Sie auf das Icon  GRASS-Rasterlayer hinzufügen klicken und den Layer `gtopo30` aus der Alaska Location auswählen und laden.
- Als nächstes drücken Sie auf das Icon  GRASS-Werkzeugkiste öffnen.
- In der Liste von Werkzeugkategorien doppelklicken Sie auf *Raster* → *Oberflächenverwaltung* → *Vektorkonturlinien erzeugen*.
- Jetzt wird ein einfacher Klick auf das Werkzeug **r.contour** einen Werkzeugdialog wie oben beschrieben öffnen (siehe *Arbeiten mit GRASS Modulen*). Die `gtopo30` Rasterdatei sollte als *Name der Eingabe-Rasterkarte* erscheinen.
- Geben Sie im Feld *Abstand zwischen den Kontourintervallen*  den Wert 100 an. (Dadurch werden Höhenlinien in einem Abstand von 100m erstellt.)
- In das Feld *Name der Vektorausgabekarte* geben Sie den Namen `hoehen_100` an.
- Klicken Sie [**Starten**] um den Prozess zu starten. Warten Sie mehrere Momente bis die Nachricht *Erfolgreiche beendet* im Ergebnisfenster auftaucht. Klicken Sie dann [**Ergebnis visualisieren**] und [**Schließen**].

Da die aktuelle GRASS Region ziemlich groß ist, kann es eine Weile dauern, bis der Layer vollständig dargestellt wird. Danach können Sie noch die Layereigenschaften verändern und eine Farbe für die Linien auswählen, die sich deutlich vom Höhenmodell unterscheidet, siehe *Vektorlayereigenschaften*.

Als nächstes zoomen Sie in einen bergigen Bereich im Zentrum Alaskas. Wenn Sie weit genug in die Karte hineingezoomt sind, werden Sie erkennen, dass die Höhenlinien teilweise sehr eckig erscheinen. Um das Erscheinungsbild zu optimieren, bietet GRASS ein Modul mit dem Namen **v.generalize**. Dabei wird mit Hilfe des Douglas Peucker Algorithmus und der einer Reduktion der Stützpunkte eine Glättung der Linien erreicht, ohne die Geometrien zu zerstören. Da der Ergebnislayer weniger Stützpunkte hat, ist er auch kleiner und kann schneller geladen werden. Die Analyse wird z.B. angewendet, wenn man sehr detaillierte Daten nur in einem kleinen Maßstab anzeigen möchte.

---

#### Tipp: Geometrien in QGIS vereinfachen

Beachten Sie dass das QGIS fTools Plugin über ein *Geometrien vereinfachen* → Werkzeug verfügt das genauso wie der GRASS **v.generalize** Douglas-Peucker Algorithmus funktioniert.

---

In diesem Beispiel wollen wir nun aber etwas anderes erreichen. Die Höhenlinien, die wir mit **r.contour** erstellt haben, zeigen teilweise sehr scharfe Winkel, die wir glätten möchten. Unter den Algorithmen des Moduls **v.generalize** befindet sich auch der Chaikens-Algorithmus, der exakt das macht, was wir möchten (auch Hermite Splines). Achten Sie aber darauf, dass es passieren kann, dass Stützpunkte nicht nur eliminiert sondern auch **hinzugefügt** werden können. Dadurch kann der Layer wieder langsamer geladen werden.

- Öffnen Sie die GRASS Werkzeugkiste und doppelklicken Sie die Kategorien *Vektor* → *Karte entwickeln* → *Generalisierung* und klicken Sie dann auf das **v.generalize** Modul um sein Optionsfenster zu öffnen.
- Stellen Sie sicher, dass 'hoehen\_100' als Vektorlayer in Feld *Name der Vektoreingabekarte* erscheint.

- Aus der Liste möglicher Algorithmen wählen Sie nun Chaiken's. Belassen Sie alle weiteren Optionen wie sie sind und gehen Sie zum Ende des Dialogs, wo Sie als *Name der Vektorausgabekarte* 'hoe-hen\_100\_smooth' angeben. Drücken Sie nun auf [**Starten**].
- Der Prozess dauert eine Weile. Sobald Erfolgreich beendet im Reiter *Ergebnis* erscheint, drücken Sie wieder auf [**Ergebnis visualisieren**] und dann auf [**Schließen**].
- Ändern Sie nun auch die Farbe des neuen Layers, damit er sich deutlich von dem Höhenmodell und den zuvor berechneten Höhenlinien abhebt. Sie werden erkennen, dass die Kanten der neuen Höhenlinien wesentlich weicher gezeichnet sind.

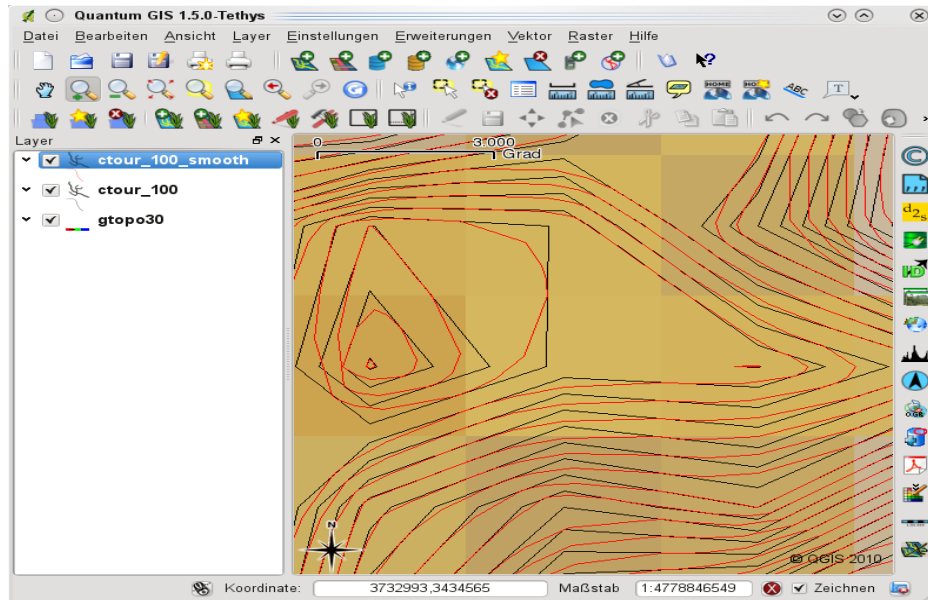



Abbildung 16.12: GRASS Modul v.generalize zum Glätten einer Vektorkarte 

**Tipp: Vektorlinien glätten mit dem GRASS Modul v.generalize**

Die oben beschriebene Anwendung kann auch in anderen Situationen verwendet werden. Wenn Sie z.B. eine Rasterkarte mit Niederschlagswerten haben, können Sie mit r.contour einen Isohyetallayer (konstanter Niederschlag) erstellen.

**Erstellen eines 3D Schummerungseffekts**

Es gibt verschiedene Methoden, um Höhenlayer anzuzeigen und ihnen einen 3D Schummerungseffekt zu verleihen. Der Gebrauch von Höhenlinien ist eine populäre Methode, die häufig angewendet wird, um topographische Karten zu erstellen. Eine andere Möglichkeit, um einen 3D Effekt zu erzeugen ist, bietet das Hillshading. Der Effekt basiert dabei auf einem Höhenmodell. Dabei wird zuerst die Hangneigung und -richtung der Zellen bestimmt und dann durch die Simulation des Sonnenstandes eine Reflexionswert erzeugt. Dadurch werden der Sonne zugewandte Bereiche aufgehellt und der Sonne abgewandte Bereiche (im Schatten) dunkler dargestellt.

- Fangen Sie mit diesem Beispiel an indem Sie zuerst den Rasterlayer gtopo30 laden. Öffnen Sie die GRASS-Werkzeugkiste und unter der Raster Kategorie wechseln Sie in den Bereich *Räumliche Analysen* → *Geländeanalyse*.
- Nun klicken Sie auf **r.shaded.relief**, um den Modulreiter zu öffnen.
- Ändern Sie den Wert im Feld *Winkel der Sonne in Grad östlich von der Nordrichtung*  270 auf 315.
- Geben Sie als Name der Schummerungskarte den Namen gtopo30\_shade an und klicken Sie auf [**Starten**].
- Nachdem die Karte berechnet wurde, visualisieren Sie sie und setzen Sie die Farbe des Rasterlayers auf Graustufen.

- Um die Schummerung und das Höhenmodell `gtopo30` zusammen zu sehen, ziehen Sie die Schummerungskarte in der Legende unter das Höhenmodell. Öffnen Sie dann den Dialog Layereigenschaften der Karte `gtopo30`, und ändern Sie die Transparenz im Reiter *Transparenz* auf den Wert 25%.

Die Höhenkarte `gtopo30` wird nun als Farbkarte leicht transparent über der Schummerungskarte angezeigt. Dadurch entsteht ein visueller 3D Effekt. Um den Unterschied besser zu erkennen, wechseln Sie über das Kontrollkästchen den Anzeigemodus der Schummerungskarte in der Legende und wieder zurück.

### Die GRASS Kommandozeile verwenden


Das GRASS Plugin in QGIS stellt die GRASS Module oftmals in vereinfachter Form und auch nicht vollständig bereit. Es ist also grundsätzlich für Anwender gestaltet, die sich nicht so gut mit GRASS und all seinen Fähigkeiten auskennen oder nur einfache Analysen durchführen wollen. Daher werden in den grafischen Moduldialogen oftmals nicht alle Optionen und Parameter, die das GRASS Modul bieten bereitgestellt, um den Umgang einfacher und intuitiver zu gestalten. Wer tiefer in GRASS einsteigen möchte, er hat die Möglichkeit, sämtliche Funktionalitäten und Module über die GRASS Kommandozeile (GRASS Shell) anzusprechen. In dem folgenden Beispiel soll eine zusätzliche Option des Moduls **r.shaded.relief** angesprochen werden, die nur über die Kommandozeile genutzt werden kann.

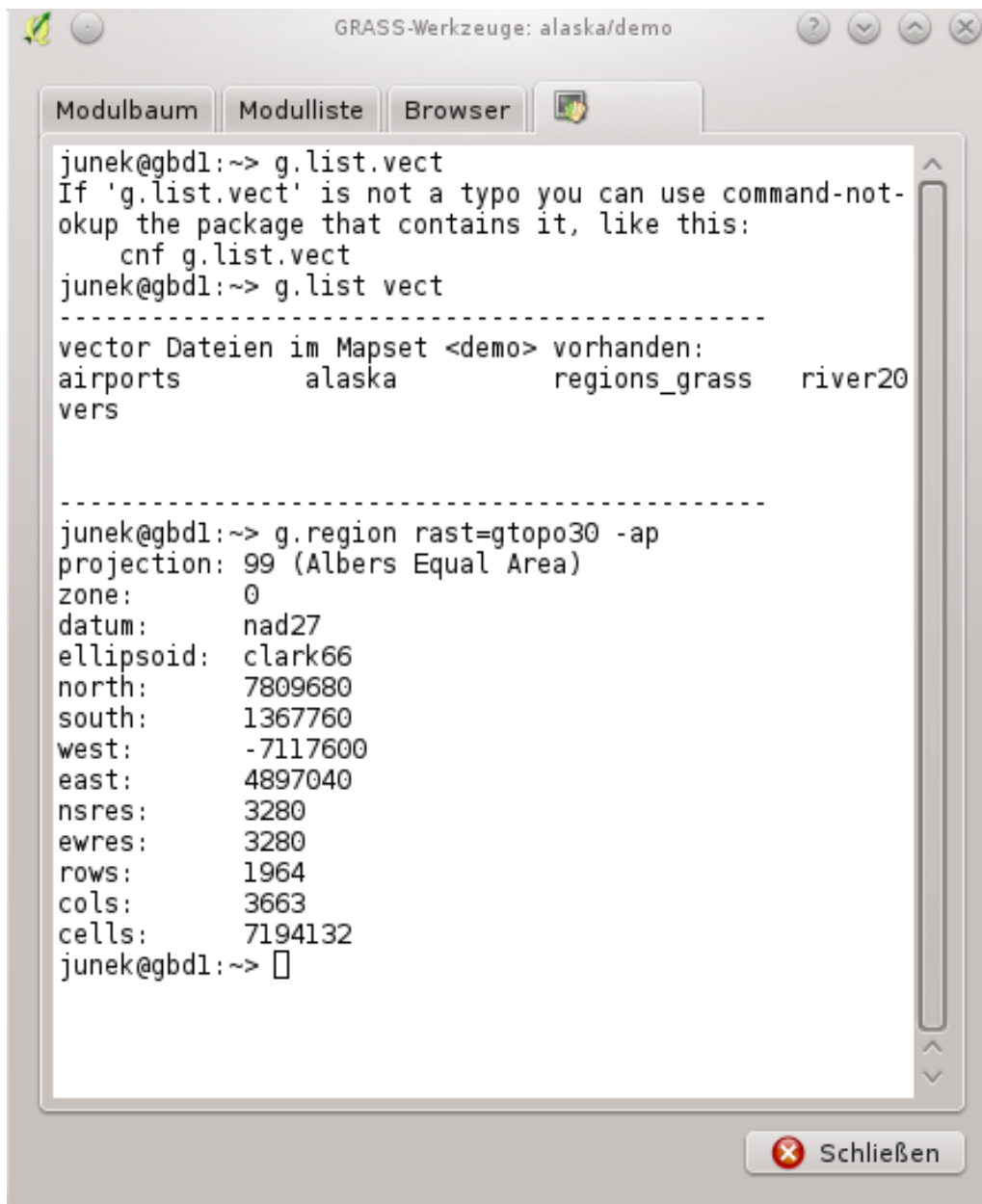
Das Modul **r.shaded.relief** stellt einen zusätzliche Parameter `zmult` bereit, über den der Höhenwert relativ zu den X-Y Werten multipliziert werden kann. Dadurch wird der Schummerungseffekt noch prägnanter.

- Laden Sie das Höhenmodell `gtopo30` wie im vorherigen Beispiel. Öffnen Sie die GRASS-Werkzeugkiste und klicken auf die GRASS Shell. In das Kommandozeilenfenster tippen Sie folgenden Befehl `r.shaded.relief map=gtopo30 shade=gtopo30_shade2 azimuth=315 zmult=3` und drücken dann die Taste [Enter].
- Wenn die Berechnung abgeschlossen ist, wechseln Sie in den Reiter *Browser* und Doppelklicken Sie auf die neu erstellte Karte `gtopo30_shade2`, um Sie in QGIS anzuzeigen.
- Wie oben erklärt verschieben Sie die Schummerungskarte unter das `gtopo30` Raster in der Legende, überprüfen Sie dann die Transparenz des farbigen `gtopo30` Layers. Sie sollten dabei erkennen, dass der Schummerungseffekt stärker verglichen mit der ersten Schummerungskarte ist.

### Rasterstatistik auf Basis eines Vektorlayer berechnen

Das folgende Beispiel zeigt, wie man univariate Statistik für Rasterwerte innerhalb von Vektorpolygonen berechnen kann und die Werte in neue Attributspalten des Vektorlayers hinzufügt.

- Verwenden Sie wieder die Alaska Location (siehe Abschnitt *Daten in eine GRASS LOCATION importieren*), und importieren Sie das Shapefile `trees` aus dem Ordner `shapefiles` nach GRASS.
- Jetzt ist ein Zwischenschritt nötig: es müssen Zentroiden zu der importierten Baumkarte hinzugefügt werden um es zu einem kompletten GRASS Flächenvektor (der sowohl Umrandungen als auch Zentroide enthält) zu machen.
- Wählen Sie aus der Werkzeugkiste *Vektor* → *Karte entwickeln* → *Objekte verwalten* und öffnen Sie das Modul **v.centroids**.
- Geben Sie als *Name für die Ausgabe-Vektorkarte* 'forest\_areas' an und starten Sie das Modul.
- Nun laden Sie den neuen Layer `forest_areas` und visualisieren Sie die verschiedenen Waldtypen in verschiedenen Farben - deciduous, evergreen und mixed. Dazu öffnen Sie den Dialog *Eigenschaften* des Layers, wechseln zum Reiter *Darstellung*, wählen  'Eindeutiger Wert' und setzen das *Klassifikationsfeld* auf 'VEGDESC' (siehe auch die Beschreibung des Reiters *Darstellung* in Abschnitt *Menü Stil* des Vektordatenabschnitts).
- Als nächstes öffnen Sie wieder die GRASS-Werkzeugkiste und wechseln nach *Vektor* → *Vektor mit anderen Karten aktualisieren*.
- Klicken Sie auf das **v.rast.stats** Modul. Geben Sie `gtopo30` und `forest_areas` ein.
- Nun fehlt nur noch ein Parameter: Geben Sie als *Spaltenpräfix* `elev` an, und klicken Sie dann auf [Starten]. Die Berechnung wird ziemlich lange dauern (wahrscheinlich bis zu zwei Stunden).




The screenshot shows a terminal window titled "GRASS-Werkzeuge: alaska/demo". The terminal displays the following commands and output:

```
junek@gbd1:~> g.list.vect
If 'g.list.vect' is not a typo you can use command-not-
okup the package that contains it, like this:
  cnf g.list.vect
junek@gbd1:~> g.list vect
-----
vector Dateien im Mapset <demo> vorhanden:
airports      alaska      regions_grass  river20
vers

-----

junek@gbd1:~> g.region rast=gtopo30 -ap
projection: 99 (Albers Equal Area)
zone:      0
datum:     nad27
ellipsoid: clark66
north:     7809680
south:     1367760
west:      -7117600
east:      4897040
nsres:     3280
ewres:     3280
rows:      1964
cols:      3663
cells:     7194132
junek@gbd1:~> []
```

A "Schließen" button with a red 'X' icon is visible at the bottom right of the terminal window.

Abbildung 16.13: Die GRASS Shell, Modul r.shaded.relief 



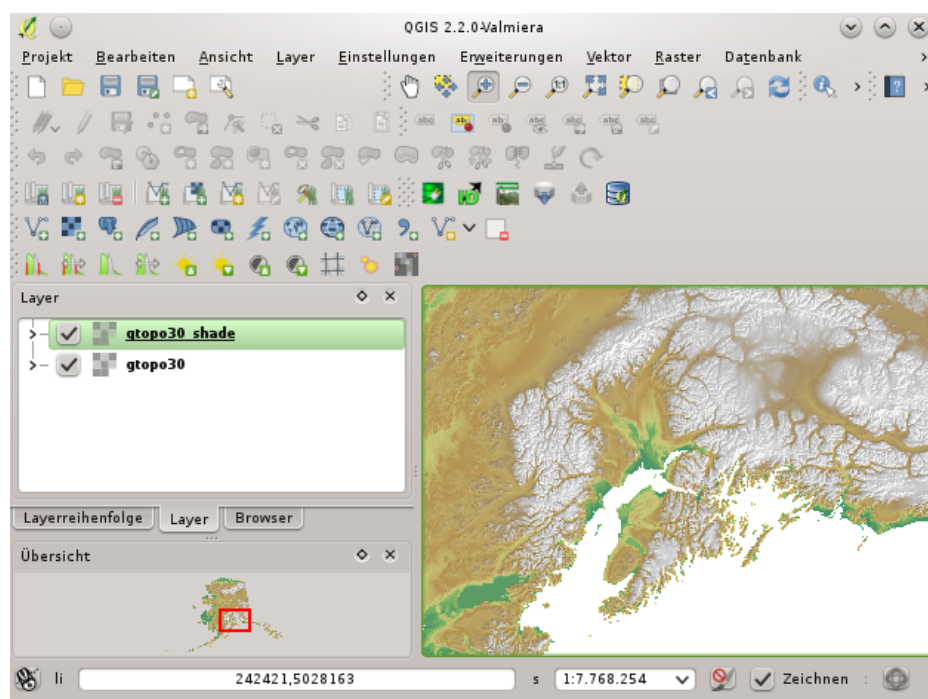



Abbildung 16.14: Darstellen einer Schummering erstellt mit dem GRASS Modul `r.shaded.relief` 







- Schließlich öffnen Sie den Layer `forest_areas` erneut und lassen Sie sich die Attributtabelle anzeigen. Dort gibt es nun weitere Spalten einschließlich `elev_min`, `elev_max`, `elev_mean` usw. für jedes Wald-polygon.

### 16.9.3 Der GRASS Datei-Browser

Eine weitere interessante Funktion in den GRASS-Werkzeugen ist der GRASS LOCATION Browser. In `figure_grass_module_7_` können Sie die aktuelle `:file:LOCATION` mit den dazu gehörigen MAPSETs sehen.

In der linken Hälfte des Datei-Browsers können Sie sich alle in der aktuellen LOCATION enthaltenen MAPSETs anschauen. In der rechten Hälfte des Fensters sehen Sie weitere Metainformationen zum ausgewählten Datensatz (beispielsweise Auflösung, BoundingBox, Datenquelle, bei Vektordaten die Attributtabelle und die COMMAND Geschichte).

Die Werkzeugleiste im *Browser* Reiter stellt folgende Werkzeuge für die ausgewählte LOCATION zur Verfügung:

-  *Ausgewählten Layer dem Kartenfenster hinzufügen*
-  *Gewählte Karte kopieren*
-  *Gewählte Karte umbenennen*
-  *Gewählte Karte löschen*
-  *Setze die Region auf die gewählte Karte*
-  *Neu zeichnen*

Die Knöpfe  *Gewählte Karte umbenennen* und  *Gewählte Karte löschen* stehen nur für die aktuelle MAPSET zur Verfügung. Die anderen Werkzeuge funktionieren auch in den anderen MAPSETs.



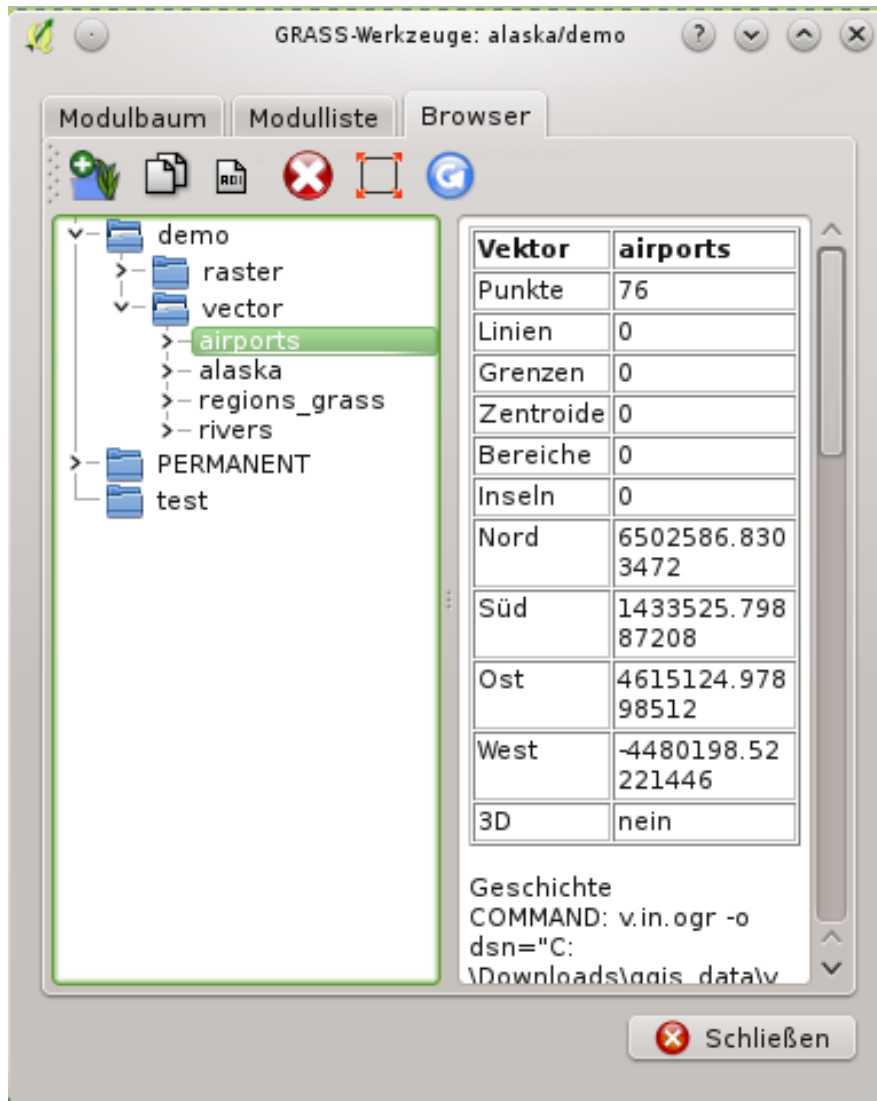


Abbildung 16.15: GRASS LOCATION Browser 

## 16.9.4 Anpassen der Module

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

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!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 Auswerter liest diese Definition und erstellt einen neuen Reiter in der Werkzeugkiste, wenn Sie das Modul auswählen. Ein detaillierte Beschreibung, um neue Module in die GRASS Werkzeugkiste zu integrieren, die Gruppe des Moduls zu ändern etc. finden Sie auf dem QGIS Wiki unter [http://hub.qgis.org/projects/quantum-gis/wiki/Adding\\_New\\_Tools\\_to\\_the\\_GRASS\\_Toolbox](http://hub.qgis.org/projects/quantum-gis/wiki/Adding_New_Tools_to_the_GRASS_Toolbox).

---

## QGIS Verarbeitung Umgebung

---

### 17.1 Einführung

Dieses Kapitel stellt das QGIS Verarbeitung Umgebung, eine leistungsstarke Analyseumgebung für raumbezogene Daten in QGIS, vor. Verarbeitung ist eine Geoverarbeitungsumgebung, die verwendet wird, um eigene und fremde Algorithmen aus QGIS heraus aufrufen zu können, so dass räumliche Analysen produktiv und einfach zu bewerkstelligen sind.

In den folgenden Abschnitten werden wir betrachten wie die grafischen Elemente dieser Umgebung benutzt werden und wie man das Beste aus jedem einzelnen von ihnen herausholen kann.

Es gibt vier Basiselemente in der Umgebungs GUI die dazu verwendet werden Algorithmen zu verschiedenen Zwecken auszuführen. Ob man das eine oder andere Werkzeug auswählt hängt von der Art der Analyse, die durchgeführt werden soll, und den Besonderheiten jedes Nutzers und Projektes ab. Sie können alle (ausgenommen die Batch Processing Schnittstelle, die aus den Werkzeugen heraus aufgerufen wird) aus dem *Verarbeitung* Menü aufgerufen werden. (Sie werden mehr als vier Einträge sehen. Die restlichen werden nicht zum Ausführen von Algorithmen benutzt und werden später in diesem Kapitel erklärt.)

- Die Werkzeuge. Das Hauptelement der GUI wird verwendet um einen einzelnen Algorithmus auszuführen oder einen Batchprozess aufbauend auf diesem Algorithmus laufen zu lassen.
- Die Grafische Modellierung. Viele Algorithmen können mit der Modellierung grafisch kombiniert werden um einen Workflow zu definieren, also einen einzelnen Prozess, der mehrere Unterprozesse beinhaltet.
- Das Protokoll. Alle durchgeführten Aktionen die eins der zuvor erwähnten Elemente verwenden werden in einer History Datei gespeichert und später auf einfache Weise anhand des Protokolls wiederhergestellt werden.
- Die Batch Processing Schnittstelle. Diese Schnittstelle ermöglicht es Ihnen Batchprozesse auszuführen und die Ausführung eines einzelnen Algorithmus auf mehrere Datensätze zu automatisieren.

In den folgenden Abschnitten werden wir auf jedes dieser Elemente im Detail eingehen.

### 17.2 Die Werkzeugkiste

Die *Verarbeitungswerkzeuge* sind das Hauptelement der Verarbeitung GUI und das was Sie eher bei Ihrer täglichen Arbeit benutzen werden. Es zeigt die Liste von verfügbaren Algorithmen in verschiedenen Blöcken gruppiert und ist der Anknüpfungspunkt um Sie auszuführen, entweder als einfachen Prozess oder als Batchprozess mit mehreren Ausführungen des gleichen Algorithmus auf verschiedene Inputdatensätze.

Die Werkzeugkiste enthält alle verfügbaren Algorithmen in fünf vordefinierte Gruppen aufgeteilt. Alle diese Gruppen können unter einem einzelnen Baumeintrag genannt *Geoalgorithms* gefunden werden.

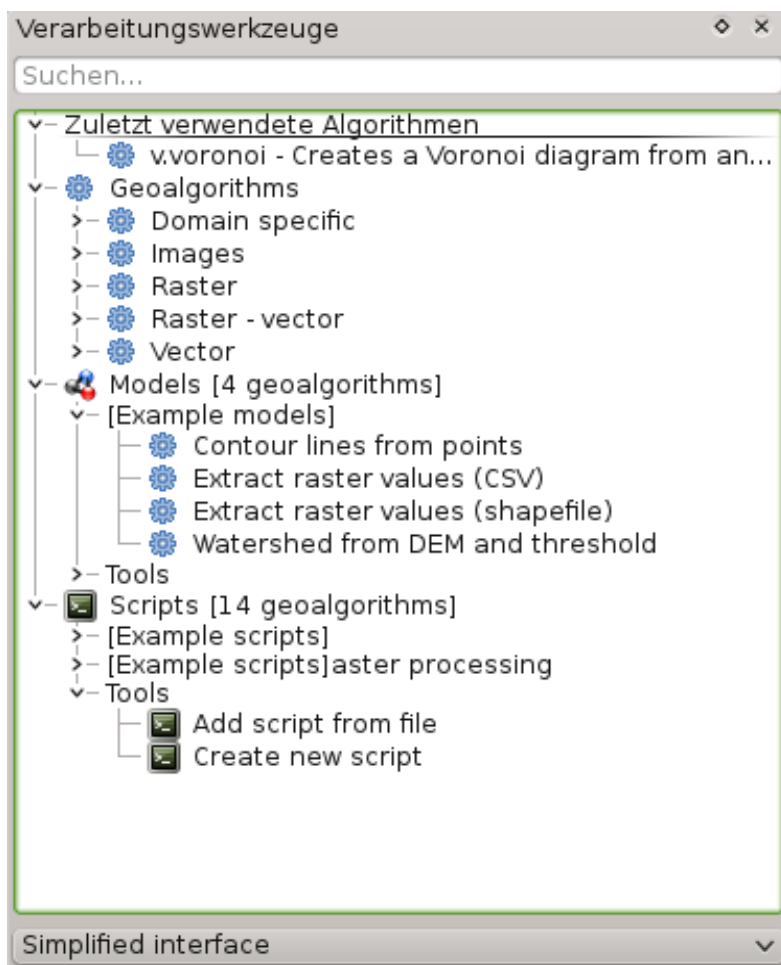


Abbildung 17.1: Verarbeitungswerkzeuge

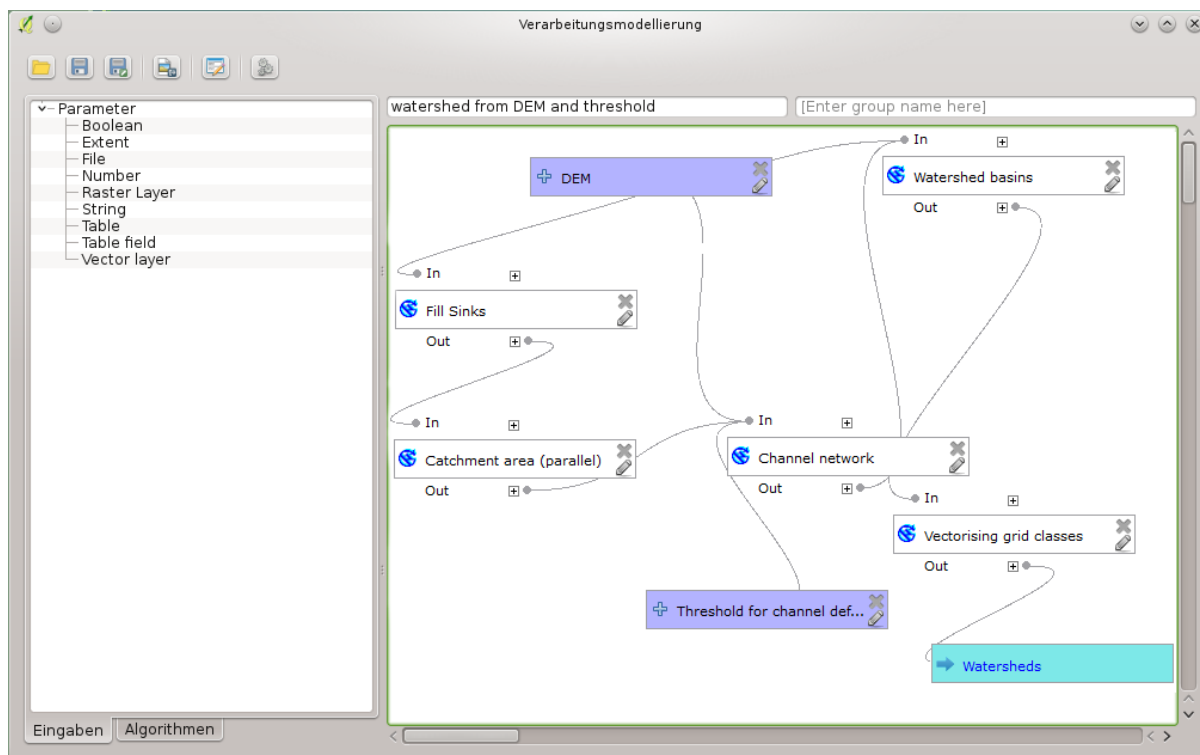


Abbildung 17.2: Verarbeitungsmodellierung

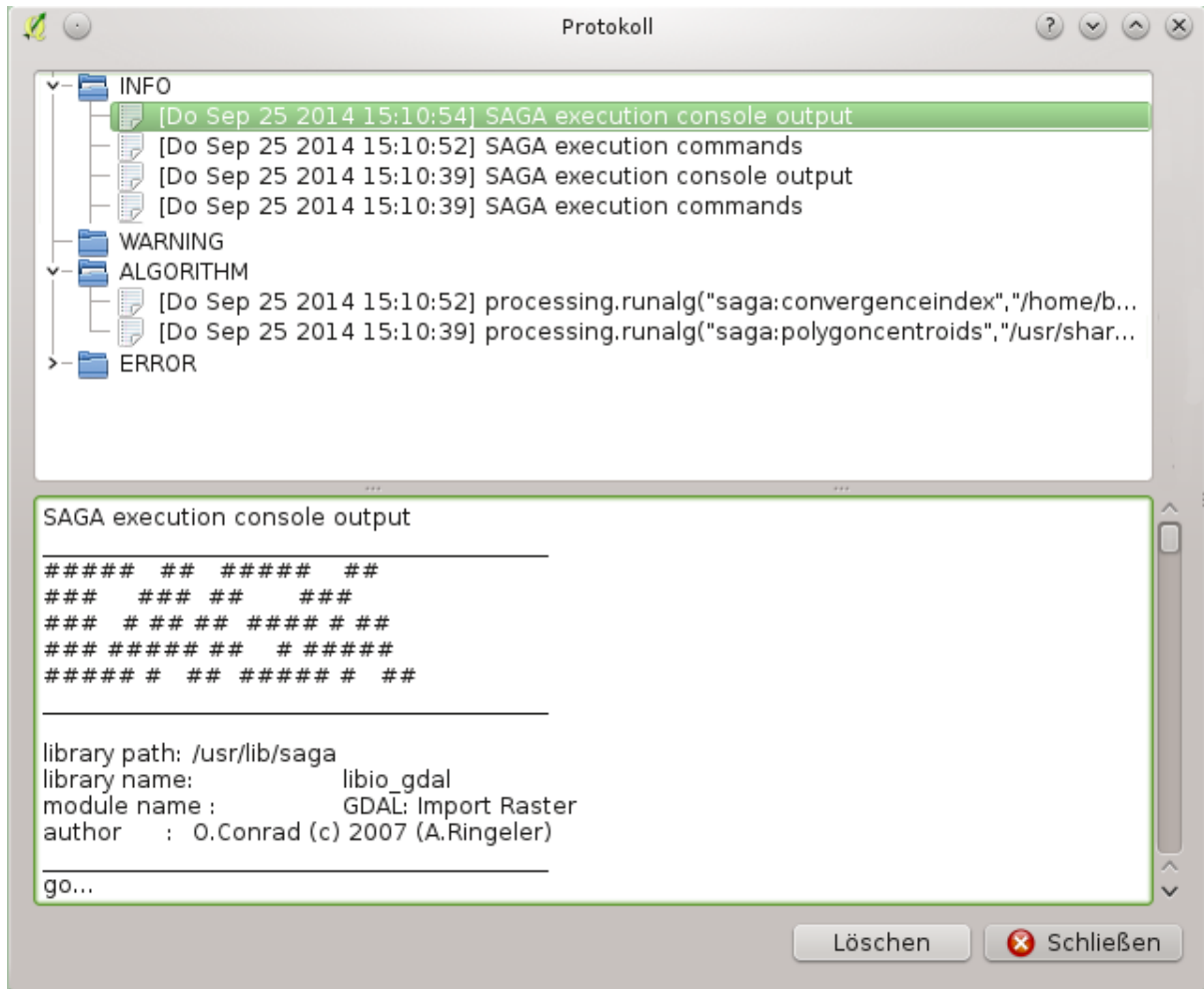


Abbildung 17.3: Verarbeitung Protokoll

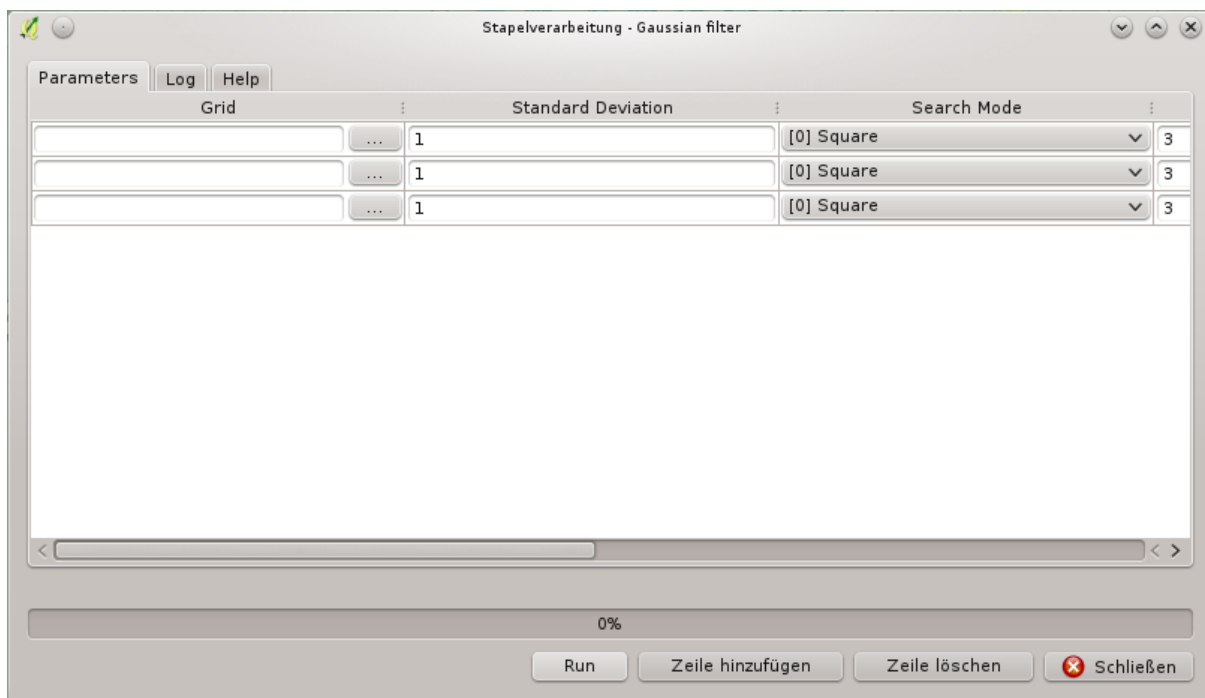


Abbildung 17.4: Batch Processing Schnittstelle

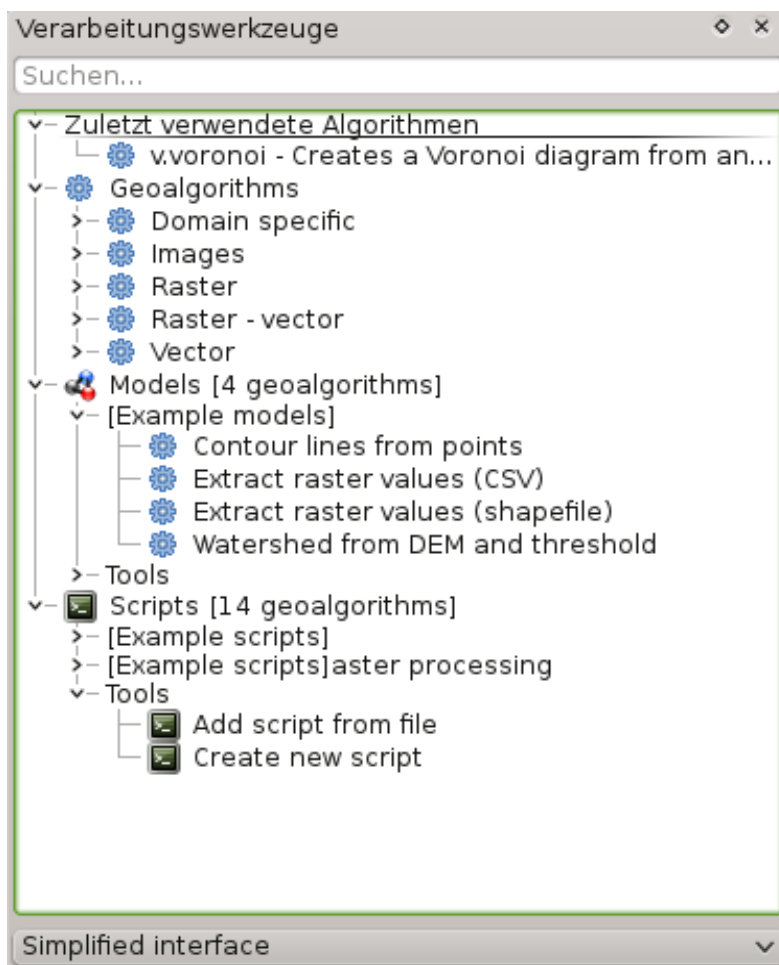



Abbildung 17.5: Verarbeitungswerkzeuge 🌐

Zusätzlich kann man zwei Einträge finden, genannt *Models* und *Scripts*. Diese beinhalten vom Anwender erstellte Algorithmen und sie ermöglichen es Ihnen Ihre eigenen Workflows und Verarbeitungsaufgaben zu definieren. Wir widmen uns ihnen später mit einem ganzen Kapitel.

Im oberen Teil der Werkzeugkiste finden Sie ein Textfeld. Um die Anzahl der Algorithmen die in der Werkzeugkiste gezeigt werden zu verringern und um es Ihnen zu erleichtern den den Sie brauchen zu finden, können Sie jedes beliebige Wort oder Satz in das Textfeld eingeben. Stellen Sie fest, dass während der Eingabe die Anzahl von Algorithmen in der Werkzeugkiste auf genau die, die den Text, den Sie eingegeben haben enthalten, verringert ist.

Im unteren Teil finden Sie ein Fenster das es Ihnen ermöglicht zwischen der vereinfachten Liste von Algorithmen (jene die oben erklärt wurde) und der erweiterten Liste zu wechseln. Wenn Sie zum erweiterten Modus wechseln sieht die Werkzeugkiste so aus:



Abbildung 17.6: Verarbeitungswerkzeuge (advanced interface) 

In der erweiterten Ansicht stellt jede Gruppe einen sogenannten ‘Algorithmen-Provider’ dar, der einen Satz von Algorithmen, die aus der gleichen Quelle kommen, ist, z.B. von einer Drittanwendung mit Geoverarbeitungsfähigkeiten. Einige dieser Gruppen repräsentieren Algorithmen von Drittanwendungen wie SAGA, GRASS oder R, während andere Algorithmen, die direkt als Teil vom Verarbeitungsplugin kodiert werden nicht von zusätzlicher Software abhängig sind.

Diese Ansicht wird den Benutzern empfohlen die ein bestimmtes Wissen über die Anwendungen, die im Hintergrund der Algorithmen laufen haben, da Sie mit Ihren ursprünglichen Namen und Gruppen gezeigt werden.

Auch sind einige zusätzliche Algorithmen nur in der erweiterten Ansicht zugänglich, so etwa u.a. die LIDAR Werkzeuge und Skripte, die auf der statistischen Rechensoftware R basieren.

Vor allem enthält die vereinfachte Ansicht Algorithmen von den folgenden Provider:

- GRASS

- SAGA
- OTB
- QGIS eigene Algorithmen

Wenn Sie QGIS unter Windows benutzen sind diese Algorithmen voll funktional in einer neuen Installation von QGIS, und Sie können eine zusätzliche Installation ausgeführt werden. Auch erfordert das Benutzen keine vorhergehenden Kenntnisse über die externen Anwendungen die sie benutzen, was sie zugänglicher für Erstanwender macht.

Wenn Sie einen Algorithmus, der nicht von einem der oben genannten Provider zur Verfügung gestellt wird, benutzen wollen, wechseln Sie in den Advanced Modus indem Sie die entsprechende Option unten in der Werkzeugkiste auswählen.

Um einen Algorithmus zu starten, klicken Sie einfach doppelt auf den Namen in der Toolbox.

### 17.2.1 Der Algorithmus Dialog

Nachdem Sie auf den Namen des Algorithmus, den Sie ausführen wollen, gedoppelt haben, wird ein Dialog so ähnlich wie der in der Abbildung unten gezeigt (in diesem Fall bezieht sich der Dialog auf den SAGA 'Convergence index' Algorithmus).

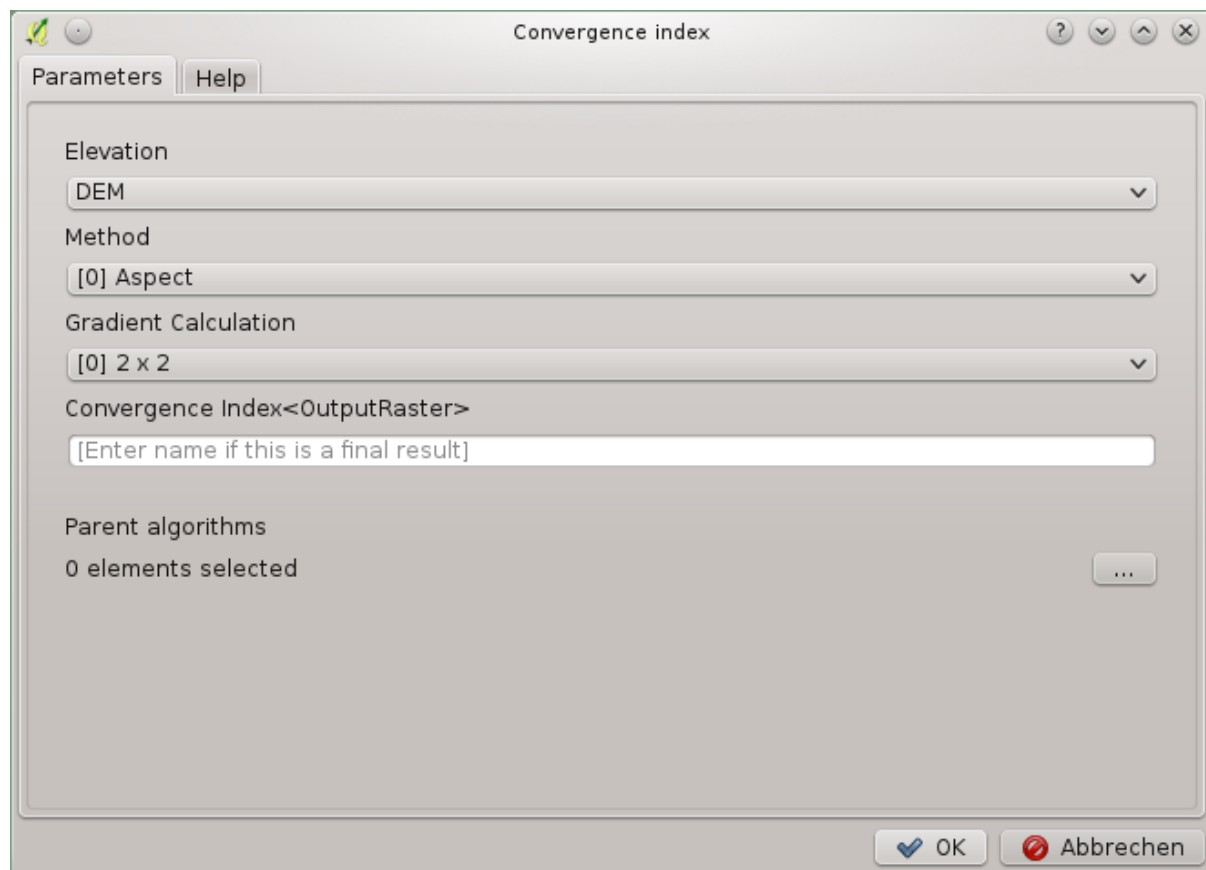


Abbildung 17.7: Der Parameters Dialog 

Dieser Dialog wird verwendet, um die Eingangs-Werte anzugeben, damit der Algorithmus ausgeführt werden kann. Er zeigt eine Tabelle, in der Eingangswerte und Konfigurationsparameter zu setzen sind. Der Inhalt des Dialogs steht in Abhängigkeit vom Algorithmus, der ausgeführt werden soll, und wird automatisch auf Basis der Anforderungen erstellt. Auf der linken Seite wird der Name des Parameters angezeigt. Auf der rechten Seite wird der Parameter gesetzt.



Obwohl die Anzahl und Art der Parameter von den Eigenschaften des Algorithmus abhängt, ist die Struktur für alle ähnlich. Die Parameter die Sie in der Tabelle finden können folgende Typen sein.

- Ein Rasterlayer, der aus der Liste aller in QGIS geladenen Layer (gerade geöffneten) ausgewählt werden kann. Die Auswahl enthält auch einen Knopf auf der rechten Seite mit dem Sie Dateinamen, die die aktuell in QGIS nicht geladenen Layer wiedergeben, auswählen können.
- Ein Vektorlayer aus der Liste aller in QGIS zur Verfügung stehenden Layer. Layer die nicht in QGIS geladen wurden können ebenfalls ausgewählt werden, wie im Fall von Rasterlayern, allerdings nur wenn der Algorithmus kein Tabellenfeld, das von der Attributtabelle des Layers entnommen wird, erfordert. In diesem Fall können nur bereits geladene Layer ausgewählt werden, um auf die Liste der Spalten zugreifen zu können, da Sie geöffnet sein müssen um die Liste der zur Verfügung stehenden Feldnamen abrufen zu können.

Sie werden einen Knopf bei jeder Vektorlayerauswahl sehen, wie in der Abbildung unten gezeigt.

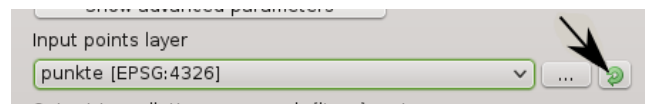


Abbildung 17.8: Vektor Iterator Knopf

Wenn der Algorithmus mehrere davon enthält können Sie auch nur einen von Ihnen umschalten. Wenn der zu einer Vektoreingabe gehörende Knopf umgeschaltet ist wird der Algorithmus iterativ mit jedem seiner Objekte ausgeführt anstatt nur einmal für den ganzen Layer und erstellt so viele Ausgaben wie der Algorithmus ausgeführt wurde. Dies ermöglicht es den Ablauf zu automatisieren wenn alle Objekte in einem Layer separat verarbeitet werden müssen.

- Eine Tabelle, um aus einer Liste von allen in QGIS vorhandenen zu wählen. Nicht-räumliche Tabellen werden in QGIS wie Vektorlayer geladen und werden auch als solche durch das Programm behandelt. Derzeit ist die Liste der verfügbaren Tabellen, die Sie sehen können wenn Sie einen Algorithmus ausführen auf Dateien im DBase- (.dbf) und CSV-Format (.csv) eingeschränkt.
- Eine Option, um mögliche Parameter aus einer Auswahlliste zu wählen.
- Ein numerischer Wert, der in ein Textfeld eingetragen wird. Hier finden Sie einen Knopf an seiner Seite. Mit einem Klick darauf erscheint ein Dialog, über den Sie einen mathematischen Ausdruck eingeben können. Sie können es als handlichen Taschenrechner verwenden. Einige nützliche Variablen, die automatisch in QGIS berechnet sind, können zu Ihrem Ausdruck hinzugefügt werden. Sie können einen Wert aus jeder dieser Variablen, wie z.B. die Zellengröße eines Layers oder den nördlichsten Koordinatenwert eines Layers, verwenden.
- Ein Wertebereich mit Minimum und Maximum, der über zwei Textfelder angegeben wird.
- Ein Text, der in ein Textfeld eingegeben wird.
- Eine Spalte, die aus einer Attributtabelle ausgewählt wird oder eine einzelne Tabelle.
- Ein Koordinatenbezugssystem. Sie können einen EPSG Code direkt ins Textfenster eingeben oder es direkt aus der Koordinatenbezugssystem-Auswahl, die erscheint wenn Sie auf den Knopf auf der rechten Seite klicken, auswählen.
- Vier Zahlen, die eingegeben werden, um die xmin, xmax, ymin, ymax Grenzen festzulegen. Mit einem Klick auf den Knopf auf der rechten Seite erscheint ein Pop-up-Menü mit zwei Optionen: um den Wert aus einem Layer zu übernehmen oder um das aktuelle Ausmaß über das Kartenfensters zu wählen, indem Sie durch Aufziehen eines Rechtecks im Kartenfenster die Ausdehnung definieren.

Wenn Sie die erste Option wählen, sehen Sie einen Dialog wie den nächsten.

Wenn Sie den zweiten wählen, wird das Parameter-Fenster verschwinden, damit Sie durch Klicken und Ziehen im Kartenfenster einen Bereich definieren können. Wenn das Rechteck definiert ist, wird der Dialog mit den Werten wieder auftauchen.

- Eine Liste von Elementen (ob Rasterlayer, Vektorlayer oder Tabellen), aus der Liste von denen, die in QGIS geladen sind. Um eine Auswahl zu treffen, klicken Sie auf die kleine Taste auf der linken Seite der entsprechenden Zeile, um einen Dialog wie den Folgenden zu sehen.

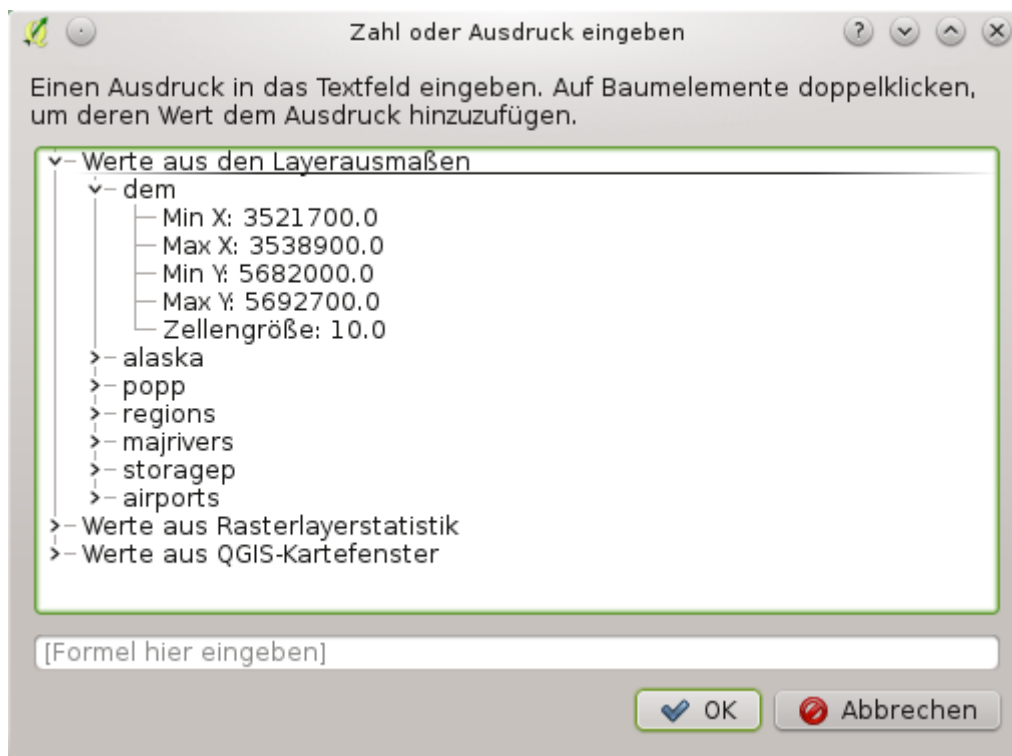


Abbildung 17.9: Zahlauswahl

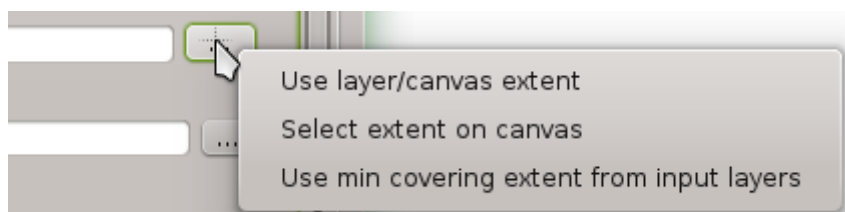


Abbildung 17.10: Ausdehnung Auswahl

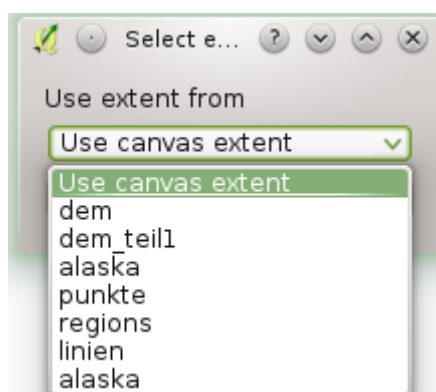


Abbildung 17.11: Ausdehnung Liste

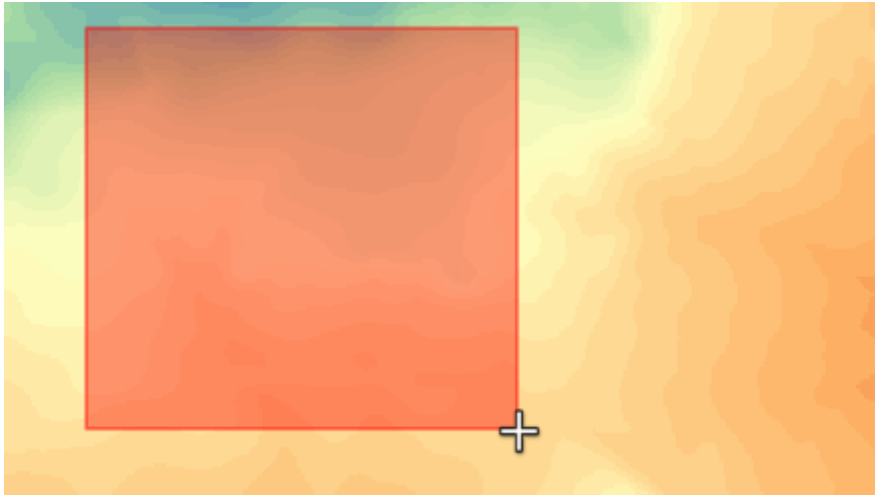


Abbildung 17.12: Ausdehnung aufziehen 🌐

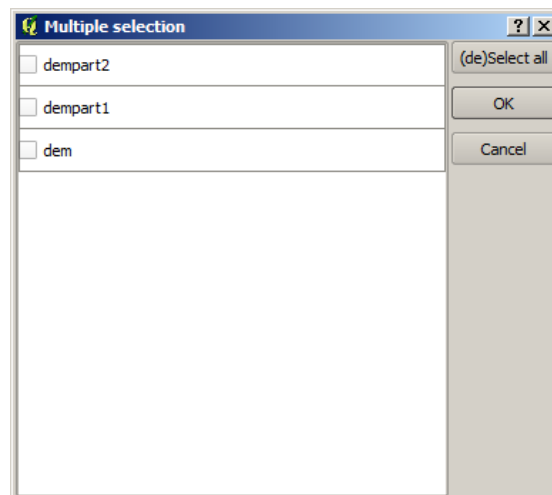


Abbildung 17.13: Mehrfachauswahl 🌐

- Eine kleine Tabelle, um vom Benutzer editiert werden kann. Diese wird verwendet, um Parameter wie Lookup-Tabellen oder Convolution Filter zu definieren.

Klicken Sie auf den Knopf auf der rechten Seite, um die Tabelle zu sehen und zu editieren.

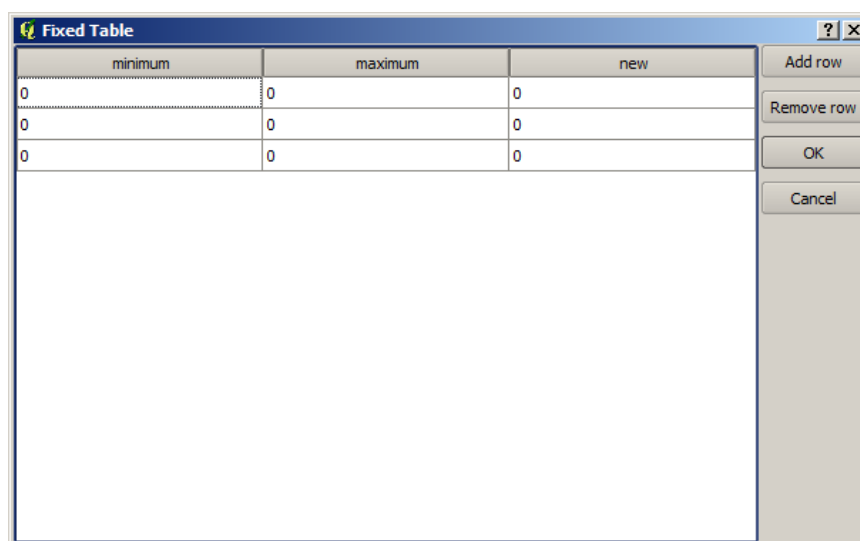


Abbildung 17.14: Feste Tabelle

In Abhängigkeit vom Algorithmus kann die Anzahl der Zeilen verändert werden oder auch nicht, indem Sie auf den Knopf rechts neben dem Fenster klicken.

Sie werden einen **[Help]** Reiter im Parameters Dialog finden. Wenn eine Hilfedatei zur Verfügung steht wird diese angezeigt und stellt Ihnen mehr Informationen über den Algorithmus und genaue Beschreibungen was jeder Parameter macht zur Verfügung. Leider gibt es für die meisten Algorithmen noch keine gute Dokumentation, aber wenn Sie etwas dazu beisteuern wollen, wäre dies eine gute Möglichkeit.

### Bemerkung zum Thema Projektionen

Algorithmen werden durch die Verarbeiten Umgebung ausgeführt - dies trifft auch auf die meisten externen Anwendungen, deren Algorithmen dadurch zur Verfügung gestellt werden, zu. Führen Sie keine Reprojizierung mit Eingabelayern durch und nehmen Sie an dass alle schon in einem gemeinsamen Koordinatensystem sind und bereit für die Analyse sind. Wann immer Sie mehr als einen Layer als Eingabe für einen Algorithmus verwenden, ob Vektorlayer oder Rasterlayer, liegt es an Ihnen sicherzustellen dass Sie alle das gleiche Koordinatensystem besitzen.

Beachten Sie dass aufgrund der Spontan-Reprojektions Fähigkeiten von QGIS trotzdem zwei Layer sich zu überlappen und zu passen scheinen, dies nicht richtig ist wenn ihre Ursprungskordinaten verwendet werden ohne Sie in ein gemeinsames Koordinatensystem zu reprojizieren. Diese Reprojektion sollte manuell durchgeführt werden und die Ergebnisse sollten als Eingabe für den Algorithmus verwendet werden. Beachten Sie auch, dass die Reprojektion mit den in der Bearbeiten Umgebung selber zur Verfügung stehenden Algorithmen durchgeführt werden kann.

Standardmäßig zeigt der Paramters Dialog eine Beschreibung des KBS jedes Layers bei Angabe seines Namens, was es einfach macht Layer auszuwählen, die das gleiche KBS verwenden und diese als Eingabelayer zu benutzen. Wenn Sie diese zusätzliche Information nicht sehen wollen, können Sie diese Funktionalität im Verarbeitungsop-tionen Dialog ausschalten indem Sie die *Show layer CRS definition in selection boxes* Option deaktivieren.

Wenn Sie versuchen einen Algorithmus auszuführen indem Sie als Eingabe zwei oder mehr Layer mit nicht zutreffenden KBSs verwenden, wird ein Warndialog gezeigt.

Sie können den Algorithmus immer noch ausführen, seien Sie sich aber dessen bewusst dass dies in den meisten Fällen zu falschen Ergebnissen führt, so z.B. leere Layer aufgrunddessen dass Eingabelayer nicht überlappen.

## 17.2.2 Von Algorithmen erstellte Datenobjekte

Von Algorithmen erstellte Datenobjekte können jeder der folgenden Typen sein:

- Rasterlayer
- Vektorlayer
- Tabelle
- HTML-Datei (wird für Text und grafische Ausgabe verwendet)

Sie werden alle auf der Festplatte gespeichert und die Parametertabelle enthält ein Textfeld entsprechend jeder Ausgabe, wo Sie eine Ausgabe eingeben können, die für die Speicherung verwendet wird. Die Ausgabe enthält die erforderlichen Informationen, um das resultierende Objekt irgendwo zu speichern. In den meisten Fällen werden Sie es in einer Datei speichern, aber die Architektur von Verarbeitung ermöglicht auch andere Arten der Speicherung. Zum Beispiel kann ein Vektorlayer in einer Datenbank gespeichert werden oder sogar zu einem entfernten Server über eine WFS-T-Schnittstelle hochgeladen werden. Obwohl Lösungen wie diese noch nicht umgesetzt sind, wird die Verarbeitung Umgebung bereit sein, es zu nutzen, wenn es zukünftig möglich sein wird.

Um einen Ausgabekanal auszuwählen klicken Sie einfach auf den Knopf auf der rechten Seite des Textfensters. Dies öffnet einen Datei speichern Dialog, indem Sie den gewünschten Dateipfad auswählen können. Unterstützte Dateierweiterungen werden in der Dateiformatauswahl des Dialogs gezeigt, abhängig von der Art der Ausgabe und des Algorithmus.

Das Format der Ausgabe ist durch die Dateinamenerweiterung definiert. Die unterstützten Formate hängen davon ab was durch den Algorithmus selber unterstützt wird. Um ein Format auszuwählen wählen Sie einfach die entsprechende Dateierweiterung aus (oder fügen Sie sie hinzu, wenn Sie anstelle dessen den Dateipfad direkt eingeben). Wenn die Erweiterung des Dateipfads, die Sie eingegeben haben, keines der unterstützten Formate trifft, wird eine Standarderweiterung (normalerweise `.dbf` für Tabellen, `.tif` für Rasterlayer und `.shp` für Vektorlayer) an den Dateipfad angehängt und das dieser Erweiterung entsprechende Dateiformat wird verwendet um den Layer oder die Tabelle zu speichern.

Wenn Sie keinen Dateinamen eingeben wird das Ergebnis als temporäre Datei im entsprechenden Standarddateiformat gespeichert und wird gelöscht nachdem Sie QGIS verlassen haben (achten Sie darauf dass Sie ihr Projekt speichern und es temporäre Layer enthält).

Sie können einen voreingestellten Ordner für Ausgabedaten Objekte einrichten. Gehen Sie in den Verarbeitungsoptionen Dialog (Sie können ihn aus dem *Verarbeitung* Menü öffnen) und in der *General* Gruppe finden Sie einen Parameter genannt *Output folder*. Dieser Output folder wird als Standardpfad für den Fall, dass Sie nur einen Dateinamen ohne Pfad (z.B. `:file:myfile.shp`) eingeben wenn Sie einen Algorithmus ausführen, verwendet.

Wenn Sie einen Algorithmus ausführen, der einen Vektorlayer im iterativen Modus verwendet, wird der eingegebene Dateipfad als Basispfad für alle erstellten Dateien, die mit Hilfe des Basisnamens benannt werden und an die eine Nummer, die den Index der Iteration darstellt, angehängt wird, verwendet. Die Dateierweiterung (und das Format) wird für alle so erstellten Dateien benutzt.

Neben Rasterlayern und Tabellen erstellen Algorithmen auch Grafiken und Text als HTML Dateien. Diese Ergebnisse werden am Ende der Ausführung des Algorithmus in einem neuen Dialog gezeigt. Dieser Dialog hält die Ergebnisse, die von jedem beliebigen Algorithmus während der aktuellen Sitzung erstellt werden, für die Dauer der aktuellen Sitzung vor und kann jederzeit durch Auswählen von *Verarbeitung* → *Ergebnisanzeige* aus dem QGIS Hauptmenü angezeigt werden.

Einige externe Anwendungen können Dateien (ohne besondere Erweiterungseinschränkungen) als Ausgabe haben, diese gehören dann aber nicht zu einer der oben genannten Kategorien. Diese Ausgabedateien werden nicht von QGIS bearbeitet (geöffnet oder in das aktuelle QGIS Projekt eingefügt), das Sie meistens den Dateiformaten oder -elementen entsprechen, die nicht von QGIS unterstützt werden. Dies ist z.B. der Fall mit LAS Dateien, die für LiDAR Daten verwendet werden. Die Dateien werden erstellt, aber Sie werden nichts neues in Ihrer QGIS Arbeitssitzung sehen.

Für alle anderen Arten von Ausgaben finden Sie ein Kontrollkästchen, mit dem sie festlegen können, ob die Datei in QGIS geladen werden soll oder nicht, wenn sie durch den Algorithmus erzeugt wurde. Standardmäßig werden alle Dateien angezeigt.

Optionale Ausgaben werden nicht unterstützt, alle Ausgaben werden erstellt, aber Sie können über das entsprechende Kontrollkästchen definieren, wenn Sie an einer bestimmten Ausgabe nicht interessiert sind, was es im Wesentlichen zu einer Optionale Ausgabe macht (Der Layer wird zwar erstellt, aber nur temporär im Hintergrund, wenn Sie das Textfeld leer lassen, und wieder gelöscht, sobald Sie QGIS verlassen).

### 17.2.3 Die Verarbeiten Umgebung konfigurieren

Wie bereits erwähnt wurde, gibt das Konfigurationsmenü Zugang zu einem neuen Dialog, in dem Sie festlegen, wie Algorithmen funktionieren. Die Konfigurationsparameter sind in separate Blöcke strukturiert und können auf der linken Seite des Dialogs ausgewählt werden.

Zusammen mit dem oben genannten *Ausgabeordner* Eintrag, enthält der *Allgemein* Block Parameter zur Einstellung der Standard-Darstellung für Ausgabe Layer (d.h. Layer die durch Benutzen der Algorithmen aus einer der Umgebungs GUI Komponenten erstellt werden). Erstellen Sie einfach den gewünschten Stil mit QGIS, speichern Sie ihn in eine Datei, und geben Sie den Pfad zu dieser Datei in den Einstellungen an, so dass die Algorithmen sie verwenden kann. Immer, wenn ein Layer von SEXTANTE in QGIS geladen wird, wird dieser Stil verwendet.

Die Darstellungsstile können individuell für jeden Algorithmus und jeden seiner Ausgaben konfiguriert werden. Machen Sie einfach einen Rechtsklick auf den Namen des Algorithmus in der Werkzeugkiste und wählen Sie *Darstellungsstile zur Ausgabe bereiten*. Sie werden einen Dialog wie unten gezeigt sehen.

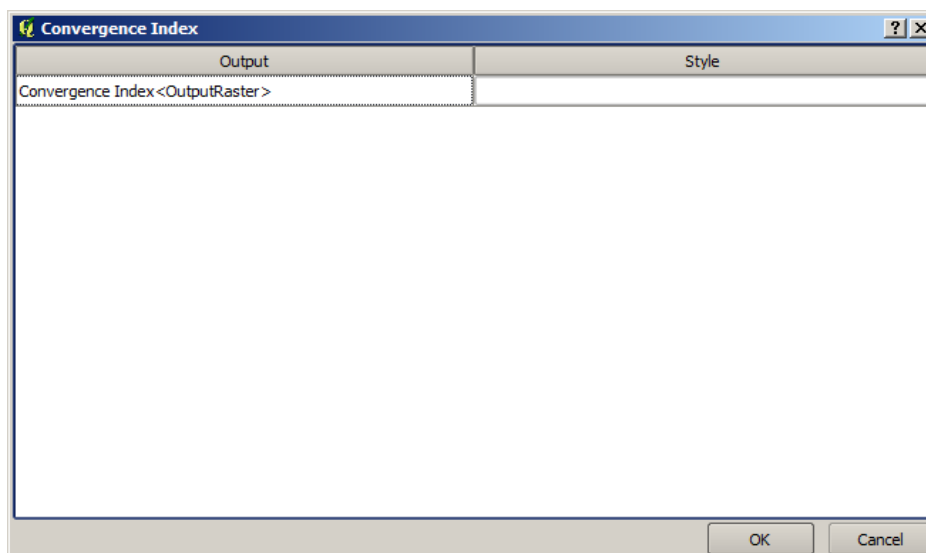


Abbildung 17.15: Darstellungsstile 

Wählen Sie die Stildatei (.qml) für jede Ausgabedatei und klicken Sie dann **[OK]**.

Andere Konfigurationsparameter in der *General* Gruppe sind unten aufgeführt:

- *Use filename as layer name.* Der Name jedes resultierenden Layers, der von einem Algorithmus erstellt wurde, wird durch den Algorithmus selber definiert. In einigen Fällen kann ein fester Name verwendet werden, was heißt dass der selbe Ausgabename benutzt wird, egal welcher Inputlayer verwendet wird. In anderen Fällen, kann der Name von dem Eingabelayer oder einigen Parametern, die verwendet werden um den Algorithmus auszuführen, abhängen. Wenn dieses Kontrollkästchen aktiviert ist wird anstelle dessen der Name dem Ausgabedateinamen entnommen. Beachten Sie dass wenn die Ausgabe in eine temporäre Datei gespeichert wird, der Name dieser temporären Datei für gewöhnlich lang und bedeutungslos ist und damit Kollisionen mit bereits existierenden Dateinamen vermieden werden sollen.
- *Use only selected features.* Wenn diese Option ausgewählt ist, werden nur ausgewählte Objekte verwendet wenn ein Vektorlayer als Eingabe für einen Algorithmus benutzt wird. Wenn der Layer keine ausgewählten Objekte hat, werden alle Objekte benutzt.

- *Pre-execution script* und *Post-execution script*. Diese Parameter beziehen sich auf Scripts, die anhand der Verarbeitung Scripting Funktionalität geschrieben worden sind und werden in dem Abschnitt, der Scripting und Konsole abgedeckt erklärt.

Abgesehen vom Bereich *General* im Menü Optionen und Konfiguration, finden Sie auch einen für jeden Algorithmus-Provider. Sie enthalten ein *aktivieren* Element, das Sie verwenden können, um festzulegen, ob Algorithmen dieses Providers in der Toolbox erscheinen oder nicht. Auch haben einige Provider ihre eigene Algorithmus-Konfiguration, die wir später noch genauer erklären wenn bestimmte Algorithmus Provider abgedeckt werden.

## 17.3 Die Grafische Modellierung

Die *grafische Modellierung* ermöglicht es, komplexe Modelle mit einer einfachen und leicht zu bedienenden Benutzeroberfläche zu erstellen. Beim Arbeiten mit einem GIS finden die meisten Analyse-Operationen nicht isoliert, sondern als Teil einer Kette von Operationen statt. Der grafische Modeller kann eine Kette von Prozessen in einen einzigen Prozess einpacken. So ist es einfacher und bequemer, als einzelne Prozesse nacheinander auszuführen. Egal, wie viele Schritte und verschiedene Algorithmen es sind, sie können als Modell in einem einzigen Algorithmus ausgeführt werden. Das spart Zeit und Mühe, besonders für größere Modelle.

Die Modellierung kann aus dem Verarbeiten Menü heraus geöffnet werden.

Der Modeller hat einen Arbeitsdialog, wo die Struktur des Modells und der Workflow dargestellt wird. Auf der linken Seite des Fensters kann ein Bereich mit zwei Reitern verwendet werden, um neue Elemente in das Modell zu integrieren.

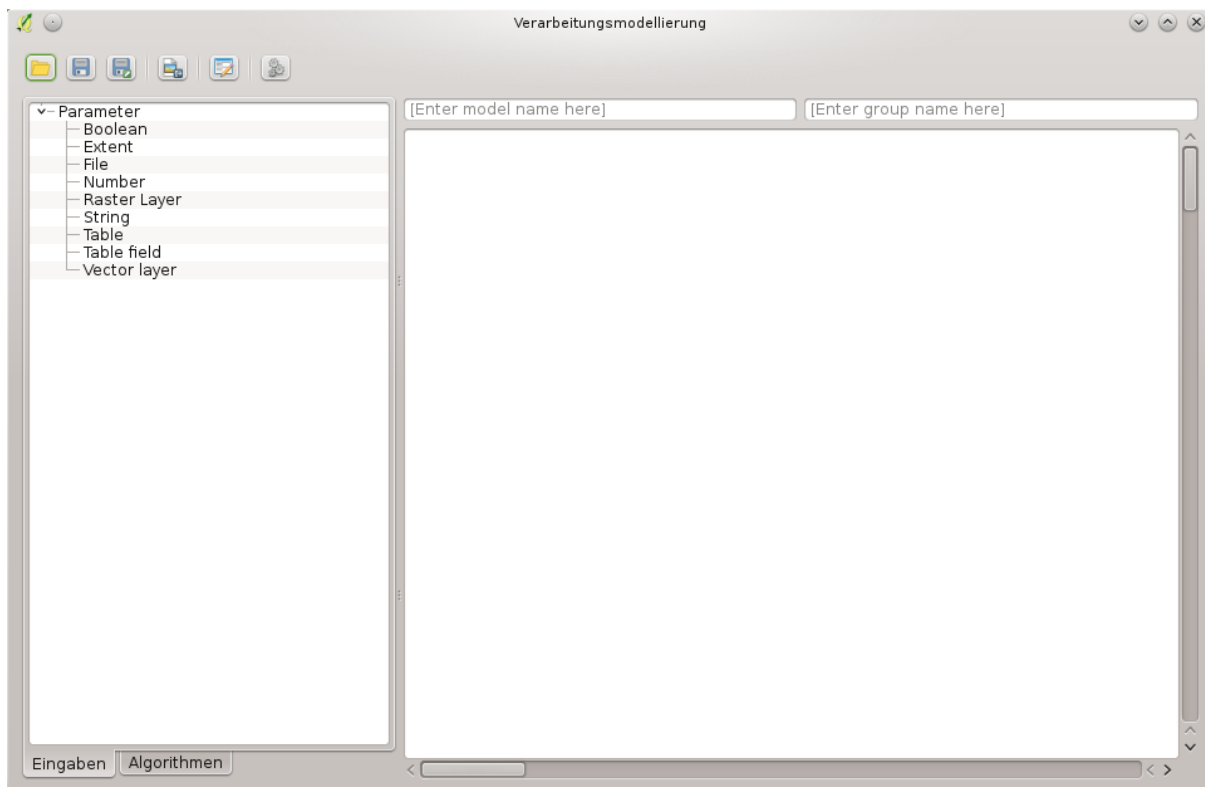



Abbildung 17.16: Modellierung 

Das Erstellen eines Modells geht in zwei Schritten:

1. *Definition von erforderlichen Eingaben*. Diese Eingaben werden dem Parameterfenster hinzugefügt, so dass der Anwender Ihre Werte einstellen kann wenn er das Modell ausführt. Das Modell selber ist ein Algo-

rithmus, also wird das Parameterfenster automatisch erstellt, so wie es mit allen Algorithmen, die in der Verarbeitung Umgebung zur Verfügung stehen, passiert.

2. *Definition des Workflows.* Während Sie die Eingabedaten des Modells verwenden, wird der Workflow durch Hinzufügen von Algorithmen und Auswählen, wie diese Eingaben und Ausgaben, die von anderen Algorithmen erzeugt werden, benutzt werden, definiert.

### 17.3.1 Definition von Eingaben

Der Erste Schritt ein Modell zu erstellen ist die Eingaben, die es benötigt zu erstellen. Die folgenden Elemente finden Sie im *Eingaben* Reiter auf der linken Seite des Modellierungsfensters:

- Rasterlayer
- Vektorlayer
- Text
- Tabellenspalte
- Tabelle
- Extent
- Number
- Boolean
- File

Wenn man einen Doppelklick auf eines dieser Elemente macht wird ein Dialog zum Definieren seiner Eigenschaften gezeigt. Abhängig vom Parameter selber kann der Dialog nur ein grundlegendes Element (die Beschreibung, was der Benutzer sehen wird wenn er das Modell ausführt) oder mehrere beinhalten. Beispielsweise wenn man einen numerischen Wert hinzufügt, wie in der nächsten Abbildung gezeigt, müssen Sie neben der Beschreibung des Parameters einen Standardwert und eine Wertespanne von gültigen Werten einstellen.

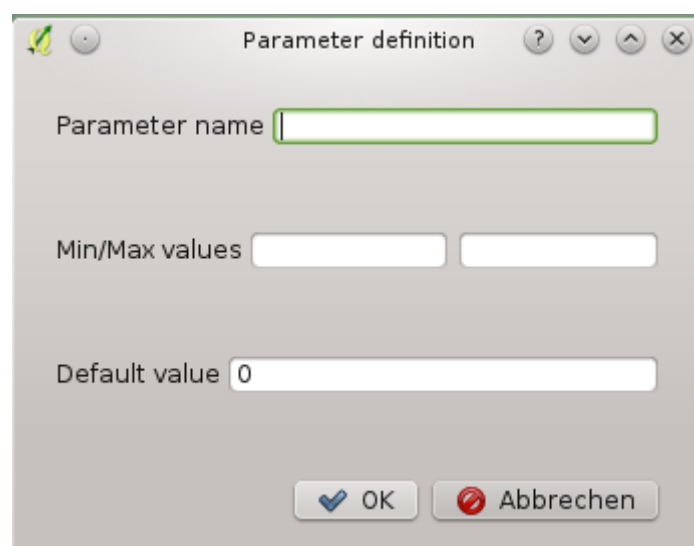


Abbildung 17.17: Modelparameter 

Für jede Eingabe wird ein neues Element im Modeller-Fenster erstellt.

You can also add inputs by dragging the input type from the list and dropping it in the modeler canvas, in the position where you want to place it.





Abbildung 17.18: Modelparameter

### 17.3.2 Definition des Workflows

Wenn die Eingaben definiert sind, ist es Zeit die Algorithmen festzulegen. Die Algorithmen können im Reiter *Algorithmen* ausgewählt werden. Die Gruppierung entspricht der Toolbox.

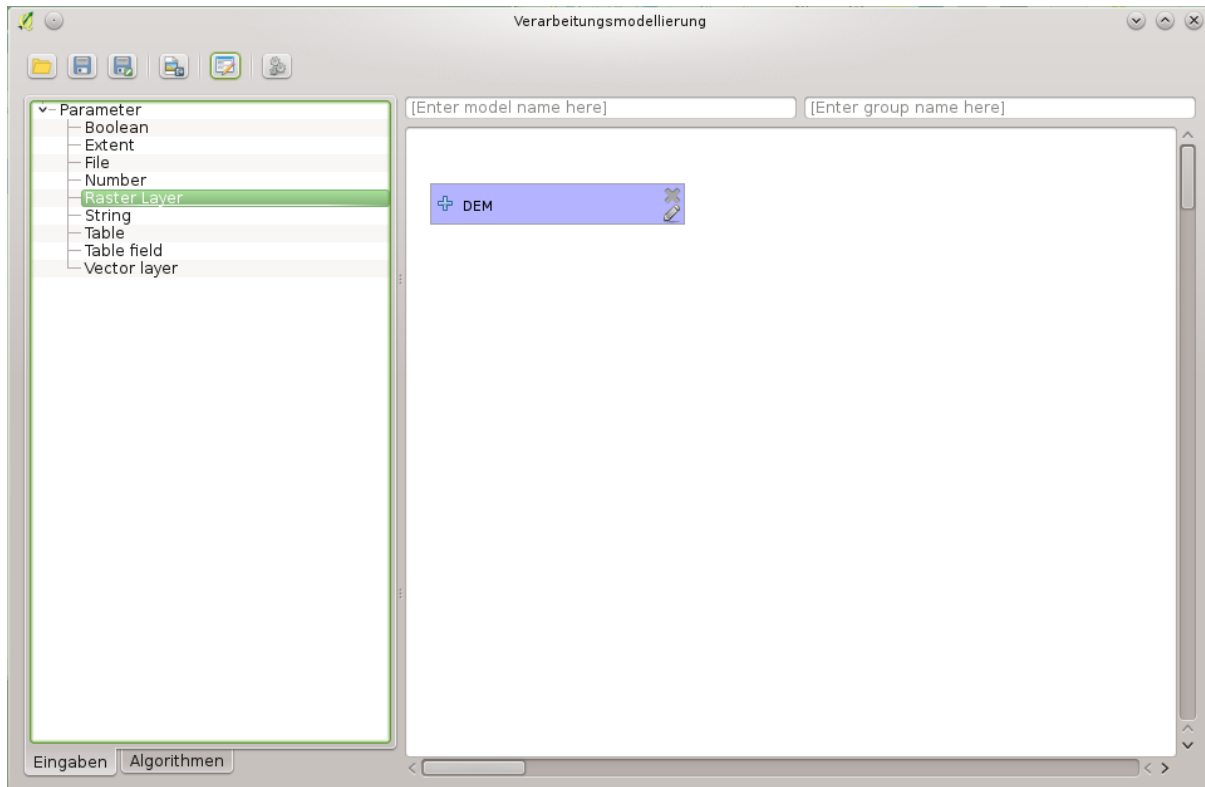


Abbildung 17.19: Modelparameter

Auch hier hat die Erscheinung der Werkzeugkiste zwei Modi: *simplified* und *advanced*. Jedoch gibt es kein Element zum Wechseln zwischen den Ansichten in der Modellierung, so dass Sie dies in der Werkzeugkiste machen müssen. Der Modus, der in der Werkzeugkiste ausgewählt ist, ist der, der für die Liste von Algorithmen in der Modellierung verwendet wird.

To add an algorithm to a model, double-click on its name or drag and drop it, just like it was done when adding inputs. An execution dialog will appear, with a content similar to the one found in the execution panel that is shown when executing the algorithm from the toolbox. The one shown next corresponds to the SAGA 'Convergence index' algorithm, the same example we saw in the section dedicated to the toolbox.

Wie Sie sehen können bestehen einige Unterschiede. Anstelle des Dateiausgabefensters, das für die Einstellung des Dateipfades für Ausgabebayer und -tabellen verwendet wurde, wird hier eine einfaches Textfenster verwendet. Wenn der vom Algorithmus erstellte Layer nur ein vorläufiges Ergebnis ist, das als Eingabe für einen anderen Algorithmus verwendet wird und nicht als endgültiges Ergebnis vorgehalten werden soll, bearbeiten Sie dieses Textfenster nicht. Wenn Sie etwas eingeben heißt das, dass das Ergebnis endgültig ist und dass der Text den Sie vergeben die Beschreibung für die Ausgabe, die die Ausgabe ist, die der Benutzer sieht wenn das Modell ausgeführt wird, ist.

Das Auswählen des Wertes für jeden Parameter ist ebenfalls etwas schwierig, da es wichtige Unterschiede zwischen dem Kontext der Modellierung und der der Werkzeugkiste gibt. Wollen wir sehen wie man die Werte für

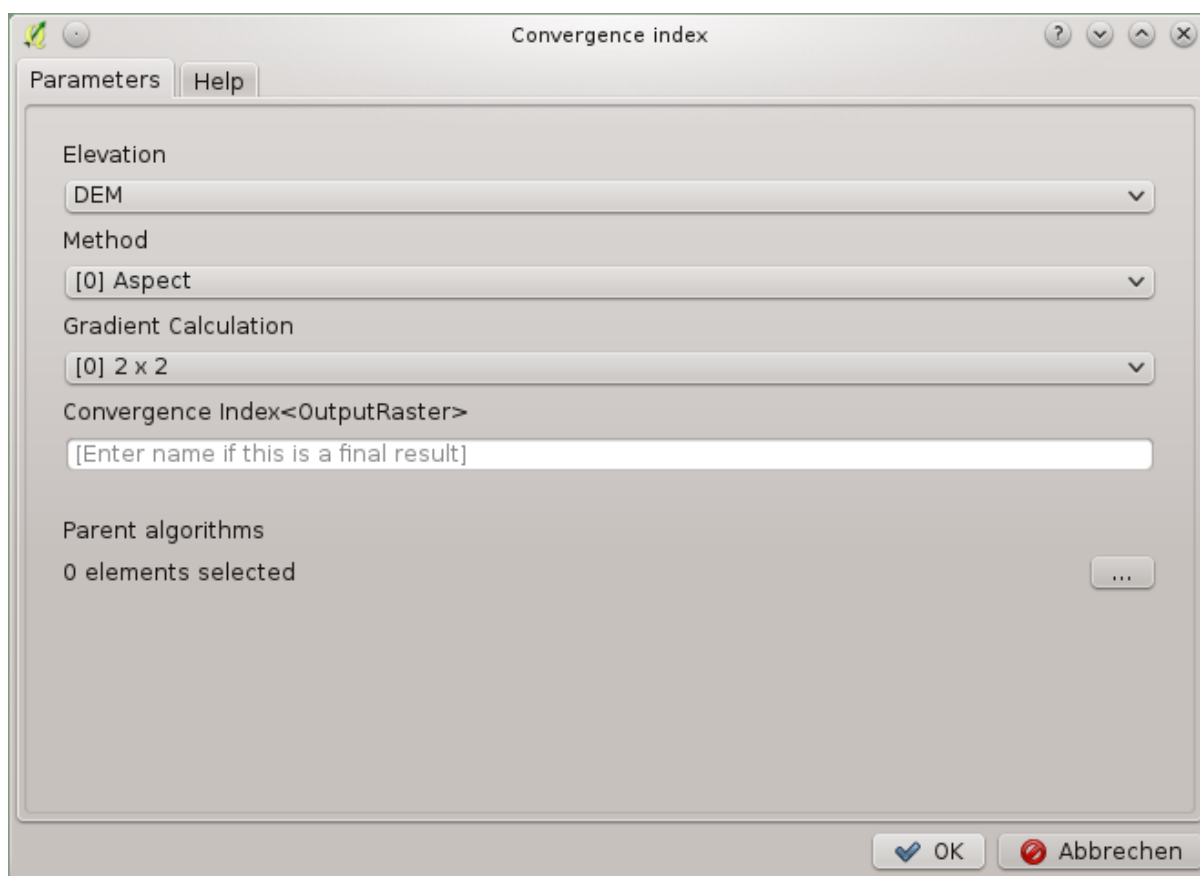



Abbildung 17.20: Modelparameter 

jeden Typ von Parameter vorstellen.

- Layer (Raster und Vektor) und Tabellen. Diese werden aus einer Liste ausgewählt, aber in diesem Fall sind die möglichen Werte nicht die Layer oder Tabellen die gerade in QGIS geladen sind, sondern die Liste von Modelleingaben des entsprechenden Typs oder andere Layer und Tabellen, die von Algorithmen, die schon dem Modell hinzugefügt wurden, erstellt wurden.
- Numerische Werte. Literale Werte können direkt im Textfenster eingeführt werden. Dieses Textfeld ist ebenfalls eine Liste, die verwendet werden kann um jede der numerischen Werteingaben des Modells auszuwählen. In diesem Fall wird der Parameter den Wert, der vom Benutzer beim Ausführen des Modells eingeführt wird, annehmen.
- String. Wie im Fall von numerischen Werten können literale Strings eingegeben werden oder ein Eingabestring kann ausgewählt werden.
- Table field. Die Felder der Elterntabelle oder -layers können nicht zur Erstellungszeit ermittelt werden, da sie von der Auswahl des Anwenders jedes mal wenn das Modell ausgeführt wird abhängen. Um den Wert für diesen Parameter einzustellen, geben Sie den Namen des Feldes direkt in das Textfenster ein oder verwenden Sie die Liste um eine Tabellenfeldeingabe, die schon dem Modell hinzugefügt wurde, auszuwählen. Die Gültigkeit des ausgewählten Feldes wird zur Laufzeit überprüft.

Sie werden in jedem Fall einen zusätzlichen Parameter genannt *Parent algorithms*, der nicht zugänglich ist wenn Sie den Algorithmus aus der Werkzeugkiste aufrufen, finden. Dieser Parameter ermöglicht es Ihnen die Reihenfolge in welcher die Algorithmen ausgeführt werden, indem ein Algorithmus ausdrücklich als Eltern des aktuellen definiert werden, festzulegen. Dies bewirkt, dass der Elteralgorithmus vor dem aktuellen ausgeführt wird.

Wenn Sie die Ausgabe eines vorherigen Algorithmus als Eingabe für Ihren Algorithmus benutzen, stellt dies implizit den vorherigen Algorithmus als Eltern des gerade verwendeten ein (und platziert den entsprechenden Pfeil in der Modellierungsoberfläche). In einigen Fällen jedoch kann ein Algorithmus von einem anderen abhängen selbst wenn er keine Ausgabe von diesem verwendet (beispielsweise ein Algorithmus der einen SQL-Satz in einer PostGIS Datenbank ausführt und ein weiterer der einen Layer in die gleiche Datenbank importiert). Wählen Sie in diesem Fall einfach den vorherigen Algorithmus im *Parent algorithms* Parameter aus und die zwei Schritte werden in der richtigen Reihenfolge ausgeführt.

Nachdem allen Parametern die gültigen Werte zugewiesen worden sind klicken Sie auf **[OK]** und der Algorithmus wird der Oberfläche hinzugefügt. Er wird mit allen anderen Elementen in der Oberfläche verbunden, egal ob Algorithmen oder Eingaben, die Objekte darstellen, die als Eingabe für diesen Algorithmus verwendet werden.

Elements can be dragged to a different position within the canvas, to change the way the module structure is displayed and make it more clear and intuitive. Links between elements are updated automatically. You can zoom in and out by using the mouse wheel.

Sie können Ihren Algorithmus zu jederzeit durch Klicken auf den **[Run]** Knopf ausführen. Um jedoch den Algorithmus aus der Werkzeugkiste zu verwenden muss er gespeichert werden und der Modellierungsdialo geschlossen werden, damit die Werkzeugkiste ihre Inhalte auffrischen kann.

### 17.3.3 Speichern und laden von Modellen

Verwenden Sie den **[Speichern]** Knopf um das aktuelle Modell zu speichern und den **[Modell öffnen]** Knopf um ein vorher gespeichertes Modell zu öffnen. Modelle werden mit der `.model` Erweiterung gespeichert. Wenn das Modell vorher aus dem Verarbeitungsmodellierung Fenster heraus gespeichert wurde, werden Sie nicht nach einem Dateinamen gefragt. Da es bereits eine Datei, die zu diesem Modell zugeordnet wurde, gibt, wird die gleiche Datei für jedes nachfolgende Speichern verwendet.

Vor dem Speichern eines Modells müssen Sie einen Namen und eine Gruppe angeben. Nutzen Sie dazu die Textfelder im oberen Teil des Fensters.

Modelle, die im `models` Ordner (der Standardordner wenn Sie aufgefordert werden einen Dateinamen zum Speichern des Modell einzugeben) gespeichert sind, erscheinen in der Werkzeugkiste im entsprechenden Baum. Wenn die Verarbeitungswerkzeuge aufgerufen werden durchsucht sie den `models` Ordner nach Dateien mit der `.model` Erweiterung und lädt die Modelle, die enthalten sind. Da ein Modell ein Algorithmus ist, kann es einfach wie jeder andere Algorithmus zur Werkzeugkiste hinzugefügt werden.

Die Verarbeitungsmodellierung kann im Verarbeitungskonfigurationsdialog in der *Models* Gruppe eingestellt werden.

Modelle, die aus dem *Models* Ordner geladen werden, erscheinen nicht nur in der Werkzeugkiste, sondern auch im Algorithmusbaum im *Algorithmen* Reiter des Modellierungsfensters.

In einigen Fällen kann ein Modell vielleicht nicht geladen werden da nicht alle im Workflow enthaltenen Algorithmen zur Verfügung stehen. Wenn Sie einen gegebenen Algorithmus als Teil Ihres Modells benutzt haben, sollte er zu Verfügung stehen (er sollte also in der Werkzeugkiste erscheinen) um das Modell laden zu können. Das Deaktivieren eines Algorithmusproviders in den Verarbeitungsoptionen rendert alle die Algorithmen des Providers, der nicht von der Modellierung benutzt werden kann, was Probleme beim Laden von Modellen verursachen kann. Denken Sie daran wenn Sie Probleme beim Laden oder Ausführen von Modellen haben.

### 17.3.4 Ein Modell editieren

Sie können das Modell, das Sie gerade erstellen editieren, den Workflow neu definieren sowie die Beziehungen zwischen den Algorithmen und den Eingangsdaten, die das Modell ausmachen ändern.

Wenn Sie mit der rechten Maustaste auf einen Algorithmus im Dialog-Fenster klicken, wird ein Kontextmenü geöffnet. wie in der nächsten Abbildung zu sehen:

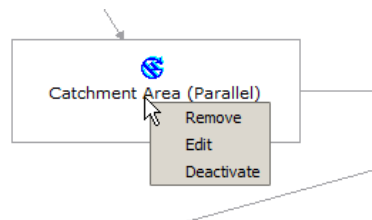



Abbildung 17.21: Rechtsklick auf den Modeler 

Wird die *Remove* Option gewählt, bewirkt dies, dass der ausgewählte Algorithmus entfernt wird. Ein Algorithmus kann nur entfernt werden wenn keine weiteren Algorithmen von ihm abhängen. Das heißt, wenn keine Ausgabe des Algorithmus in einem anderen als Eingabe verwendet wird. Wenn Sie versuchen einen Algorithmus, von dem andere abhängen, zu entfernen, wird eine Warnmeldung wie die, die Sie unten sehen können, gezeigt:

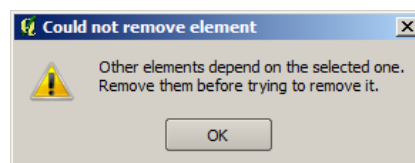


Abbildung 17.22: Kann Algorithmus nicht entfernen 

Auswählen der *Bearbeiten* Option oder einfaches Doppelklicken auf das Symbol des Algorithmus zeigt den Parameter-Dialog des Algorithmus, um die Ein- und Ausgabe Parameter-Werte zu ändern. Nicht alle Eingabe-Elemente des Modells werden in diesem Fall als verfügbare Eingänge angezeigt. Layer oder Werte, die in einem fortgeschrittenen Schritt im Workflow durch das Modell erzeugt wurden, stehen nicht zur Verfügung, wenn sie zirkuläre Abhängigkeiten verursachen.

Wählen Sie die neuen Werte und klicken dann auf den **[OK]** Knopf wie gewohnt. Die Verbindungen zwischen den Elementen des Modells ändern sich entsprechend im Modeller-Fenster.

### 17.3.5 Editieren der Modell-Hilfe Datei und der Metainformationen

Sie können Ihre Modelle aus der Modellierung heraus dokumentieren. Klicken Sie einfach auf den **[Modellhilfe bearbeiten]** Knopf und ein Dialog, wie er als nächstes gezeigt wird, erscheint.

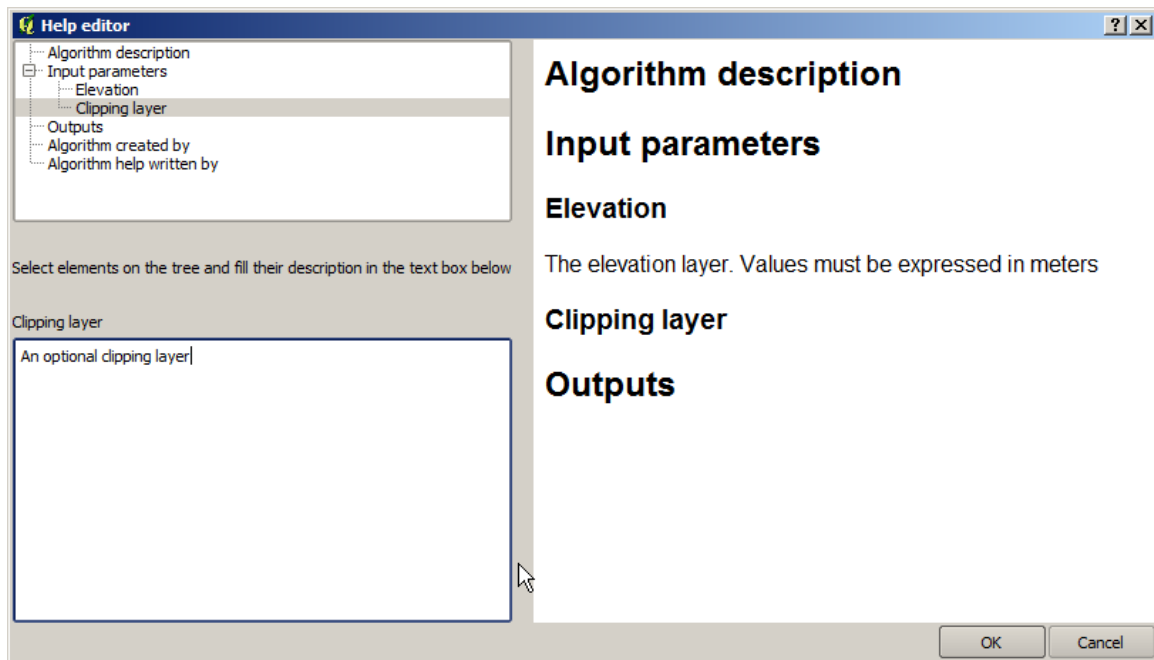



Abbildung 17.23: Hilfeeditor 

Auf der rechten Seite finden Sie eine einfache HTML-Seite, die anhand der Eingabeparameter und Ausgaben des Algorithmus erstellt wurde, zusammen mit einigen zusätzlichen Einträgen wie eine allgemeine Beschreibung des Modells und seines Autors. Wenn Sie den Hilfeeditor das erste Mal öffnen, sind alle diese Beschreibungen leer, Sie können Sie aber mit Hilfe der Elemente auf der linken Seite des Dialogs bearbeiten. Wählen Sie ein Element im oberen Teil aus und schreiben Sie seine Beschreibung in das Textfeld unten.

Die Modell Hilfe wird in einer Datei im gleichen Ordner wie das Modell selbst gespeichert. Sie müssen nichts tun, um es zu speichern, da es automatisch geschieht.

### 17.3.6 Zu den verfügbaren Algorithmen

Sie haben vielleicht bemerkt, dass einige Algorithmen, die aus der Toolbox ausgeführt werden können, nicht in der Liste der verfügbaren Algorithmen erscheint, wenn Sie ein Modell entwerfen. Um in einem Modell enthalten zu sein, muss der Algorithmus eine korrekte Semantik haben, um im Workflow verknüpft zu werden. Wenn ein Algorithmus keine solche definierte Semantik besitzt (zum Beispiel, wenn die Anzahl der Ausgangskanäle oder Layer nicht im Voraus bekannt ist), dann ist es nicht möglich, diesen in einem Modell zu verwenden, und somit erscheint er auch nicht in der Liste.

Zusätzlich werden Sie einige Algorithmen in der Modellierung sehen, die nicht in der Werkzeugkiste gefunden werden können. Diese Algorithmen sollen ausschließlich als Teil eines Modells benutzt werden und sie in einem anderen Kontext nicht von Interesse. Der 'Calculator' Algorithmus ist ein Beispiel dafür. Er ist nur ein einfacher arithmetischer Rechner den Sie benutzen können um numerische Werte zu verändern (sie können durch den Benutzer eingegeben werden oder durch einen anderen Algorithmus erstellt werden). Dieses Werkzeug ist wirklich nützlich innerhalb eines Modells, aber außerhalb dieses Kontexts machen es nicht allzuviel Sinn.

## 17.4 Die Stapelprozeß Schnittstelle

### 17.4.1 Einführung

Alle Algorithmen (inclusive Modelle) können als Stapelprozeß ausgeführt werden. Ds heißt, sie können nicht nur anhand eines einzigen Satz von Eingaben, sondern mehreren von ihnen ausgeführt werden, wobei der Prozess so

oft wie benötigt ausgeführt werden kann. Dies ist nützlich wenn große Datenmengen prozessiert werden, da es nicht nötig ist den Algorithmus viele Male von der Werkzeugkiste aus zu starten.

Um einen Algorithmus als Batch-Prozess ausführen, klicken Sie mit der rechten Maustaste auf seinen Namen in der Toolbox und wählen Sie die Option *Ausführen als Batch-Prozess*.

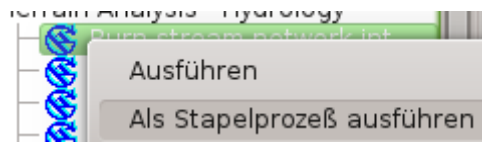


Abbildung 17.24: Stapelprozess anhand eines Rechtsklicks

### 17.4.2 Die Parameter-Tabelle

Ausführen eines Batch-Prozesses ist vergleichbar mit der Durchführung der einmaligen Ausführung eines Algorithmus. Parameter-Werte müssen definiert werden, aber in diesem Fall müssen wir nicht nur einen einzelnen Wert für jeden Parameter angeben, sondern eine Reihe von ihnen, einen für jedes Mal, wenn der Algorithmus ausgeführt werden soll. Werte werden mit Hilfe einer Tabelle wie in der nächsten Abbildung zu sehen übergeben.

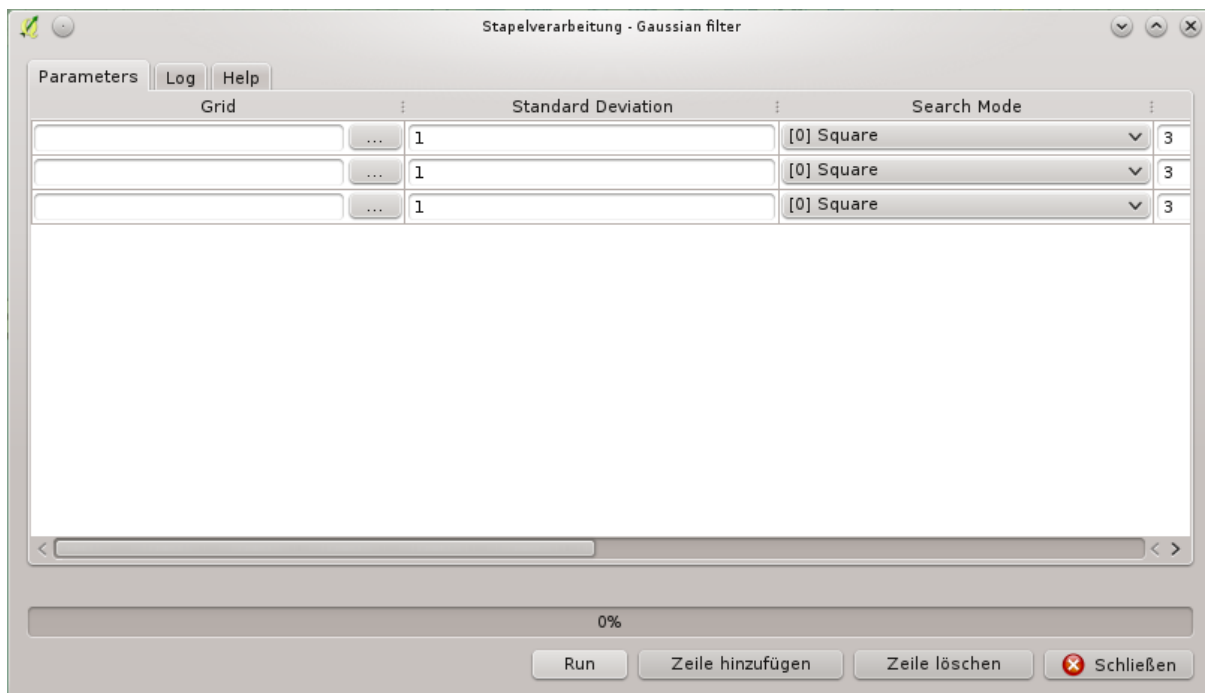


Abbildung 17.25: Batch Processing

Jede Zeile dieser Tabelle stellt eine einzelne Ausführung des Algorithmus dar, und jede Zelle enthält den Wert eines der Parameter. Es ist vergleichbar mit dem Parameter-Dialog, den Sie sehen, wenn Sie die Ausführung eines Algorithmus aus der Toolbox starten, aber mit einer anderen Anordnung.


Standardmäßig enthält die Tabelle nur zwei Zeilen. Sie können weitere hinzufügen oder entfernen mit den Tasten am unteren Teil des Fensters.

Sobald die Größe der Tabelle gesetzt ist, muss sie mit den gewünschten Werten gefüllt werden.

### 17.4.3 Füllen der Parameter-Tabelle

Bei den meisten Parametern ist das Setzen des Wertes trivial. Geben Sie einfach den Wert ein oder wählen Sie diesen aus der Liste zur Verfügung stehender Optionen, abhängig vom Parametertyp, aus.

Die wichtigsten Unterschiede liegen zwischen den Parametern für Layer oder Tabellen, und in der Angabe der Ausgabedateipfade. Bezüglich Eingabelayern und Tabellen, wenn ein Algorithmus als Teil eines Batch-Prozesses ausgeführt wird, werden die Eingangsdaten direkt von den Dateien genommen, und nicht aus denen, die bereits in QGIS geladen sind. Aus diesem Grund kann jeder Algorithmus als Batch-Prozess verwendet werden, auch wenn die entsprechenden Daten nicht geöffnet sind und der Algorithmus aus der Toolbox heraus nicht ausgeführt werden kann.

Dateinamen für Eingabedatenobjekte werden direkt eingegeben, indem Sie sie eintippen oder bequemer durch das Klicken des  Knopfes auf der rechten Seite der Zelle, welcher einen typischen Dateiauswahldialog zeigt. Es können mehrere Dateien auf einmal ausgewählt werden. Wenn der Eingabeparameter ein einzelnes Datenobjekt darstellt und mehrere Dateien ausgewählt sind, wird jede von ihnen in eine separate Zeile gesetzt, und neue hinzugefügt wenn nötig. Wenn der Parameter eine Mehrfacheingabe wiedergibt werden alle ausgewählten Dateien in eine Zelle getrennt durch Semikolons (;) gesetzt.

Ausgabedaten Objekte werden immer in einer Datei gespeichert und im Gegensatz zur Ausführung eines Algorithmus aus der Toolbox, ist das vorübergehende Speichern als temporäre Datei nicht gestattet. Sie können den Namen direkt eingeben oder im Dateiauswahldialog, wenn Sie auf den zugehörigen Button klicken.

Sobald Sie die Datei auswählen, erscheint ein neuer Dialog, der Autovervollständigung von anderen Zellen in derselben Spalte (gleiche Parameter) ermöglicht.

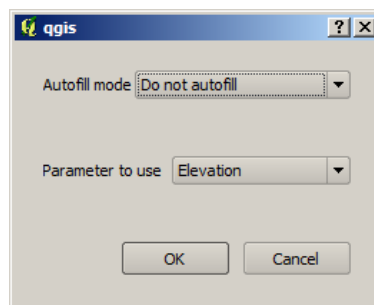


Abbildung 17.26: Batch Processing Save

Wenn der Standardwert ('Do not autocomplete') gewählt ist, wird einfach der gewählte Dateiname in die ausgewählte Zelle der Parameters Tabelle eingegeben. Wenn eine der anderen Optionen gewählt ist, werden alle Zellen unter der ausgewählten automatisch anhand von definierten Kriterien ausgefüllt. So ist es erheblich einfacher die Tabelle auszufüllen und der Stapelprozeß kann mit weniger Aufwand definiert werden.

Automatische Füllung kann durch einfaches Hinzufügen von korrelativen Zahlen zum ausgewählten Dateipfad oder Anhängen des Wertes eines anderen Feldes zu derselben Zeile erreicht werden. Dies ist besonders nützlich für die Benennung von Ausgabedaten-Objekte entsprechend der Eingänge.

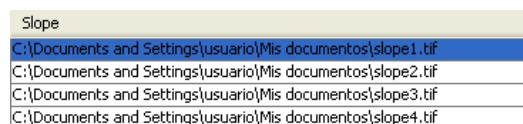


Abbildung 17.27: Batch Processing Dateipfad 

### 17.4.4 Ausführen eines Batch-Prozesses

Um den Stapelprozess, nachdem Sie alle nötigen Werte eingeführt haben, auszuführen, klicken Sie einfach auf [OK]. Der Fortschritt dieses globalen Batch Tasks wird in der Statusleiste im unteren Teil des Dialogs gezeigt.

## 17.5 Verarbeitung Algorithmen von der Konsole aus verwenden

Die Konsole ermöglicht es fortgeschrittenen Anwendern ihre Produktivität zu erhöhen und komplexe Operationen, die nicht anhand eines der GUI Elemente der Verarbeitung Umgebung ausgeführt werden können, durchzuführen. Modelle mit mehreren Algorithmen können anhand der Kommandozeilenschnittstelle definiert werden und zusätzliche Operationen wie Schleifen und Bedingungssätze können hinzugefügt werden, um flexiblere und leistungsfähigere Workflows zu erstellen.

Es gibt in QGIS keine Verarbeitung Konsole, aber alle Verarbeitung Befehle stehen anstedessen in der eingebauten QGIS Python Konsole zur Verfügung. Das heisst dass Sie diese Befehle in Ihre Konsolenarbeit integrieren können und Verarbeitung Algorithmen mit allen anderen dort zur Verfügung stehenden Objekten (einschließlich Methoden aus der QGIS API) verbinden können.

Der Code den Sie von der Python Konsole aus ausführen können, auch wenn er keine spezifische Verarbeitung Methode aufruft, kann in einen neuen Algorithmus überführt werden den Sie später aus der Werkzeugkiste, der Grafischen Modellierung oder jeder anderen Komponente aufrufen können, so wie Sie es mit jedem anderen Algorithmus tun würden. In der Tat sind einige Algorithmen, die Sie in der Werkzeugkiste finden können einfache Scripte.

In diesem Abschnitt werden wir uns angucken wie man Verarbeitung Algorithmen aus der QGIS Python Konsole heraus verwendet und auch wie man Algorithmen in Python schreibt.

### 17.5.1 Algorithmen von der Python Konsole aus aufrufen

Das erste, was Sie machen müssen, ist die Verarbeitung Funktionen mit der folgenden Zeile importieren:

```
>>> import processing
```

Nun gibt es im Grunde nur eine (interessante) Sache, die Sie damit von der Konsole aus tun können: einen Algorithmus ausführen. Das wird anhand der `runalg()` Methode gemacht, die den Namen des Algorithmus nimmt um ihn als seinen ersten Parameter auszuführen und dann eine variable Anzahl von zusätzlichen Parametern abhängig von den Anforderungen des Algorithmus. Das erste was Sie also wissen müssen ist der Name des Algorithmus, der ausgeführt werden soll. Das ist nicht der Name, den Sie in der Werkzeugkiste sehen können, sondern ein eindeutiger Kommandozeilenname. Um den richtigen Namen für Ihren Algorithmus zu finden, können Sie die `algslist()` Methode verwenden. Geben Sie den folgenden Satz in Ihre Konsole ein:

```
>>> processing.algslist()
```

Sie werden etwas wie dieses sehen.

```
Accumulated Cost (Anisotropic)----->saga:accumulatedcost (anisotropic)
Accumulated Cost (Isotropic)----->saga:accumulatedcost (isotropic)
Add Coordinates to points----->saga:addcoordinatestopoints
Add Grid Values to Points----->saga:addgridvaluestopoints
Add Grid Values to Shapes----->saga:addgridvaluestoshapes
Add Polygon Attributes to Points----->saga:addpolygonattributestopoints
Aggregate----->saga:aggregate
Aggregate Point Observations----->saga:aggregatepointobservations
Aggregation Index----->saga:aggregationindex
Analytical Hierarchy Process----->saga:analyticalhierarchyprocess
Analytical Hillshading----->saga:analyticalhillshading
Average With Mask 1----->saga:averagewithmask1
Average With Mask 2----->saga:averagewithmask2
Average With Thershold 1----->saga:averagewiththershold1
Average With Thershold 2----->saga:averagewiththershold2
Average With Thershold 3----->saga:averagewiththershold3
```



```
B-Spline Approximation----->saga:b-splineapproximation
...
```

Das ist eine Liste von allen zur Verfügung stehenden Algorithmen, alphabetisch geordnet, zusammen mit ihren entsprechenden Kommandozeilennamen.

Sie können einen String als Parameter für diese Methode benutzen. Statt die ganze Liste von Algorithmen herauszugeben wird sie nur die zeigen die diesen String enthalten. Wenn Sie beispielsweise nach einem Algorithmus zum Berechnen der Hangneigung aus einem DEM suchen, geben Sie `alglis t("slope")` ein, um folgendes Ergebnis zu erhalten:

```
DTM Filter (slope-based)----->saga:dtmfilter(slope-based)
Downslope Distance Gradient----->saga:downslopedistancegradient
Relative Heights and Slope Positions----->saga:relativeheightsandlopepositions
Slope Length----->saga:sloplength
Slope, Aspect, Curvature----->saga:slopeaspectcurvature
Upslope Area----->saga:upslopearea
Vegetation Index[slope based]----->saga:vegetationindex[slopebased]
```

Dieses Ergebnis kann sich abhängig von den Ihnen zur Verfügung stehenden Algorithmen ändern.

Es ist jetzt einfacher den Algorithmus und seinem Kommandozeilennamen, nach dem Sie suchen, zu finden, in diesem Fall `saga:slopeaspectcurvature`.

Wenn Sie erstmal den Kommandozeilennamen des Algorithmus wissen ist das nächste, was Sie tun müssen die richtige Syntax, um ihn auszuführen, zu bestimmen. Das heißt zu wissen welche Parameter benötigt werden und die Reihenfolge, in der sie eingegeben werden müssen, wenn die `runalg()` Methode aufgerufen wird. Es gibt eine Methode, um den Algorithmus im Detail zu beschreiben, die dazu benutzt werden kann um eine Liste von Parametern, die ein Algorithmus erfordert und die Ausgaben, die er erzeugt, zu erstellen. Um diese Information zu erhalten können Sie die `alghelp(name_of_the_algorithm)` Methode verwenden. Verwenden Sie den Kommandozeilennamen des Algorithmus, nicht den vollen beschreibenden Namen.

Wenn Sie die Methode mit `saga:slopeaspectcurvature` als Parameter aufrufen erhalten Sie die folgende Beschreibung:

```
>>> processing.alghelp("saga:slopeaspectcurvature")
ALGORITHM: Slope, Aspect, Curvature
  ELEVATION <ParameterRaster>
  METHOD <ParameterSelection>
  SLOPE <OutputRaster>
  ASPECT <OutputRaster>
  CURV <OutputRaster>
  HCURV <OutputRaster>
  VCURV <OutputRaster>
```

Jetzt haben Sie alles was Sie brauchen um einen Algorithmus auszuführen. Wie wir bereits erwähnt haben, gibt es nur einen einzigen Befehl um einen Algorithmus auszuführen: `runalg()`. Seine Syntax ist wie folgt:

```
>>> processing.runalg(name_of_the_algorithm, param1, param2, ..., paramN,
  Output1, Output2, ..., OutputN)
```

Die Liste von Parametern und Ausgaben hängen von dem Algorithmus den Sie ausführen wollen ab und ist genau die Liste die die `alghelp()` Methode Ihnen herausgibt, in der gleichen Reihenfolge wie gezeigt.

Abhängig vom Parametertyp werden Werte verschieden eingeführt. Die nächste Liste gibt einen kurzen Überblick darüber wie man Werte für jeden Typ von Eingabeparameter einführt:

- Rasterlayer, Vektorlayer oder Tabelle. Verwenden Sie einfach einen String mit dem Namen, der das zu verwendende Datenobjekt (den Namen, den es im QGIS Inhaltsverzeichnis hat) identifiziert oder einen Dateinamen (wenn der entsprechende Layer nicht geöffnet ist, wird er geöffnet aber nicht dem Kartenfenster hinzugefügt). Wenn Sie eine Instanz eines QGIS Objektes, das den Layer repräsentiert, haben, können Sie es als Parameter eingeben. Wenn der Input optional ist und Sie kein Datenobjekt verwenden wollen, benutzen Sie `None`.

- Auswahl. Wenn ein Algorithmus einen Auswahlparameter besitzt, sollte der Wert des Parameters eingegeben werden, indem man einen Integerwert verwendet. Um die zur Verfügung stehenden Optionen zu ermitteln, können Sie den `algorithms()` Befehl verwenden, wie im folgenden Beispiel gezeigt:

```
>>> processing.algorithms("saga:slopeaspectcurvature")
METHOD(Method)
0 - [0] Maximum Slope (Travis et al. 1975)
1 - [1] Maximum Triangle Slope (Tarboton 1997)
2 - [2] Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996)
3 - [3] Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985)
4 - [4] Fit 2.Degree Polynom (Heerdegen & Beran 1982)
5 - [5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
6 - [6] Fit 3.Degree Polynom (Haralick 1983)
```

In diesem Fall hat der Algorithmus einen solchen Parameter mit sieben Optionen. Beachten Sie dass die Bearbeitungsreihenfolge bei null anfängt.

- Multiple input. Der Wert ist ein String mit Eingabebeschreibungen getrennt durch Semikolons (;). Wie im Fall von einfachen Layern oder Tabellen kann jede Eingabebeschreibung der Datenobjektname oder sein Dateipfad sein.
- Tabellen Feld von XXX. Verwenden Sie einen String mit dem Namen des Feldes, das benutzt werden soll. Dieser Parameter unterscheidet zwischen Groß- und Kleinschreibung.
- Fixed Table. Geben Sie die Liste aller Tabellenwerte, die durch Kommas (,) getrennt sind und zwischen Anführungsstrichen (") eingeschlossen sind. Die Werte beginnen in der oberen Zeile und gehen von links nach rechts. Sie können auch einen 2D-Array von Werten, die die Tabelle repräsentieren, verwenden.
- CRS. Geben Sie den EPSG Code des gewünschten KRS ein.
- Extent. Sie müssen einen String mit `xmin`, `xmax`, `ymin` und `ymax` Werten getrennt durch Kommas (,) eingeben.

Boolean, file, string und numerical parameters brauchen keien zusätzlichen Erläuterungen.

Input parameters such as strings, booleans, or numerical values have default values. To use them, specify `None` in the corresponding parameter entry.

Für Ausgabedatenobjekte geben Sie den Dateipfad, der benutzt wird um es zu speichern, ein, genauso wie es in der Werkzeugkiste gemacht wird. Wenn Sie das Ergebnis in einer temporären Datei speichern wollen, verwenden sie `None`. Die Erweiterung der Datei bestimmt das Dateiformat. Wenn Sie eine Dateierweiterung, die nicht vom Algorithmus unterstützt wird, eingeben, wird das Standarddateiformat für den Ausgabebetyp verwendet und seine entsprechende Erweiterung an den gegebenen Dateipfad angehängt.

Im Gegensatz zum Ausführen aus der Werkzeugkiste werden Ausgaben nicht zum Kartenfenster hinzugefügt, wenn Sie den gleichen Algorithmus aus der Python Konsole ausführen. Wenn Sie eine Ausgabe dem Kartenfenster hinzufügen wollen, müssen Sie das selber nach dem Ausführen des Algorithmus machen. Sie können dafür QGIS API Befehle, oder noch einfacher eine der praktischen Methoden, die für solche Aufgaben bereitgestellt werden, verwenden.

Die `runalg` Methode gibt ein Wörterbuch mit den Ausgabenamen (diejenigen die in the Algorithmusbeschreibung gezeigt werden) als Schlüssel und die Dateipfade dieser Ausgaben als Werte zurück. Sie können diese Layer laden indem Sie die entsprechenden Dateipfade an die `load()` Methode übergeben.

## 17.5.2 Zusätzliche Funktionen für den Umgang mit Daten

Abgesehen von den Funktionen, die verwendet werden um Algorithmen aufzurufen, importiert das Importieren des `processing` Paketes einige zusätzliche Funktionen, die es einfacher machen mit Daten zu arbeiten, insbesondere Vektordaten. Sie sind einfach Komfortfunktionen, die etwas Funktionalität aus der QGIS API zusammenfassen, für gewöhnlich mit einer weniger komplexen Syntax. Diese Funktionen sollten verwendet werden wenn neue Algorithmen entwickelt werden, das sie es einfacher machen mit den Eingabedaten zu verfahren.

Unten steht eine Liste mit diesen Befehlen. Mehr Informationen können Sie in den Klassen im `processing/tools` Paket und auch in den Beispielscripts, die mit QGIS bereitgestellt werden, finden.

- `getObject(obj)`: Returns a QGIS object (a layer or table) from the passed object, which can be a filename or the name of the object in the QGIS Table of Contents.
- `values(layer, fields)`: Gibt die Werte für die zutreffenden Felder aus der Attributtabelle eines Vektorlayers zurück. Felder können als Feldnamen oder als null-basierte Feldindizes zurückgegeben werden. Es gibt ein Wörterbuch von Listen zurück, mit den zurückgegebenen Feld Identifiers als Schlüssel. Es wird die bestehende Auswahl berücksichtigt.
- `features(layer)`: Returns an iterator over the features of a vector layer, considering the existing selection.
- `uniqueValues(layer, field)`: Returns a list of unique values for a given attribute. Attributes can be passed as a field name or a zero-based field index. It considers the existing selection.

### 17.5.3 Scripte erstellen und diese aus der Werkzeugkiste starten.

Sie können Ihre eigenen Algorithmen erstellen indem Sie entsprechenden Python Code schreiben und einige zusätzliche Zeilen schreiben um zusätzliche Informationen, die benötigt werden um die Semantik des Algorithmus zu definieren, zu liefern. Sie können ein *Create new script* Menü in der *Tools* Gruppe im *Script* Algorithmus Block der Werkzeugkiste finden. Machen Sie einen Doppelklick darauf um den Skripteditor Dialog zu öffnen. Dort sollten Sie Ihren Code eingeben. Wenn Sie von dort aus mit der Erweiterung `.py` in den Ordner `scripts` (der Standardordner wenn Sie den Datei speichern Dialog öffnen) speichern wird automatisch ein entsprechender Algorithmus erstellt.

Der Name des Algorithmus (derjenige den Sie in der Werkzeugkiste sehen) wird aus dem Dateinamen erstellt, seine Erweiterung entfernt und Unterstriche mit Leerzeichen ersetzt.

Lassen Sie uns einen Blick auf den folgenden Code, der den Topographic Wetness Index (TWI) direkt von einem DGM berechnet, werfen.

```
##dem=raster
##twi=output
ret_slope = processing.runalg("saga:slopeaspectcurvature", dem, 0, None,
                             None, None, None, None)
ret_area = processing.runalg("saga:catchmentarea(mass-fluxmethod)", dem,
                             0, False, False, None, None, None, None)
processing.runalg("saga:topographicwetnessindex(twi), ret_slope['SLOPE'],
                 ret_area['AREA'], None, 1, 0, twi)
```

Wie Sie sehen können beinhaltet die Berechnung drei Algorithmen, die alle aus SAGA kommen. Der letzte davon berechnet den TWI, er braucht aber einen Hangneigungslayer und einen Abflussakkumulationslyer. Diese Layer haben wir nicht, aber da ein DGM zur Verfügung steht, können wir sie berechnen indem wir die entsprechenden SAGA Algorithmen aufrufen.

Der Teil des Codes wo dieses Verarbeiten stattfindet ist nicht schwer zu verstehen wenn Sie die vorherigen Abschnitte in diesem Kapitel gelesen haben. Die ersten Zeilen jedoch erfordern einige zusätzliche Erklärungen. Sie stellen Informationen, die für das Umwandeln Ihres Codes in einen Algorithmus, der von jeder der GUI Komponenten, wie die Werkzeugkiste oder die Grafische Modellierung, ausgeführt werden kann, zur Verfügung.

Diese Zeilen beginnen mit einem doppelten Python Kommentar (`“##”`) und haben die folgende Struktur:

```
[parameter_name]=[parameter_type] [optional_values]
```

Hier ist eine Liste von allen Parametertypen, die in Verarbeitung Scripts unterstützt werden, mit Ihrer Syntax und einigen Beispielen.

- `raster`. Ein Rasterlayer.
- `vector`. Ein Vektorlayer.
- `table`. Eine Tabelle.
- `number`. Ein numerischer Wert. Es muss ein Standardwert vergeben werden. Beispielsweise `depth=number 2.4`.

- `string`. Ein Text String. Wie im Fall von numerischen Werten muss ein Standardwert eingefügt werden. Beispielsweise `name=string Victor`.
- `boolean`. Ein boolescher Wert. Geben Sie `True` oder `False` danach ein um den Standardwert zu setzen. Zum Beispiel `verbose=boolean True`.
- `multiple raster`. Ein Satz von Eingaberasterlayern.
- `multiple vector`. Ein Satz von Eingabevektorlayern.
- `field`. Ein Feld in der Attributtabelle eines Vektorlayers. Der Name des Layers muss nach dem `field` Tag hinzugefügt werden. Beispielsweise wenn Sie eine Vektoreingabe mit `mylayer=vector` deklariert haben, könnten Sie `myfield=field mylayer` verwenden um ein Feld dieses Layers als Parameter hinzuzufügen.
- `folder`. Ein Ordner.
- `file`. Ein Dateiname.

Der Parametername ist der Name, der dem Anwender gezeigt wird wenn er den Algorithmus ausführt und gleichzeitig der Variablenname der verwendet werden muss, um den Script Code zu verwenden.

Wenn der Name des Parameters dem Anwender gezeigt wird, wird der Name bearbeitet um seine Erscheinung zu verbessern, dabei werden Unterstriche mit Leerzeilen ersetzt. Wenn Sie also beispielsweise wollen, dass der Anwender einen Parameter mit Namen `A numerical value` sieht, können Sie den Variablennamen `A_numerical_value` verwenden.

Layer und Tabellenwerte sind Strings, die einen Dateipfad des entsprechenden Objekts enthalten. Um Sie zu einem QGIS Objekt zu machen können Sie die `processing.getObjectFromUri()` Funktion verwenden. Mehrfacheingaben haben auch einen String Wert, welcher die Dateipfade zu allen ausgewählten Objekten, getrennt durch Semikolons (;), enthält.

Ausgaben werden auf ähnliche Weise mit Hilfe der folgenden Tags definiert:

- `output raster`
- `output vector`
- `output table`
- `output html`
- `output file`
- `output number`
- `output string`

Der der Ausgabevariable zugewiesene Wert ist immer ein String mit einem Dateipfad. Er entspricht einem temporären Dateipfad wenn der Anwender nicht einen Ausgabedateinamen eingegeben hat.

Wenn Sie eine Ausgabe deklarieren wird der Algorithmus versuchen ihn zu QGIS hinzuzufügen nachdem sie beendet wurde. Darum wird auch, trotzdem die `runalg()` Methode nicht die Layer, die sie produziert, lädt, der endgültige TWI Layer geladen (indem der Fall unseres vorherigen Beispiels verwendet wird), da er in die Datei gespeichert wird, die von Anwender eingegeben wird, welche der Wert der entsprechenden Ausgabe ist.

Verwenden Sie die `load()` Methode nicht in Ihren Script Algorithmen, nur wenn Sie mit der Kommandozeile arbeiten. Wenn ein Layer als Ausgabe eines Algorithmus erstellt wird, sollte er als solcher deklariert sein. Andernfalls werden Sie nicht in der Lage sein den Algorithmus richtig in der Modellierung zu verwenden, da seine Syntax (die oben erklärt durch Tags definiert wird) nicht das treffen wird was der Algorithmus wirklich erstellt.

Versteckte Ausgaben (Nummern und Strings) haben keinen Wert. Ansteddessen müssen Sie ihnen einen Wert zuweisen. Um das zu tun setzen Sie einfach den Wert einer Variablen mit dem Namen, den Sie verwendet haben, um die Ausgabe zu deklarieren. Wenn Sie beispielsweise diese Deklaration verwendet haben,

```
##average=output number
```

die folgende Zeile setzt den Wert der Ausgabe auf 5:

```
average = 5
```

Zusätzlich zu den Tags für Parameter und Ausgaben können Sie zusätzlich die Gruppe, in der der Algorithmus dann gezeigt wird, definieren, indem Sie den `group` Tag benutzen.

Wenn Ihr Algorithmus lange Zeit zum Berechnen braucht ist es eine gute Idee den Anwender zu benachrichtigen. Es steht Ihnen eine Globale genannt `progress` mit zwei möglichen Methoden zur Verfügung: `setText(text)` und `setPercentage(percent)` mit denen Sie den `progress` Text und die `progress` Leiste verändern können.

Mehrere Beispiele werden zur Verfügung gestellt. Bitte gehen Sie sie durch um anhand von echten Beispielen zu erfahren wie man Algorithmen anhand der Verarbeitung Umgebung Klassen erstellt. Sie können auf jeden Script Algorithmus rechtecklicken und *Edit script* wählen um seinen Code anzuschauen oder ihn zu bearbeiten.

## 17.5.4 Ihre Scripte dokumentieren

Wie bei Modellen können Sie zusätzliche Dokumentationen für Ihre Scripts erstellen, um zu erklären was sie tun und wie man sie benutzen kann. Im Skript-Editor Dialog werden Sie einen **[Skripthilfe bearbeiten]** Knopf finden. Klicken Sie darauf und er wird Sie zum Hilfe-Editor Dialog bringen. Gehen Sie den Abschnitt über die Grafische Modellierung durch um mehr über diesen Dialog und seine Anwendung zu erfahren.

Hilfedateien werden in dem gleichen Ordner wie das Script selber unter Hinzufügen der `.help` Erweiterung gespeichert. Beachten Sie dass Sie Ihre Script Hilfe bearbeiten können bevor Sie das Script zum ersten Mal speichern. Wenn Sie später den Script-Editor Dialog ohne das Script zu speichern (z.B. wenn Sie es verwerfen) schließen geht der Hilfeinhalt, den Sie geschrieben haben, verloren. Wenn Ihr Script bereits gespeichert wurde und mit einem Dateinamen versehen wurde, läuft das Speichern des Hilfeinhalts automatisch ab.

## 17.5.5 Pre- und Post-execution Script Hooks

Scripts können auch verwendet werden um Pre- und Post-execution Hooks, die vor und nach einem Algorithmus ausgeführt werden, zu setzen. Dies kann dafür benutzt werden um Aufgaben zu automatisieren wann immer ein Algorithmus ausgeführt wird.

Die Syntax ist identisch zu der oben erklärten Syntax, es steht aber eine zusätzliche Globalvariable genannt `alg` zur Verfügung, die den Algorithmus, der gerade ausgeführt wurde (oder ausgeführt wird), repräsentiert.

In der *General* Gruppe des Verarbeitungsoptionen Dialogs finden Sie zwei Einträge genannt *Pre-execution script* und *Post-execution script* wo die Dateinamen der Scripts, die ausgeführt werden sollen, in jedem Fall eingegeben werden können.

## 17.6 Das Protokoll

### 17.6.1 Das Verarbeitung Protokoll

Jedes mal wenn Sie einen Algorithmus ausführen werden Informationen über den Prozess im Protokoll gespeichert. Zusammen mit den verwendeten Parametern werden auch das Datum und die Zeit der Ausführung gespeichert.

Auf diese Weise ist es einfach alle Arbeiten, die entwickelt wurden anhand der Verarbeiten Umgebung, zu verfolgen und zu kontrollieren und sie auf einfache Art und Weise zu reproduzieren.

Das Protokoll ist ein Satz von Registryeinträgen, die entsprechend ihres Ausführungsdatums gruppiert werden, um leicht Informationen über einen Algorithmus, der zu einem bestimmten Moment ausgeführt wurde, zu finden.

Prozessinformationen werden als Kommandozeilen-Ausdruck gespeichert, auch wenn der Algorithmus aus der Toolbox gestartet wurde. Dies macht es auch nützlich zu lernen, wie man die Kommandozeilen-Schnittstelle

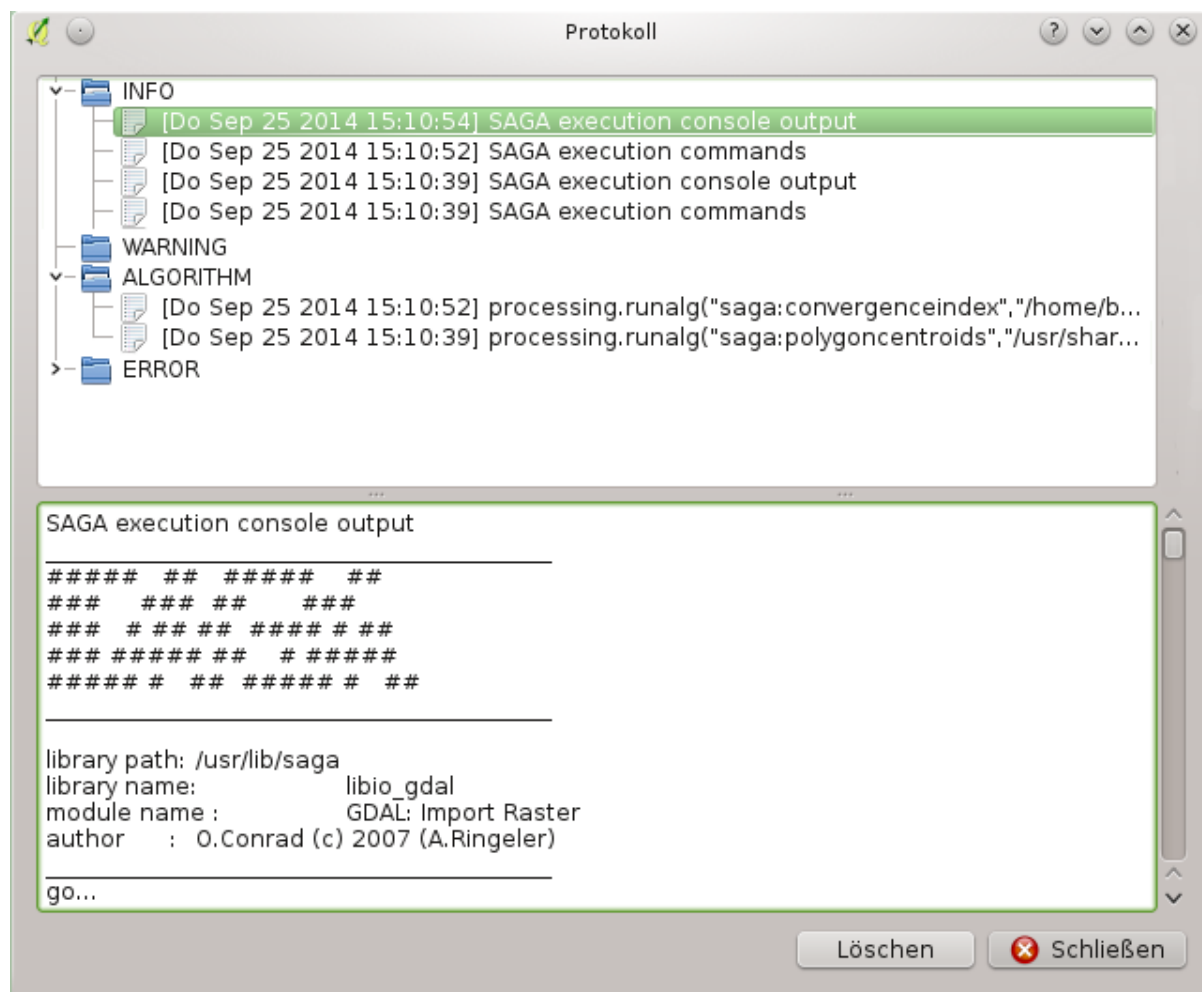



Abbildung 17.28: Protokoll 

verwendet, da man damit auch einen Algorithmus mit der Toolbox anrufen und dann im History Manager sehen und prüfen kann, wie der Algorithmus von der Kommandozeile aus aufgerufen werden kann.

Abgesehen vom Durchsuchen der Einträge in der Registry können Sie auch Prozesse wiederausführen, indem Sie auf den entsprechenden Eintrag doppelklicken.

Zusammen mit der Aufzeichnung von Algorithmusausführungen kommuniziert die Verarbeitung Umgebung mit dem Anwender anhand von anderen Gruppen der Registry, nämlich *Errors*, *Warnings* und *Information*. Für den Fall dass etwas nicht richtig funktioniert kann ein Blick auf die *Errors* vielleicht helfen herauszufinden, was passiert. Wenn Sie in Kontakt mit einem Entwickler kommen, um einen Bug oder Fehler zu melden, wird die Information in dieser Gruppe für sie/ihn sehr nützlich sein um herauszufinden, was falsch läuft.

Algorithmen von Drittanbietern werden normalerweise ausgeführt, indem ihre Kommandozeilenschnittstellen, die mit dem Anwender über die Konsole kommunizieren, aufgerufen werden. Trotzdem diese Konsole nicht gezeigt wird, wird ein vollständiger Auszug davon in der *Information* Gruppe jedesmal wenn Sie einen dieser Algorithmen ausführen, gespeichert. Wenn Sie beispielsweise Problem beim Ausführen eines SAGA Algorithmus haben, suchen Sie nach einem Eintrag genannt 'SAGA execution console output', um alle von SAGA erzeugten Nachrichten zu überprüfen, und um zu versuchen herauszufinden wo das Problem ist.

Einige Algorithmen können, selbst wenn Sie ein Ergebnis mit den gegebenen Eingabedaten erzeugen können, einen Kommentar oder zusätzliche Informationen an den *Warning* Block hinzufügen, wenn sie potentielle Probleme mit den Daten auffinden, um Sie zu warnen. Vergewissern Sie sich, dass Sie diese Nachrichten überprüfen, wenn Sie unerwartete Ergebnisse erhalten.

## 17.7 Writing new Processing algorithms as python scripts

You can create your own algorithms by writing the corresponding Python code and adding a few extra lines to supply additional information needed to define the semantics of the algorithm. You can find a *Create new script* menu under the *Tools* group in the *Script* algorithms block of the toolbox. Double-click on it to open the script edition dialog. That's where you should type your code. Saving the script from there in the `scripts` folder (the default one when you open the save file dialog), with `.py` extension, will automatically create the corresponding algorithm.

The name of the algorithm (the one you will see in the toolbox) is created from the filename, removing its extension and replacing low hyphens with blank spaces.

Let's have the following code, which calculates the Topographic Wetness Index (TWI) directly from a DEM

```
##dem=raster
##twi=output raster
ret_slope = processing.runalg("saga:slopeaspectcurvature", dem, 0, None,
                             None, None, None, None)
ret_area = processing.runalg("saga:catchmentarea", dem,
                             0, False, False, False, False, None, None, None, None)
processing.runalg("saga:topographicwetnessindextwi", ret_slope['SLOPE'],
                 ret_area['AREA'], None, 1, 0, twi)
```

As you can see, it involves 3 algorithms, all of them coming from SAGA. The last one of them calculates the TWI, but it needs a slope layer and a flow accumulation layer. We do not have these ones, but since we have the DEM, we can calculate them calling the corresponding SAGA algorithms.

The part of the code where this processing takes place is not difficult to understand if you have read the previous chapter. The first lines, however, need some additional explanation. They provide the information that is needed to turn your code into an algorithm that can be run from any of the GUI components, like the toolbox or the graphical modeler.

These lines start with a double Python comment symbol (##) and have the following structure

```
[parameter_name]=[parameter_type] [optional_values]
```

Here is a list of all the parameter types that are supported in processign scripts, their syntax and some examples.

- `raster`. A raster layer



- `vector`. A vector layer
- `table`. A table
- `number`. A numerical value. A default value must be provided. For instance, `depth=number 2.4`
- `string`. A text string. As in the case of numerical values, a default value must be added. For instance, `name=string Victor`
- `longstring`. Same as `string`, but a larger text box will be shown, so it is better suited for long strings, such as for a script expecting a small code snippet.
- `boolean`. A boolean value. Add `True` or `False` after it to set the default value. For example, `verbose=boolean True`.
- `multiple raster`. A set of input raster layers.
- `multiple vector`. A set of input vector layers.
- `field`. A field in the attributes table of a vector layer. The name of the layer has to be added after the `field` tag. For instance, if you have declared a vector input with `mylayer=vector`, you could use `myfield=field mylayer` to add a field from that layer as parameter.
- `folder`. A folder
- `file`. A filename
- `crs`. A Coordinate Reference System

The parameter name is the name that will be shown to the user when executing the algorithm, and also the variable name to use in the script code. The value entered by the user for that parameter will be assigned to a variable with that name.

When showing the name of the parameter to the user, the name will be edited to improve its appearance, replacing low hyphens with spaces. So, for instance, if you want the user to see a parameter named `A numerical value`, you can use the variable name `A_numerical_value`.

Layers and tables values are strings containing the filepath of the corresponding object. To turn them into a QGIS object, you can use the `processing.getObjectFromUri()` function. Multiple inputs also have a string value, which contains the filepaths to all selected objects, separated by semicolons (`;`).

Outputs are defined in a similar manner, using the following tags:

- `output raster`
- `output vector`
- `output table`
- `output html`
- `output file`
- `output number`
- `output string`
- `output extent`

The value assigned to the output variables is always a string with a filepath. It will correspond to a temporary filepath in case the user has not entered any output filename.

In addition to the tags for parameters and outputs, you can also define the group under which the algorithm will be shown, using the `group` tag.

The last tag that you can use in your script header is `##nomodeler`. Use that when you do not want your algorithm to be shown in the modeler window. This should be used for algorithms that do not have a clear syntax (for instance, if the number of layers to be created is not known in advance, at design time), which make them unsuitable for the graphical modeler



## 17.8 Handing data produced by the algorithm

When you declare an output representing a layer (raster, vector or table), the algorithm will try to add it to QGIS once it is finished. That is the reason why, although the `runalg()` method does not load the layers it produces, the final *TWI* layer will be loaded, since it is saved to the file entered by the user, which is the value of the corresponding output.

Do not use the `load()` method in your script algorithms, but just when working with the console line. If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since its syntax (as defined by the tags explained above) will not match what the algorithm really creates.

Hidden outputs (numbers and strings) do not have a value. Instead, it is you who has to assign a value to them. To do so, just set the value of a variable with the name you used to declare that output. For instance, if you have used this declaration,

```
##average=output number
```

the following line will set the value of the output to 5:

```
average = 5
```

## 17.9 Communicating with the user

If your algorithm takes a long time to process, it is a good idea to inform the user. You have a global named `progress` available, with two available methods: `setText(text)` and `setPercentage(percent)` to modify the progress text and the progress bar.

If you have to provide some information to the user, not related to the progress of the algorithm, you can use the `setInfo(text)` method, also from the `progress` object.

If your script has some problem, the correct way of propagating it is to raise an exception of type `GeoAlgorithmExecutionException()`. You can pass a message as argument to the constructor of the exception. Processing will take care of handling it and communicating with the user, depending on where the algorithm is being executed from (toolbox, modeler, Python console...)

## 17.10 Documenting your scripts

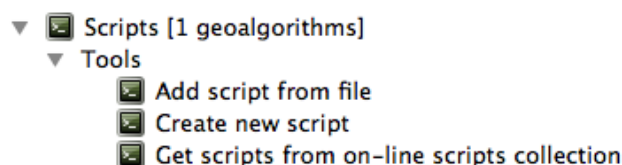
As in the case of models, you can create additional documentation for your script, to explain what they do and how to use them. In the script editing dialog you will find a **[Edit script help]** button. Click on it and it will take you to the help editing dialog. Check the chapter about the graphical modeler to know more about this dialog and how to use it.

Help files are saved in the same folder as the script itself, adding the `.help` extension to the filename. Notice that you can edit your script's help before saving it for the first time. If you later close the script editing dialog without saving the script (i.e. you discard it), the help content you wrote will be lost. If your script was already saved and is associated to a filename, saving is done automatically.

## 17.11 Beispielskripte

Several examples are available in the on-line collection of scripts, which you can access by selecting the *Get script from on-line script collection* tool under the *Scripts/tools* entry in the toolbox.

Please, check them to see real examples of how to create algorithms using the processing framework classes. You can right-click on any script algorithm and select *Edit script* to edit its code or just to see it.



## 17.12 Best practices for writing script algorithms

Here's a quick summary of ideas to consider when creating your script algorithms and, especially, if you want to share with other QGIS users. Following these simple rules will ensure consistency across the different Processing elements such as the toolbox, the modeler or the batch processing interface.

- Do not load resulting layers. Let Processing handle your results and load your layers if needed.
- Always declare the outputs your algorithm creates. Avoid things such as declaring one output and then using the destination filename set for that output to create a collection of them. That will break the correct semantics of the algorithm and make it impossible to use it safely in the modeler. If you have to write an algorithm like that, make sure you add the `##nomodeler` tag.
- Do not show message boxes or use any GUI element from the script. If you want to communicate with the user, use the `setInfo()` method or throw an `GeoAlgorithmExecutionException`
- As a rule of thumb, do not forget that your algorithm might be executed in a context other than the Processing toolbox.

## 17.13 Pre- and post-execution script hooks

Scripts can also be used to set pre- and post-execution hooks that are run before and after an algorithm is run. This can be used to automate tasks that should be performed whenever an algorithm is executed.

The syntax is identical to the syntax explained above, but an additional global variable named `alg` is available, representing the algorithm that has just been (or is about to be) executed.

In the *General* group of the processing config dialog you will find two entries named *Pre-execution script file* and *Post-execution script file* where the filename of the scripts to be run in each case can be entered.

## 17.14 Konfiguration externer Anwendungen

Die Verarbeiten Umgebung kann anhand von zusätzlichen Anwendungen erweitert werden. Derzeit werden SAGA, GRASS, OTB (Orfeo Toolbox) und R zusammen mit einigen anderen Kommandozeilenanwendungen, die räumliche Datenanalysefunktionalitäten bereitstellen, unterstützt. Algorithmen, die von einer externen Anwendung herrühren, werden von ihrem eigenen Algorithmusprovider verwaltet.

Dieses Kapitel zeigt Ihnen, wie Sie die Verarbeiten Umgebung so konfigurieren, dass diese zusätzlichen Anwendungen integriert sind, zudem werden einige Besonderheiten zu den Algorithmen, die auf ihnen basieren, erklären. Wenn das System korrekt konfiguriert ist, sind Sie in der Lage, externe Algorithmen von jeder Verarbeiten Komponente, wie die Toolbox oder der grafischen Modellierung, auszuführen, so wie Sie es mit jedem anderen Geo-Algorithmus gewohnt sind.

Standardmäßig sind alle Algorithmen, die aus einer externen Anwendung kommen und nicht in QGIS enthalten sind, deaktiviert. Sie können Sie in den Verarbeitungsoptionen aktivieren. Stellen Sie sicher, dass die entsprechende Anwendung schon auf Ihrem System installiert ist. Das Aktivieren des Algorithmusproviders ohne das Installieren der Anwendung, die benötigt wird, bewirkt, dass Algorithmen in der Werkzeugkiste erscheinen, es wird dann aber ein Fehler ausgegeben wenn Sie versuchen sie auszuführen.

Das ist so, weil die Beschreibung des Algorithmus (erforderlich, um den Parameter Dialog zu erstellen und die benötigten Informationen bereitzustellen) nicht in jeder Applikation enthalten ist, dafür aber in QGIS. Das heißt, sie sind Teil von QGIS, so haben Sie sie in Ihrer Installation, auch wenn Sie keine andere Software installiert haben. Zum Ausführen des Algorithmus müssen jedoch auch die Binärdateien der Anwendung auf Ihrem System installiert sein.

### 17.14.1 Ein Hinweis für Windows Anwender

Wenn Sie kein fortgeschrittener Anwender sind und QGIS unter Windows benutzen, sind Sie vielleicht nicht daran interessiert, den Rest dieses Kapitels zu lesen. Stellen Sie sicher, dass Sie QGIS auf Ihrem System mit Hilfe des Standalone Installers installiert haben. Das wird automatisch SAGA, GRASS und OTB auf Ihrem System installieren und sie konfigurieren, so dass sie von QGIS aus zum Laufen gebracht werden können. Alle Algorithmen in der simplified Ansicht der Werkzeugkiste werden fertig zum Ausführen ohne weitere Konfigurationen sein. Wenn Sie über die OSGeo4W Anwendung installieren stellen Sie sicher dass SAGA und OTB zur Installation ebenfalls ausgewählt sind.

Wenn Sie mehr darüber, wie die Provider arbeiten, wissen wollen oder wenn Sie einige Algorithmen, die nicht in der simplified Werkzeugkiste enthalten sind (wie R Scripts), benutzen wollen, lesen Sie weiter.

### 17.14.2 Eine Bemerkung zu den Datenformaten

Bei Verwendung einer externen Software, bedeutet das Öffnen einer Datei in QGIS nicht, dass es auch von der anderen Software geöffnet und verarbeitet werden kann. In den meisten Fällen können sie zwar auch lesen, was Sie in QGIS geöffnet haben, aber in einigen Fällen ist das vielleicht nicht der Fall. Bei der Verwendung von Datenbanken oder seltenen Dateiformaten, ob für Raster- oder Vektorlayer, könnten Probleme auftreten. Wenn das passiert, versuchen Sie, bekannte Dateiformate, bei denen Sie sicher sind, dass sie von beiden Programmen verstanden werden, zu verwenden, und überprüfen Sie die Ausgabe (im History und Log-Dialog) für weitere Informationen.

Das Verwenden von RASS Rasterlayern ist beispielsweise ein Fall, in dem Sie vielleicht Probleme haben und nicht in der Lage sein werden Ihre Arbeit zu vervollständigen, wenn Sie einen externen Algorithmus mit einem solchen Layer als Eingabe aufrufen. Aus diesem Grund erscheinen diese Layer als nicht zur Verfügung stehend für die Algorithmen.

Bei Vektorlayern sollten jedoch keine Probleme auftreten, da QGIS diese automatisch vom ursprünglichen Dateiformat zu einem von der externen Anwendung akzeptierten Format konvertiert bevor es den Layer übergibt. Dies benötigt zusätzliche Bearbeitungszeit, die erheblich sein kann, wenn der Layer groß ist. Also nicht wundern, wenn es länger braucht, um einen Layer aus einer DB-Verbindung zu analysieren, als bei einem ähnlich großen Shapefile.

Provider, die keine externe Anwendung verwenden können jeden beliebigen Layer, den Sie in QGIS öffnen können, prozessieren, da Sie ihn für Analysen anhand von QGIS öffnen.

In bezug auf Ausgabeformate können alle von QGIS als Ausgabe unterstützen Formate verwendet werden, für Raster- und Vektorlayer. Einige Provider unterstützen keine bestimmten Formate, aber alle können in gängige Rasterlayerformate exportieren, die später von QGIS automatisch transformiert werden können. Wie im Fall von Eingabelayern kann das, wenn eine Konvertierung benötigt wird, die Rechenzeit erhöhen.

Wenn die Erweiterung des Dateinamens, der festgelegt wird, wenn ein Algorithmus nicht die Erweiterung eines der von QGIS unterstützten Formate besitzt, dann wird ein Suffix hinzugefügt um das Standardformat zu setzen. Im Fall von Rasterlayern wird die Erweiterung `.tif` verwendet, während `.shp` für Vektorlayer verwendet wird.

### 17.14.3 Eine Bemerkung zu Vektorlayern mit ausgewählten Objekten

Externe Anwendungen können auch über die Auswahl die in Vektorlayern innerhalb von QGIS vorliegt unterrichtet werden. Das jedoch erfordert das Umschreiben aller Eingabevektorlayer, so als würden Sie ursprünglich in

einem Format, das nicht von der externen Anwendung unterstützt wird, vorliegen. Nur wenn eine Auswahl besteht, oder die *Use only selected features* Option in den Verarbeitungsoptionen nicht aktiviert ist, kann ein Layer direkt an eine externe Anwendung übergeben werden.

In anderen Fällen wird das Exportieren von nur ausgewählten Objekten benötigt, was bewirkt, dass sich die Ausführungszeiten verlängern.

### SAGA

SAGA Algorithmen können von QGIS aus benutzt werden wenn Sie SAGA auf Ihrem System installiert haben und Sie die Verarbeitung Umgebung richtig konfigurieren, so dass es ausführbare Module von SAGA finden kann. Insbesondere das SAGA Kommandozeilenmodul wird benötigt um SAGA Algorithmen zum Laufen zu bringen.

Wenn Sie Windows benutzen beinhalten sowohl das Installationspaket und der OSGeo4W Installer SAGA unter QGIS und er Pfad ist automatisch konfiguriert, es muss also nichts weiter getan werden.

Wenn Sie SAGA selber installiert haben (denken Sie daran, dass Sie Version 2.1 brauchen) muss der Pfad zu den ausführbaren Modulen konfiguriert werden. Um dies zu tun öffnen Sie den Optionen und Konfiguration Dialog. Im SAGA Block finden Sie eine Einstellung genannt *SAGA Folder*. Geben Sie den Pfad zu dem Ordner wo SAGA installiert ist an. Schließen Sie den Verarbeitungsoptionendialog und jetzt können Sie SAGA Algorithmen aus QGIS heraus ausführen.

Wenn Sie Linux benutzen, sind SAGA Binärpakete nicht in der Verarbeitung Umgebung enthalten, also müssen Sie die Software selber downloaden und installieren. Bitte überprüfen Sie die SAGA Webseite für mehr Informationen. Es wird SAGA 2.1 benötigt.

In diesem Fall braucht der Pfad zu den ausführbaren SAGA Modulen nicht konfiguriert werden und Sie werden diese Ordner nicht sehen. Anstedessen müssen Sie sicherstellen, dass SAGA richtig installiert ist und seine Ordner zur PATH Umgebungsvariable hinzugefügt wurden. Öffnen Sie einfach eine Konsole und geben Sie `saga_cmd` ein um zu überprüfen, dass das System herausfindet, wo sich die SAGA Binärpakete befinden.

### 17.14.4 Über die SAGA KBS Einschränkungen

Die meisten SAGA Algorithmen, die mehrere Eingaberasterlayer benötigen, haben die Anforderung, dass diese das gleiche Grid-System haben. Das heisst, dass Sie das gleiche geografische Gebiet abdecken und die gleiche Zellgröße besitzen müssen, so dass Ihre korrespondierenden Grids zueinanderpassen. Wenn SAGA Algorithmen von QGIS aus aufgerufen werden, können Sie jeden Layer unabhängig von Zellgröße und Ausdehnung benutzen. Wenn Mehrfachrasterlayer als Eingabe für einen SAGA Algorithmus verwendet werden, resampled QGIS sie zu einem gemeinsamen Grid-System und übergibt diese dann an SAGA (es sei denn der SAGA Algorithmus kann mit Layern aus verschiedenen Grid-Systemen operieren).

Die Definition dieses gemeinsamen Grid-Systems wird durch den Anwender kontrolliert und Sie werden dafür mehrere Parameter in der SAGA Gruppe im Einstellungen Fenster finden. Es gibt zwei Wege das Ziel-Grid-System einzustellen:

- Manuelles Einstellen. Sie definieren das Ausmaß indem Sie die Werte der folgenden Parameter setzen:
  - *Resampling min X*
  - *Resampling max X*
  - *Resampling min Y*
  - *Resampling max Y*
  - *Resampling cellsize*

Beachten Sie, dass QGIS alle Eingabelayer bis zu diesem Ausmaß resampled, auch wenn sie sich nicht damit überschneiden.

- Automatisches Einstellen aus Eingabelayern. Um diese Option auszuwählen, überprüfen Sie einfach die *Use min covering grid system for resampling* Option. Alle anderen Einstellungen werden ignoriert und das minimale Ausmaß, das alle Eingabelayer abdeckt, wird benutzt. Die Zellgröße des Ziellayers ist das Maximum aller Zellgrößen des Eingabelayers.

Für Algorithmen, die nicht mit mehreren Rasterlayern arbeiten, oder für diejenigen, die kein eindeutiges KRS brauchen, wird kein Resampling vor dem Aufruf von SAGA durchgeführt, und die Parameter werden nicht verwendet.

### 17.14.5 Einschränkungen für Multi-Band-Layer

Im Gegensatz zu QGIS hat SAGA keine Unterstützung für Multikanallayer. Wenn Sie einen Multikanallayer benutzen wollen (so wie ein RGB oder ein Multispektralbild), müssen Sie ihn erst in Einkanalbilder aufspalten. Um das zu tun, können Sie den 'SAGA/Grid - Tools/Split RGB image' Algorithmus (der drei Bilder aus einem RGB Bild erstellt) oder den 'SAGA/Grid - Tools/Extract band' Algorithmus (um einen einzelnen Kanal zu extrahieren) verwenden.

### 17.14.6 Einschränkungen in der Zellgröße

SAGA geht davon aus, dass Rasterlayer die selbe Pixelgröße in X- und Y-Richtung haben. Wenn Sie mit einem Layer mit unterschiedlichen Werten für die horizontale und vertikale Pixelgröße arbeiten, erhalten Sie möglicherweise unerwartete Ergebnisse. In diesem Fall wird eine Warnung im Verarbeitung Protokoll hinzugefügt, das anzeigt, dass ein Layer möglicherweise nicht geeignet ist, von SAGA verarbeitet zu werden.

### 17.14.7 Logging

Wenn QGIS SAGA aufruft, findet das über die Kommandozeilen-Schnittstelle statt mit einer Reihe von Befehlen, um alle erforderlichen Operation durchzuführen. SAGA zeigt den Fortschritt, indem es Informationen an die Konsole übergibt, die den Prozentsatz der Verarbeitung beinhaltet, zusammen mit zusätzlichen Inhalten. Diese Ausgabe wird gefiltert und verwendet, um die Fortschrittsanzeige zu aktualisieren, während der Algorithmus läuft.

Sowohl die von QGIS gesendeten Befehle und die von SAGA gedruckte zusätzliche Information kann mit anderen Verarbeitung Protokollnachrichten gelogged werden und Sie könnten es nützlich finden im Detail zu verfolgen was passiert wenn QGIS einen SAGA Algorithmus aufruft. Sie werden zwei Einstellungen finden, nämlich *Log console output* und *Log execution commands* um diesen Protokollierungsmechanismus zu aktivieren.

Die meisten anderen Provider, die eine externe Anwendung verwenden und diese aus der Kommandozeile aufrufen, haben ähnliche Optionen, also finden Sie sie auch an anderer Stelle in der Verarbeitungsoptionen Liste.

## R. Creating R scripts

Die R Integration in QGIS unterscheidet sich von der SAGA Integration, da es keinen vordefinierten Satz von Algorithmen gibt, den man ausführen kann (bis auf einige Beispiele). Stattdessen sollten Sie Ihre Skripte schreiben und dann über R Befehle aufrufen, ähnlich wie man es von R kennt, und in einer sehr ähnlichen Weise zu dem, was wir sahen, im Kapitel Verarbeitung Skripte. Dieses Kapitel zeigt Ihnen die Syntax, die Sie verwenden sollten, um die R-Befehle von QGIS anzusprechen und wie QGIS Objekte verwendet werden (Layer, Tabellen).

Das erste, was Sie tun müssen, wie wir im Fall von SAGA gesehen haben, ist zu sagen, wo sich die R Binärdateien befinden. Sie können dies tun, indem Sie den *R folder* Eintrag im Verarbeitung Konfigurationsdialog ansprechen. Nachdem Sie die Parameter definiert haben, können Sie anfangen, eigene R-Skripte zu erstellen und auszuführen.

Dies ist wieder anders in Linux. Dort müssen Sie nur sicherstellen, dass der R-Ordner in der Umgebungsvariablen PATH enthalten ist. Wenn Sie nur die Eingabe R in eine Konsole eingeben können, kann es losgehen.

Um einen neuen Algorithmus hinzuzufügen, der eine R Funktion (oder ein komplexeres RScript, das sie entwickelt haben und das sie gerne zur in QGIS zur Verfügung stehen haben wollen) aufruft, müssen Sie eine Script Datei erstellen, die der Verarbeiten Umgebung, wie diese Operation durchgeführt wird, und die entsprechenden R Befehle mitteilt um das zu tun.

R Scripts Dateien habe die Erweiterung `.rsx` und es ist ziemlich einfach sie zu erstellen wenn Sie nur etwas grundlegende Kenntnisse über die R Syntax und R Scripting besitzen. Sie sollten im R Scripts Ordner gespeichert werden. Sie können diesen Ordner in der R Einstellungen Gruppe (steht im Verarbeitungsoptionen Dialog) einstellen, genauso wie Sie es mit dem Ordner für reguläre Verarbeitung Scripts machen.

Werfen Sie einen Blick auf eine sehr einfache Scripts-Datei, die die R-Methode `spsample` aufruft und ein zufälliges Gitter innerhalb der Grenzen eines Polygon in einem bestimmten Polygon-Layer erstellt. Diese Methode gehört zum Paket zur `MapTools`. Grundsätzlich brauchen Sie für R-Algorithmen, die Sie in Verarbeitung für räumliche geostatistische Analysen nutzen wollen, zumindest Kenntnisse der Pakete wie `MapTools` und speziell `sp`.

```
##polyg=vector
##numpoints=number 10
##output=output vector
##sp=group
pts=spsample(polyg,numpoints,type="random")
output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

Die ersten Zeilen, die mit einem doppelten Python Kommentarzeichen (`##`) beginnen teilen QGIS die Eingaben des Algorithmus, der in der Datei beschrieben wird und die Ausgaben, die er erzeugen wird, mit. Sie arbeiten mit genau der gleichen Syntax wie den Verarbeitung Scripts, die wir bereits gesehen haben, also werden Sie hier nicht noch einmal beschrieben.

Wenn Sie einen Eingabeparameter definieren, verwendet QGIS diese Informationen für zwei Dinge: Erstellung der Benutzeroberfläche, um die Benutzer nach Eingaben von Parameters zu fragen und die Schaffung einer entsprechenden R Variable, die später als Eingabe für die R-Befehle verwendet werden.

In dem obigen Beispiel erklären wir die Eingabe vom Typ `Vektor` mit dem Namen `polyg`. Bei der Ausführung des Algorithmus wird durch QGIS R geöffnet und der Layer, der durch den Benutzer ausgewählt wurde, in einer Variablen `polyg` abgelegt. Der Name des Parameters ist also auch der Name der Variablen, die wir in R für den Zugriff auf den Wert des Parameters nutzen (vermeiden Sie, reservierte R Begriffe als Parameter zu verwenden).

Räumliche Elemente wie Vektor- und Rasterlayer werden mit den `readOGR()` und `brick()` Befehlen eingelesen (sie brauchen sich nicht darum kümmern diese Befehle Ihrer Beschreibungsdatei hinzuzufügen - das macht QGIS) und sie werden als `Spatial*DataFrame` Objekte gespeichert. Tabellenfelder werden als Strings, die den Namen des ausgewählten Feldes beinhalten, gespeichert.

Tabellen werden anhand des `read.csv()` Befehls geöffnet. Wenn ein Anwender eine Tabelle eingibt, die nicht im CSV Format ist, wird diese vor dem Importieren in R konvertiert.

Zusätzlich können Rasterdateien mit Hilfe des `readGDAL()` Befehls anstelle von `brick()` eingelesen werden indem `##userreadgdal` benutzt wird.

Wenn Sie ein fortgeschrittener Anwender sind und nicht wollen, dass QGIS ein Objekt, das den Layer repräsentiert, erstellt, können Sie den `##passfilename` Tag benutzen um anzuzeigen, dass Sie anstedessen einen String mit dem Dateinamen bevorzugen. In diesem Fall ist es an Ihnen, die Datei vor dem Durchführen von Operationen mit Daten, die es beinhaltet, zu öffnen.

Mit den oben genannten Informationen können wir jetzt die erste Zeile unseres ersten Beispiel Scripts verstehen (dabei startet die erste Zeile nicht mit einem Python Kommentar).

```
pts=spsample(polyg,numpoints,type="random")
```

Die Variable `Polygon` enthält bereits ein `SpatialPolygonsDataFrame` Objekt, damit kann es verwendet werden, um das `spsample` Verfahren aufzurufen, ebenso wie die `NumPoints`, die die Anzahl der Punkte zeigt, die zum Gitter hinzugefügt werden sollen.

Da wir eine Ausgabe vom Typ `Vektor` definiert haben mit dem Namen `out`, müssen wir nun eine Variable namens `out` erstellen und `Spatial*DATAframe` Objekt dafür erstellen (in diesem Fall ein `SpatialPointsDataFrame`). Sie können einen beliebigen Namen für Ihre Zwischenvariablen verwenden. Stellen Sie nur sicher, dass die Variable Ihres Endergebnisses den gleichen Namen hat, den Sie zuvor definiert haben und dass sie einen geeigneten Wert enthält.

In diesem Fall muss das von der `spsample` Methode erhaltene Ergebnis ausdrücklich in ein `SpatialPointsDataFrame` Objekt konvertiert werden, da es ein Objekt der Klasse `ppp` ist, welches keine



geeignete Klasse ist um an QGIS zurückgegeben zu werden.

Wenn Ihr Algorithmus Rasterlayer erstellt, hängt die Art und Weise wie Sie gespeichert werden davon ab ob Sie die `#dontuserasterpackage` Option verwendet haben. Wenn Sie sie verwendet haben werden die Layer anhand der `writeGDAL()` Methode gespeichert. Wenn nicht wird die `writeRaster()` Methode aus dem `raster` Paket benutzt.

Wenn Sie die `#passfilename` Option verwendet haben, werden Ausgaben anhand des `raster` Paketes erstellt (mit `writeRaster()`), trotzdem es nicht für Eingaben verwendet wird.

Wenn Ihr Algorithmus keinen Layer erzeugt sondern stattdessen ein Textergebnis in der Konsole, müssen Sie angeben, dass die Konsole gezeigt wird, nachdem die Ausführung beendet ist. Um das zu tun starten Sie die Kommandozeilen, die die Ergebnisse produzieren, die Sie mit dem `>` (größer) Zeichen drucken wollen. Die Ausgabe aller anderen Zeilen wird nicht gezeigt. Zum Beispiel ist hier die Beschreibungsdatei eines Algorithmus, der einen Test auf Normalverteilung mit einem vorhandenen Feld (Spalte) eines Vektorlayerattributes durchführt:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

Die Ausgabe der letzten Zeile wird gedruckt, die Ausgabe der ersten aber nicht (und auch nicht die Ausgaben von anderen Kommandozeilen, die automatisch von QGIS hinzugefügt werden).

Wenn Ihr Algorithmus eine Grafiken erstellt (mit der `plot()`-Methode), fügen Sie die folgende Zeile ein:

```
##showplots
```

Dies bewirkt, dass QGIS alle grafischen Ausgaben von R in eine temporäre Datei, die geöffnet wird nachdem die Ausführung von R beendet wurde, umleitet.

Sowohl grafische als auch Konsolenergebnisse werden im Verarbeitung Ergebnisanzeige gezeigt.

Für weitere Informationen schauen Sie sich bitte die Skript-Dateien an, die Verarbeitung bereits enthält. Die meisten von ihnen sind ziemlich einfach und werden wesentlich dazu beitragen, wie Sie Ihre eigenen erstellen.

---

**Bemerkung:** `rgdal` and `mapproj` Bibliotheken werden standardmäßig geladen, Sie müssen also nicht die entsprechenden `library()` Befehle hinzufügen (Sie müssen nur sicherstellen, dass diese zwei Pakete in Ihrer R-Distribution installiert sind). Jedoch müssen andere zusätzliche Bibliotheken, die Sie vielleicht brauchen, explizit geladen werden. Geben Sie einfach die notwendigen Befehle am Anfang Ihres Scripts ein. Auch müssen Sie sicherstellen, dass die entsprechenden Pakete in der R-Distribution, die von QGIS genutzt wird, installiert sind. Die Verarbeitung Umgebung kümmert sich nicht um Paketinstallationen. Wenn Sie ein Skript ausführen, das ein Paket erfordert, das nicht installiert ist, wird die Ausführung fehlschlagen und die Verarbeitung wird versuchen, zu ermitteln welche Pakete fehlen. Sie müssen diese Bibliotheken manuell installieren bevor sie den Algorithmus ausführen können.

---

## GRASS

Die Konfiguration von GRASS ist nicht viel anders als die Konfiguration von SAGA. Zuerst muss der Pfad zu dem Ordner GRASS definiert werden, aber nur, wenn Sie Windows verwenden. Zusätzlich gibt es einen Shell-Interpreter (in der Regel: `msys.exe`, den Sie in den meisten GRASS für Windows-Distributionen finden) der definiert werden muss, genauso wie der Pfad dorthin.

Standardmäßig versucht die Verarbeitung Umgebung seine GRASS Verbindung so zu konfigurieren, dass die GRASS-Distribution, die mit QGIS geliefert wird, benutzt werden kann. Dies sollte ohne Probleme in den meisten Systemen funktionieren, aber wenn Probleme auftauchen, müssen Sie vielleicht die GRASS Verbindung manuell konfigurieren. Auch wenn Sie eine andere GRASS Installation benutzen wollen, können Sie diese Einstellung ändern und auf den Ordner , wo die andere Version installiert ist, verweisen. Damit die Algorithmen korrekt funktionieren wird GRASS 6.4. benötigt.

Wenn Sie Linux verwenden, müssen Sie nur noch sicherstellen, dass GRASS richtig installiert ist, und dass es problemlos von einer Konsole ausgeführt werden kann.

GRASS Algorithmen verwenden eine Region für die Berechnungen. Diese Region kann manuell definiert werden unter Verwendung von Werten ähnlich denen, die in der SAGA Konfiguration stehen oder automatisch, wobei die minimale Ausdehnung aller Eingangslayer verwendet werden, wenn der Algorithmus ausgeführt wird. Wenn dies das Verhalten ist, das Sie bevorzugen, können Sie es über die Option *Verwende min abdecken Region* in den GRASS Konfigurationsparametern definieren.

Der letzte Parameter, der konfiguriert werden muss, bezieht sich auf die Mapset. Eine Mapset wird benötigt um GRASS benutzen zu können und die Verarbeiten Umgebung erstellt eine temporäre für jede Ausführung. Sie müssen angeben, ob die Daten, mit denen Sie arbeiten Geografische (lat/lon) Koordinaten oder projizierte verwenden.

### GDAL

Ist keine zusätzliche Konfiguration erforderlich, um GDAL Algorithmen laufen zu lassen. Da es bereits in QGIS integriert ist, können die Algorithmen ihre Konfiguration davon ableiten.



### Orfeo Toolbox

Orfeo Toolbox (OTB) Algorithmen können von QGIS aus ausgeführt werden wenn Sie OTB auf Ihrem System installiert haben und Sie QGIS richtig konfiguriert haben, so dass es alle notwendigen Dateien finden kann (Kommandozeilentools und Bibliotheken).

Wie im Fall von SAGA sind OTB Binärpakete im Installationspaket für Windows enthalten, sie sind aber nicht enthalten wenn Sie Linux benutzen, also müssen Sie die Software selber downloaden und installieren. Bitte überprüfen Sie die OTB Webseite für weitere Informationen.

Starten Sie QGIS nachdem OTB installiert wurde, öffnen Sie den Verarbeitungsoptionen Dialog und konfigurieren Sie den OTB Algorithmus Provider. Im *Orfeo Toolbox (Image analysis)* Block finden Sie alle für OTB relevanten Einstellungen. Stellen Sie zuerst sicher, dass die Algorithmen aktiviert sind.

Konfigurieren Sie dann den Pfad zu dem Ordner wo die ORB Kommandozeilentools und -bibliotheken installiert sind:

-  Normalerweise verweist der *OTB applications folder* zu `/usr/lib/otb/applications` und der *OTB command line tools folder* zu `/usr/bin`.
-  Wenn Sie den OSGeo4W Installer benutzen, installieren Sie das `otb-bin` Paket und geben Sie `C:\OSGeo4W\apps\orfeotoolbox\applications` als *OTB applications folder* und `C:\OSGeo4W\bin` als *OTB command line tools folder* ein. Diese Werte sollten standardmäßig konfiguriert werden, aber wenn Sie eine abweichende OTB Installation durchgeführt haben, konfigurieren Sie sie auf die entsprechenden Werte in Ihrem System.

### TauDEM

Um diesen Provider zu verwenden müssen Sie TauDEM Kommandozeilentools installieren.

## 17.14.8 Windows

Bitte besuchen Sie die [TauDEM homepage](#) für Installationhinweise und vorkompilierte Binärpakete für 32-bit und 64-bit Systeme. **WICHTIG:** Sie benötigen die ausführbaren Dateien von TauDEM 5.0.6. Version 5.2 wird derzeit nicht unterstützt.



## 17.14.9 Linux

Es gibt keine Pakete für die meisten Linux Distributionen, Sie sollten TauDEM also selber kompilieren. Da TauDEM MPICH2 verwendet, installieren Sie dieses zuerst und verwenden Sie dabei ihren bevorzugten Paketmanager. Alternativ dazu funktioniert TauDEM gut mit Open MPI, Sie können es also anstelle von MPICH2 verwenden.

Laden Sie TauDEM 5.0.6 [source code](#) und extrahieren Sie die Dateien in einen gewissen Ordner.

Öffnen Sie die `linearpart.h` Datei, und nach der Zeile

```
#include "mpi.h"
```

fügen Sie eine neue Zeile ein mit

```
#include <stdint.h>
```

dadurch bekommen Sie

```
#include "mpi.h"
#include <stdint.h>
```

Speichern Sie die Änderungen und schließen Sie die Datei. Öffnen Sie jetzt `tiffIO.h`, finden Sie die Zeile `#include "stdint.h"` und ersetzen Sie die Anführungszeichen ("`"`") mit `<>` so dass Sie jetzt folgendes erhalten

```
#include <stdint.h>
```

Speichern Sie die Änderungen und schließen Sie die Datei. Erstellen Sie ein Build-Verzeichnis und machen Sie ein `cd` dort hin.

```
mkdir build
cd build
```

Konfigurieren Sie Ihr `build` mit dem Befehl

```
CXX=mpicxx cmake -DCMAKE_INSTALL_PREFIX=/usr/local ..
```

und dann kompilieren

```
make
```

Schließlich, um TauDEM in `/usr/local/bin` zu installieren, geben Sie ein

```
sudo make install
```

.

## 17.15 The QGIS Commander

Processing includes a practical tool that allows you to run algorithms without having to use the toolbox, but just by typing the name of the algorithm you want to run.

This tool is known as the *QGIS commander*, and it is just a simple text box with autocompletion where you type the command you want to run.

The Commander is started from the *Analysis* menu or, more practically, by pressing `Shift + Ctrl + M` (you can change that default keyboard shortcut in the QGIS configuration if you prefer a different one). Apart from executing Processing algorithms, the Commander gives you access to most of the functionality in QGIS, which means that it gives you a practical and efficient way of running QGIS tasks and allows you to control QGIS with reduced usage of buttons and menus.

Zudem ist der Commander konfigurierbar, Sie können also benutzerdefinierte Befehle hinzufügen und finden Sie dann nur ein paar Tastendrucke entfernt, was es zu einem mächtigen Werkzeug macht, das Ihnen hilft bei Ihrer täglichen Arbeit produktiver mit QGIS zu werden.

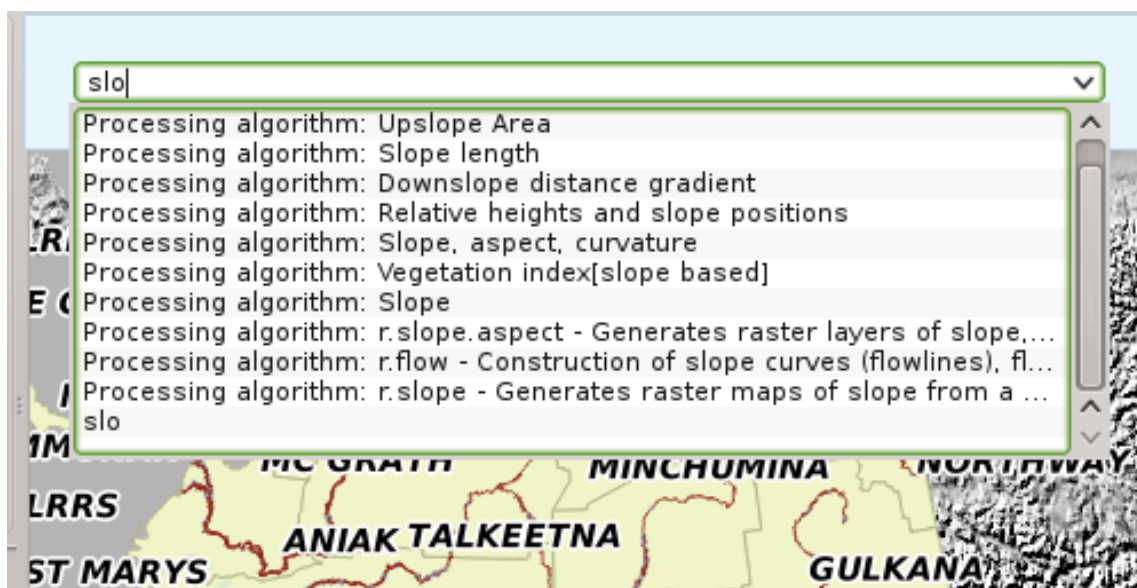



Abbildung 17.29: The QGIS Commander 

### 17.15.1 Zur Verfügung stehende Befehle

Die zur Verfügung stehenden Befehle im Commander fallen in die folgenden Kategorien:

- Processing algorithms. These are shown as `Processing algorithm: <name of the algorithm>`.
- Menu actions. Diese werden als `Menu action:<Menüeintragtext>` gezeigt. Alle zur Verfügung stehenden Menüelemente von der QGIS Oberfläche sind zugänglich, auch wenn sie sich in einem Untermenü befinden.
- Pythonfunktionen. Sie können kurze Pythonfunktionen erstellen, die dann in die Liste von zur Verfügung stehenden Befehlen eingefügt werden. Sie werden als `Function:<Funktionsname>` gezeigt.

Um einen der obigen Befehle zum Laufen zu bringen fangen Sie einfach an einzugeben und wählen Sie das entsprechende Element aus der Liste der zur Verfügung stehenden Befehle, das nach dem Filtern der ganzen Liste von Befehlen erscheint, aus.

Im Falle des Aufrufen eines Pythonfunktion können Sie den Eintrag in der Liste, dem der Begriff `Function:` vorausgeht, auswählen (zum Beispiel `Function:removeall`) oder Sie geben den Funktionsnamen (`removeall` im vorigen Beispiel) direkt ein. Sie müssen keine Klammern nach dem Funktionsnamen eingeben.

### 17.15.2 Benutzerdefinierte Funktionen erstellen

Benutzerdefinierte Funktionen werden eingegeben indem Sie den entsprechenden Pythoncode in die `commands.py` Datei, die Sie im `.qgis/sextante/commander` directory in Ihrem Benutzerverzeichnis finden, eingeben. Es ist nur eine einfache Python Datei in die Sie alle Funktionen, die Sie brauchen eingeben können.

Die Datei wird mit einigen Beispielfunktionen erstellt wenn Sie den Commander das erste Mal öffnen. Wenn Sie den Commander noch nicht geöffnet haben können Sie die Datei selber erstellen. Um die Befehlsdatei zu bearbeiten benutzen Sie Ihren bevorzugten Texteditor. Sie können auch einen integrierten Editor, indem Sie den `edit` Befehl aus dem Commander heraus aufrufen, benutzen. Dies öffnet den Editor mit der Befehlsdatei und Sie können diesen direkt bearbeiten und dann Ihre Änderungen speichern.

Beispielsweise können Sie die folgende Funktion, die alle Layer entfernt, hinzufügen.

```
from qgis.gui import *
```

```
def removeall():
    mapreg = QgsMapLayerRegistry.instance()
    mapreg.removeAllMapLayers()
```

Nachdem Sie die Funktion hinzugefügt haben, steht sie im Commander zur Verfügung und Sie können sie durch Eingabe von `removeall` aufrufen. Es braucht nichts weiter getan werden als die Funktion selber zu schreiben.

Funktionen können Parameter empfangen. Fügen Sie `*args` Ihrer Funktionsdefinition hinzu um Argumente zu empfangen. Wenn Sie die Funktion aus dem Commander aufrufen müssen Parameter getrennt durch Leerzeichen übergeben werden.

Hier ist ein Beispiel einer Funktion, die einen Layer lädt und einen Parameter mit dem Dateinamen von dem zu ladenden Layer nimmt.

```
import processing
```

```
def load(*args):
    processing.load(args[0])
```

Wenn Sie den Layer in `/home/myuser/points.shp` laden wollen, geben Sie `load /home/myuser/points.shp` in das Commander Textfeld ein.

.



---

## Processing providers and algorithms

---

### 18.1 GDAL algorithm provider

GDAL (Geospatial Data Abstraction Library) is a translator library for raster and vector geospatial data formats.

#### 18.1.1 GDAL analysis

##### Perspektive

##### Beschreibung

<fügen Sie hier die Übersetzung des Algorithmus ein>

##### Parameter

**Eingabelayer [raster]** <geben Sie hier die Parameterbeschreibung ein>

**Kanalnummer [number]** <geben Sie hier die Parameterbeschreibung ein>

Vorgabewert: *1*

**Kanten berechnen [boolean]** <geben Sie hier die Parameterbeschreibung ein>

Vorgabewert: *False*

**Zevenbergen&&Thornesche (statt Hornsche) Formel verwenden [boolean]** <geben Sie hier die Parameterbeschreibung ein>

Vorgabewert: *False*

**Trigonometrischen Winkel (statt Azimut) zurückgeben [boolean]** <geben Sie hier die Parameterbeschreibung ein>

Vorgabewert: *False*

**0 (statt -9999) für Ebene zurückgeben [boolean]** <geben Sie hier die Parameterbeschreibung ein>

Vorgabewert: *False*

## Ausgaben

**Ausgaberraster** [raster] <fügen Sie hier die Übersetzung der Ausgabe ein>

## Verwendung der Konsole

```
processing.runalg('gdalogr:aspect', input, band, compute_edges, zevenbergen, trig_angle, zero_flat)
```

## Siehe auch

### Farbre relief

### Beschreibung

<fügen Sie hier die Übersetzung des Algorithmus ein>

### Parameter

**Eingabelayer** [raster] <geben Sie hier die Parameterbeschreibung ein>

**Kanalnummer** [number] <geben Sie hier die Parameterbeschreibung ein>

Vorgabe: 1

**Kanten berechnen** [boolean] <geben Sie hier die Parameterbeschreibung ein>

Vorgabewert: *False*

**Farbkonfigurationsdatei** [file] <geben Sie hier die Parameterbeschreibung ein>

**Zuordnungsmodus** [selection] <geben Sie hier die Parameterbeschreibung ein>

Optionen:

- 0 — “0,0,0,0” RGBA
- 1 — Exact color
- 2 — Nearest color

Default: 0

### Outputs

**Output file** [raster] <put output description here>

### Console usage

```
processing.runalg('gdalogr:colorrelief', input, band, compute_edges, color_table, match_mode, out)
```

### See also

### Fill nodata

### Description

<put algorithm description here>

## Parameters

**Input layer** [raster] <put parameter description here>

**Search distance** [number] <put parameter description here>

Default: *100*

**Smooth iterations** [number] <put parameter description here>

Default: *0*

**Band to operate on** [number] <put parameter description here>

Default: *1*

**Validity mask** [raster] Optional.

<put parameter description here>

**Do not use default validity mask** [boolean] <put parameter description here>

Default: *False*

## Outputs

**Output layer** [raster] <put output description here>

## Console usage

```
processing.runalg('gdalogr:fillnodata', input, distance, iterations, band, mask, no_default_mask,
```

## See also

### Grid (Moving average)

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: point] <put parameter description here>

**Z field** [tablefield: numeric] Optional.

<put parameter description here>

**Radius 1** [number] <put parameter description here>

Default: *0.0*

**Radius 2** [number] <put parameter description here>

Default: *0.0*

**Min points** [number] <put parameter description here>

Default: *0.0*

**Angle** [number] <put parameter description here>

Default: *0.0*

**Nodata** [number] <put parameter description here>

Default: 0.0

**Output raster type** [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

## Outputs

**Output file** [raster] <put output description here>

## Console usage

```
processing.runalg('gdalogr:gridaverage', input, z_field, radius_1, radius_2, min_points, angle, n
```

## See also

## Grid (Data metrics)

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: point] <put parameter description here>

**Z field** [tablefield: numeric] Optional.

<put parameter description here>

**Metrics** [selection] <put parameter description here>

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Range



- 3 — Count
- 4 — Average distance
- 5 — Average distance between points

Default: 0

**Radius 1** [number] <put parameter description here>

Default: 0.0

**Radius 2** [number] <put parameter description here>

Default: 0.0

**Min points** [number] <put parameter description here>

Default: 0.0

**Angle** [number] <put parameter description here>

Default: 0.0

**Nodata** [number] <put parameter description here>

Default: 0.0

**Output raster type** [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

## Outputs

**Output file** [raster] <put output description here>

## Console usage

```
processing.runalg('gdalogr:griddatametrics', input, z_field, metric, radius_1, radius_2, min_point
```

See also

## Grid (Inverse distance to a power)

### Description

<put algorithm description here>

### Parameters

**Input layer** [**vector: point**] <put parameter description here>

**Z field** [**tablefield: numeric**] Optional.

<put parameter description here>

**Power** [**number**] <put parameter description here>

Default: 2.0

**Smoothing** [**number**] <put parameter description here>

Default: 0.0

**Radius 1** [**number**] <put parameter description here>

Default: 0.0

**Radius 2** [**number**] <put parameter description here>

Default: 0.0

**Max points** [**number**] <put parameter description here>

Default: 0.0

**Min points** [**number**] <put parameter description here>

Default: 0.0

**Angle** [**number**] <put parameter description here>

Default: 0.0

**Nodata** [**number**] <put parameter description here>

Default: 0.0

**Output raster type** [**selection**] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32

- 10 — CFloat64

Default: 5

## Outputs

**Output file [raster]** <put output description here>

## Console usage

```
processing.runalg('gdalogr:gridinvdist', input, z_field, power, smothing, radius_1, radius_2, max.
```

## See also

## Grid (Nearest neighbor)

### Description

<put algorithm description here>

### Parameters

**Input layer [vector: point]** <put parameter description here>

**Z field [tablefield: numeric]** Optional.

<put parameter description here>

**Radius 1 [number]** <put parameter description here>

Default: 0.0

**Radius 2 [number]** <put parameter description here>

Default: 0.0

**Angle [number]** <put parameter description here>

Default: 0.0

**Nodata [number]** <put parameter description here>

Default: 0.0

**Output raster type [selection]** <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32

- 9 — CFloat32
- 10 — CFloat64

Default: 5

## Outputs

**Output file [raster]** <put output description here>

## Console usage

```
processing.runalg('gdalogr:gridnearestneighbor', input, z_field, radius_1, radius_2, angle, nodata)
```

## See also

## Hillshade

### Description

<put algorithm description here>

### Parameters

**Input layer [raster]** <put parameter description here>

**Band number [number]** <put parameter description here>

Default: 1

**Compute edges [boolean]** <put parameter description here>

Default: *False*

**Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean]** <put parameter description here>

Default: *False*

**Z factor (vertical exaggeration) [number]** <put parameter description here>

Default: 1.0

**Scale (ratio of vert. units to horiz.) [number]** <put parameter description here>

Default: 1.0

**Azimuth of the light [number]** <put parameter description here>

Default: 315.0

**Altitude of the light [number]** <put parameter description here>

Default: 45.0

## Outputs

**Output file [raster]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:hillshade', input, band, compute_edges, zevenbergen, z_factor, scale, a
```

### See also

### Near black

#### Description

<put algorithm description here>

#### Parameters

**Input layer [raster]** <put parameter description here>

**How far from black (white) [number]** <put parameter description here>

Default: *15*

**Search for nearly white pixels instead of nearly black [boolean]** <put parameter description here>

Default: *False*

#### Outputs

**Output layer [raster]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:nearblack', input, near, white, output)
```

### See also

### Proximity (raster distance)

#### Description

<put algorithm description here>

#### Parameters

**Input layer [raster]** <put parameter description here>

**Values [string]** <put parameter description here>

Default: *(not set)*

**Dist units [selection]** <put parameter description here>

Options:

- 0 — GEO
- 1 — PIXEL

Default: 0

**Max dist (negative value to ignore) [number]** <put parameter description here>

Default: -1

**No data (negative value to ignore) [number]** <put parameter description here>

Default: -1

**Fixed buf val (negative value to ignore) [number]** <put parameter description here>

Default: -1

**Output raster type [selection]** <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

## Outputs

**Output layer [raster]** <put output description here>

## Console usage

```
processing.runalg('gdalogr:proximity', input, values, units, max_dist, nodata, buf_val, rtype, out)
```

## See also

## Roughness

## Description

<put algorithm description here>

## Parameters

**Input layer [raster]** <put parameter description here>

**Band number [number]** <put parameter description here>

Default: 1

**Compute edges** [boolean] <put parameter description here>

Default: *False*

### Outputs

**Output file** [raster] <put output description here>

### Console usage

```
processing.runalg('gdalogr:roughness', input, band, compute_edges, output)
```

### See also

### Sieve

### Description

<put algorithm description here>

### Parameters

**Input layer** [raster] <put parameter description here>

**Threshold** [number] <put parameter description here>

Default: 2

**Pixel connection** [selection] <put parameter description here>

Options:

- 0 — 4
- 1 — 8

Default: 0

### Outputs

**Output layer** [raster] <put output description here>

### Console usage

```
processing.runalg('gdalogr:sieve', input, threshold, connections, output)
```

### See also

### Slope

### Description

<put algorithm description here>

### Parameters

**Input layer** [raster] <put parameter description here>

**Band number** [number] <put parameter description here>

Default: *1*

**Compute edges** [boolean] <put parameter description here>

Default: *False*

**Use Zevenbergen&Thorne formula (instead of the Horn's one)** [boolean] <put parameter description here>

Default: *False*

**Slope expressed as percent (instead of degrees)** [boolean] <put parameter description here>

Default: *False*

**Scale (ratio of vert. units to horiz.)** [number] <put parameter description here>

Default: *1.0*

### Outputs

**Output file** [raster] <put output description here>

### Console usage

```
processing.runalg('gdalogr:slope', input, band, compute_edges, zevenbergen, as_percent, scale, out)
```

### See also

### TPI (Topographic Position Index)

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [raster] <put parameter description here>

**Band number** [number] <put parameter description here>

Default: *1*

**Compute edges** [boolean] <put parameter description here>

Default: *False*

#### Outputs

**Output file** [raster] <put output description here>



### Console usage

```
processing.runalg('gdalogr:tpitopographicpositionindex', input, band, compute_edges, output)
```

### See also

## TRI (Terrain Ruggedness Index)

### Description

<put algorithm description here>

### Parameters

**Input layer** [raster] <put parameter description here>

**Band number** [number] <put parameter description here>

Default: *1*

**Compute edges** [boolean] <put parameter description here>

Default: *False*

### Outputs

**Output file** [raster] <put output description here>

### Console usage

```
processing.runalg('gdalogr:triterrainruggednessindex', input, band, compute_edges, output)
```

### See also

.

## 18.1.2 GDAL conversion

### gdal2xyz

### Description

<put algorithm description here>

### Parameters

**Input layer** [raster] <put parameter description here>

**Band number** [number] <put parameter description here>

Default: *1*

## Outputs

**Output file** [table] <put output description here>

## Console usage

```
processing.runalg('gdalogr:gdal2xyz', input, band, output)
```

## See also

## PCT to RGB

### Description

<put algorithm description here>

### Parameters

**Input layer** [raster] <put parameter description here>

**Band to convert** [selection] <put parameter description here>

Options:

- 0 — 1
- 1 — 2
- 2 — 3
- 3 — 4
- 4 — 5
- 5 — 6
- 6 — 7
- 7 — 8
- 8 — 9
- 9 — 10
- 10 — 11
- 11 — 12
- 12 — 13
- 13 — 14
- 14 — 15
- 15 — 16
- 16 — 17
- 17 — 18
- 18 — 19
- 19 — 20
- 20 — 21

- 21 — 22
- 22 — 23
- 23 — 24
- 24 — 25

Default: 0

### Outputs

**Output layer [raster]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:pcttorgb', input, nband, output)
```

### See also

### Polygonize (raster to vector)

#### Description

<put algorithm description here>

#### Parameters

**Input layer [raster]** <put parameter description here>

**Output field name [string]** <put parameter description here>

Default: *DN*

### Outputs

**Output layer [vector]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:polygonize', input, field, output)
```

### See also

### Rasterize (vector to raster)

#### Beschreibung

<put algorithm description here>

### Parameter

**Input layer** [vector: any] <put parameter description here>

**Attribute field** [tablefield: any] <put parameter description here>

**Write values inside an existing raster layer (\*)** [boolean] <put parameter description here>

Default: *False*

**Set output raster size (ignored if above option is checked)** [selection] <put parameter description here>

Optionen:

- 0 — Ausgabegröße in Pixel
- 1 — Output resolution in map units per pixel

Vorgabe: *1*

**Horizontal** [Zahl] <put parameter description here>

Vorgabe: *100.0*

**Vertikal** [Zahl] <put parameter description here>

Vorgabe: *100.0*

**Rastertyp** [Auswahl] <put parameter description here>

Optionen:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Vorgabe: *0*

### Ausgaben

**Output layer:** mandatory to choose an existing raster layer if the (\*) option is selected  
<put output description here>

### Console usage

```
processing.runalg('gdalogr:rasterize', input, field, writeover, dimensions, width, height, rtype,
```

Siehe auch

## RGB to PCT

### Description

<put algorithm description here>

### Parameters

**Input layer** [**raster**] <put parameter description here>

**Number of colors** [**number**] <put parameter description here>

Default: 2

### Outputs

**Output layer** [**raster**] <put output description here>

### Console usage

```
processing.runalg('gdalogr:rgbtocpct', input, ncolors, output)
```

### See also

## Translate (convert format)

### Description

<put algorithm description here>

### Parameters

**Input layer** [**raster**] <put parameter description here>

**Set the size of the output file (In pixels or %)** [**number**] <put parameter description here>

Default: 100

**Output size is a percentage of input size** [**boolean**] <put parameter description here>

Default: *True*

**Nodata value, leave as none to take the nodata value from input** [**string**] <put parameter description here>

Default: *none*

**Expand** [**selection**] <put parameter description here>

Options:

- 0 — none
- 1 — gray
- 2 — rgb

- 3 — rgba

Default: 0

**Output projection for output file** [leave blank to use input projection] [crs]  
<put parameter description here>

Default: *None*

**Subset based on georeferenced coordinates** [extent] <put parameter description here>

Default: 0,1,0,1

**Copy all subdatasets of this file to individual output files** [boolean] <put parameter description here>

Default: *False*

**Additional creation parameters** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Output raster type** [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

## Outputs

**Output layer** [raster] <put output description here>

## Console usage

```
processing.runalg('gdalogr:translate', input, outsize, outsize_perc, no_data, expand, srs, projwin)
```

## See also

.

### 18.1.3 GDAL extraction

#### Clip raster by extent

##### Description

<put algorithm description here>

##### Parameters

**Input layer** [raster] <put parameter description here>

**Nodata value, leave as none to take the nodata value from input** [string] <put parameter description here>

Default: *none*

**Clipping extent** [extent] <put parameter description here>

Default: *0,1,0,1*

**Additional creation parameters** [string] Optional.

<put parameter description here>

Default: *(not set)*

##### Outputs

**Output layer** [raster] <put output description here>

##### Console usage

```
processing.runalg('gdalogr:cliprasterbyextent', input, no_data, projwin, extra, output)
```

##### See also

#### Clip raster by mask layer

##### Description

<put algorithm description here>

##### Parameters

**Input layer** [raster] <put parameter description here>

**Mask layer** [vector: polygon] <put parameter description here>

**Nodata value, leave as none to take the nodata value from input** [string] <put parameter description here>

Default: *none*

**Create and output alpha band** [boolean] <put parameter description here>

Default: *False*

**Keep resolution of output raster [boolean]** <put parameter description here>

Default: *False*

**Additional creation parameters [string]** Optional.

<put parameter description here>

Default: *(not set)*

### Outputs

**Output layer [raster]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:cliprasterbymasklayer', input, mask, no_data, alpha_band, keep_resolut
```

### See also

### Contour

### Description

<put algorithm description here>

### Parameters

**Input layer [raster]** <put parameter description here>

**Interval between contour lines [number]** <put parameter description here>

Default: *10.0*

**Attribute name (if not set, no elevation attribute is attached) [string]**

Optional.

<put parameter description here>

Default: *ELEV*

**Additional creation parameters [string]** Optional.

<put parameter description here>

Default: *(not set)*

### Outputs

**Output file for contour lines (vector) [vector]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:contour', input_raster, interval, field_name, extra, output_vector)
```



**See also**

.

## 18.1.4 GDAL miscellaneous

### Build Virtual Raster

**Description**

<put algorithm description here>

**Parameters**

**Input layers** [**multipleinput: rasters**] <put parameter description here>

**Resolution** [**selection**] <put parameter description here>

Options:

- 0 — average
- 1 — highest
- 2 — lowest

Default: *0*

**Layer stack** [**boolean**] <put parameter description here>

Default: *True*

**Allow projection difference** [**boolean**] <put parameter description here>

Default: *False*

**Outputs**

**Output layer** [**raster**] <put output description here>

**Console usage**

```
processing.runalg('gdalogr:buildvirtualraster', input, resolution, separate, proj_difference, outp
```

**See also**

### Merge

**Description**

<put algorithm description here>

### Parameters

**Input layers** [**multipleinput: rasters**] <put parameter description here>

**Grab pseudocolor table from first layer** [**boolean**] <put parameter description here>

Default: *False*

**Layer stack** [**boolean**] <put parameter description here>

Default: *False*

**Output raster type** [**selection**] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

### Outputs

**Output layer** [**raster**] <put output description here>

### Console usage

```
processing.runalg('gdalogr:merge', input, pct, separate, rtype, output)
```

### See also

#### Build overviews (pyramids)

#### Description

<put algorithm description here>

### Parameters

**Input layer** [**raster**] <put parameter description here>

**Overview levels** [**string**] <put parameter description here>

Default: 2 4 8 16

**Remove all existing overviews** [boolean] <put parameter description here>

Default: *False*

**Resampling method** [selection] <put parameter description here>

Options:

- 0 — nearest
- 1 — average
- 2 — gauss
- 3 — cubic
- 4 — average\_mp
- 5 — average\_magphase
- 6 — mode

Default: *0*

**Overview format** [selection] <put parameter description here>

Options:

- 0 — Internal (if possible)
- 1 — External (GTiff .ovr)
- 2 — External (ERDAS Imagine .aux)

Default: *0*

## Outputs

**Output layer** [raster] <put output description here>

## Console usage

```
processing.runalg('gdalogr:overviews', input, levels, clean, resampling_method, format)
```

## See also

## Information

### Description

<put algorithm description here>

### Parameters

**Input layer** [raster] <put parameter description here>

**Suppress GCP info** [boolean] <put parameter description here>

Default: *False*

**Suppress metadata info** [boolean] <put parameter description here>

Default: *False*

## Outputs

**Layer information [html]** <put output description here>

## Console usage

```
processing.runalg('gdalorg:rasterinfo', input, nogcp, nometadata, output)
```

## See also

.

## 18.1.5 GDAL projections

### Extract projection

#### Description

<put algorithm description here>

#### Parameters

**Input file [raster]** <put parameter description here>

**Create also .prj file [boolean]** <put parameter description here>

Default: *False*

#### Outputs

#### Console usage

```
processing.runalg('gdalorg:extractprojection', input, prj_file)
```

## See also

### Warp (reproject)

#### Description

<put algorithm description here>

#### Parameters

**Input layer [raster]** <put parameter description here>

**Source SRS (EPSG Code) [crs]** <put parameter description here>

Default: *EPSG:4326*

**Destination SRS (EPSG Code) [crs]** <put parameter description here>

Default: *EPSG:4326*

**Output file resolution in target georeferenced units (leave 0 for no change) [number]**  
 <put parameter description here>

Default: 0.0

**Resampling method [selection]** <put parameter description here>

Options:

- 0 — near
- 1 — bilinear
- 2 — cubic
- 3 — cubicspline
- 4 — lanczos

Default: 0

**Additional creation parameters [string]** Optional.

<put parameter description here>

Default: *(not set)*

**Output raster type [selection]** <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

## Outputs

**Output layer [raster]** <put output description here>

## Console usage

```
processing.runalg('gdalogr:warpreproject', input, source_srs, dest_srs, tr, method, extra, rtype,
```

## See also

.

## 18.1.6 OGR conversion

### Convert format

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

**Destination Format** [selection] <put parameter description here>

Options:

- 0 — ESRI Shapefile
- 1 — GeoJSON
- 2 — GeoRSS
- 3 — SQLite
- 4 — GMT
- 5 — MapInfo File
- 6 — INTERLIS 1
- 7 — INTERLIS 2
- 8 — GML
- 9 — Geoconcept
- 10 — DXF
- 11 — DGN
- 12 — CSV
- 13 — BNA
- 14 — S57
- 15 — KML
- 16 — GPX
- 17 — PGDump
- 18 — GPSTrackMaker
- 19 — ODS
- 20 — XLSX
- 21 — PDF

Default: 0

**Creation Options** [string] Optional.

<put parameter description here>

Default: *(not set)*

### Outputs

**Output layer [vector]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:convertformat', input_layer, format, options, output_layer)
```

### See also

.

## 18.1.7 OGR geoprocessing

### Clip vectors by extent

#### Description

<put algorithm description here>

#### Parameters

**Input layer [vector: any]** <put parameter description here>

**Clip extent [extent]** <put parameter description here>

Default: *0,1,0,1*

**Additional creation Options [string]** Optional.

<put parameter description here>

Default: *(not set)*

### Outputs

**Output layer [vector]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:clipvectorsbyextent', input_layer, clip_extent, options, output_layer)
```

### See also

### Clip vectors by polygon

#### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Clip layer** [vector: polygon] <put parameter description here>

**Additional creation Options** [string] Optional.

<put parameter description here>

Default: *(not set)*

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('gdalogr:clipvectorsbypolygon', input_layer, clip_layer, options, output_layer)
```

### See also

.

## 18.1.8 OGR miscellaneous

### Execute SQL

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

**SQL** [string] <put parameter description here>

Default: *(not set)*

#### Outputs

**SQL result** [vector] <put output description here>

#### Console usage

```
processing.runalg('gdalogr:executesql', input, sql, output)
```



**See also****Import Vector into PostGIS database (available connections)****Description**

<put algorithm description here>

**Parameters**

**Database (connection name) [selection]** <put parameter description here>

Options:

- 0 — local

Default: 0

**Input layer [vector: any]** <put parameter description here>

**Output geometry type [selection]** <put parameter description here>

Options:

- 0 —
- 1 — NONE
- 2 — GEOMETRY
- 3 — POINT
- 4 — LINESTRING
- 5 — POLYGON
- 6 — GEOMETRYCOLLECTION
- 7 — MULTIPOINT
- 8 — MULTIPOLYGON
- 9 — MULTILINESTRING

Default: 5

**Input CRS (EPSG Code) [crs]** <put parameter description here>

Default: *EPSG:4326*

**Output CRS (EPSG Code) [crs]** <put parameter description here>

Default: *EPSG:4326*

**Schema name [string]** Optional.

<put parameter description here>

Default: *public*

**Table name, leave blank to use input name [string]** Optional.

<put parameter description here>

Default: *(not set)*

**Primary Key [string]** Optional.

<put parameter description here>

Default: *id*

**Geometry column name** [string] Optional.

<put parameter description here>

Default: *geom*

**Vector dimensions** [selection] <put parameter description here>

Options:

- 0 — 2
- 1 — 3

Default: *0*

**Distance tolerance for simplification** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Maximum distance between 2 nodes (densification)** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Select features by extent (defined in input layer CRS)** [extent] <put parameter description here>

Default: *0,1,0,1*

**Clip the input layer using the above (rectangle) extent** [boolean] <put parameter description here>

Default: *False*

**Select features using a SQL "WHERE" statement (Ex: column="value")** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Group "n" features per transaction (Default: 20000)** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Overwrite existing table?** [boolean] <put parameter description here>

Default: *True*

**Append to existing table?** [boolean] <put parameter description here>

Default: *False*

**Append and add new fields to existing table?** [boolean] <put parameter description here>

Default: *False*

**Do not launder columns/table name/s?** [boolean] <put parameter description here>

Default: *False*

**Do not create Spatial Index?** [boolean] <put parameter description here>

Default: *False*

**Continue after a failure, skipping the failed feature** [boolean] <put parameter description here>

Default: *False*

**Additional creation options** [string] Optional.

<put parameter description here>

Default: *(not set)*

## Outputs

## Console usage

```
processing.runalg('gdalogr:importvectorintopostgisdatabaseavailableconnections', database, input_
```

## See also

## Import Vector into PostGIS database (new connection)

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Output geometry type** [selection] <put parameter description here>

Options:

- 0 —
- 1 — NONE
- 2 — GEOMETRY
- 3 — POINT
- 4 — LINESTRING
- 5 — POLYGON
- 6 — GEOMETRYCOLLECTION
- 7 — MULTIPOINT
- 8 — MULTIPOLYGON
- 9 — MULTILINESTRING

Default: 5

**Input CRS (EPSG Code)** [crs] <put parameter description here>

Default: *EPSG:4326*

**Output CRS (EPSG Code)** [crs] <put parameter description here>

Default: *EPSG:4326*

**Host** [string] <put parameter description here>

Default: *localhost*

**Port** [string] <put parameter description here>

Default: *5432*

**Username** [string] <put parameter description here>

Default: *(not set)*

**Database Name** [string] <put parameter description here>

Default: *(not set)*

**Password** [string] <put parameter description here>

Default: *(not set)*

**Schema name** [string] Optional.

<put parameter description here>

Default: *public*

**Table name, leave blank to use input name** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Primary Key** [string] Optional.

<put parameter description here>

Default: *id*

**Geometry column name** [string] Optional.

<put parameter description here>

Default: *geom*

**Vector dimensions** [selection] <put parameter description here>

Options:

- 0 — 2
- 1 — 3

Default: *0*

**Distance tolerance for simplification** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Maximum distance between 2 nodes (densification)** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Select features by extent (defined in input layer CRS)** [extent] <put parameter description here>

Default: *0,1,0,1*

**Clip the input layer using the above (rectangle) extent** [boolean] <put parameter description here>

Default: *False*

**Select features using a SQL "WHERE" statement (Ex: column="value")** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Group "n" features per transaction (Default: 20000) [string]** Optional.

<put parameter description here>

Default: *(not set)*

**Overwrite existing table? [boolean]** <put parameter description here>

Default: *True*

**Append to existing table? [boolean]** <put parameter description here>

Default: *False*

**Append and add new fields to existing table? [boolean]** <put parameter description here>

Default: *False*

**Do not launder columns/table name/s? [boolean]** <put parameter description here>

Default: *False*

**Do not create Spatial Index? [boolean]** <put parameter description here>

Default: *False*

**Continue after a failure, skipping the failed feature [boolean]** <put parameter description here>

Default: *False*

**Additional creation options [string]** Optional.

<put parameter description here>

Default: *(not set)*

## Outputs

### Console usage

```
processing.runalg('gdalogr:importvectorintopostgisdatabasewconnection', input_layer, gtype, s_s
```

### See also

## Information

### Description

<put algorithm description here>

### Parameters

**Input layer [vector: any]** <put parameter description here>

### Outputs

**Layer information [html]** <put output description here>

### Console usage

```
processing.runalg('gdalogr:information', input, output)
```

### See also

.

## 18.2 LAStools

LAStools is a collection of highly efficient, multicore command line tools for LiDAR data processing.

### 18.2.1 las2las\_filter

#### Description

<put algorithm description here>

#### Parameters

**verbose** [**boolean**] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [**file**] Optional.

<put parameter description here>

**filter (by return, classification, flags)** [**selection**] <put parameter description here>

Options:

- 0 — —
- 1 — keep\_last
- 2 — keep\_first
- 3 — keep\_middle
- 4 — keep\_single
- 5 — drop\_single
- 6 — keep\_double
- 7 — keep\_class 2
- 8 — keep\_class 2 8
- 9 — keep\_class 8
- 10 — keep\_class 6
- 11 — keep\_class 9
- 12 — keep\_class 3 4 5
- 13 — keep\_class 2 6
- 14 — drop\_class 7
- 15 — drop\_withheld

Default: 0

**second filter (by return, classification, flags) [selection]** <put parameter description here>

Options:

- 0 — —
- 1 — keep\_last
- 2 — keep\_first
- 3 — keep\_middle
- 4 — keep\_single
- 5 — drop\_single
- 6 — keep\_double
- 7 — keep\_class 2
- 8 — keep\_class 2 8
- 9 — keep\_class 8
- 10 — keep\_class 6
- 11 — keep\_class 9
- 12 — keep\_class 3 4 5
- 13 — keep\_class 2 6
- 14 — drop\_class 7
- 15 — drop\_withheld

Default: 0

**filter (by coordinate, intensity, GPS time, ...) [selection]** <put parameter description here>

Options:

- 0 — —
- 1 — clip\_x\_above
- 2 — clip\_x\_below
- 3 — clip\_y\_above
- 4 — clip\_y\_below
- 5 — clip\_z\_above
- 6 — clip\_z\_below
- 7 — drop\_intensity\_above
- 8 — drop\_intensity\_below
- 9 — drop\_gps\_time\_above
- 10 — drop\_gps\_time\_below
- 11 — drop\_scan\_angle\_above
- 12 — drop\_scan\_angle\_below
- 13 — keep\_point\_source
- 14 — drop\_point\_source
- 15 — drop\_point\_source\_above

- 16 — drop\_point\_source\_below
- 17 — keep\_user\_data
- 18 — drop\_user\_data
- 19 — drop\_user\_data\_above
- 20 — drop\_user\_data\_below
- 21 — keep\_every\_nth
- 22 — keep\_random\_fraction
- 23 — thin\_with\_grid

Default: 0

**value for filter (by coordinate, intensity, GPS time, ...)** [string] <put parameter description here>

Default: *(not set)*

**second filter (by coordinate, intensity, GPS time, ...)** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — clip\_x\_above
- 2 — clip\_x\_below
- 3 — clip\_y\_above
- 4 — clip\_y\_below
- 5 — clip\_z\_above
- 6 — clip\_z\_below
- 7 — drop\_intensity\_above
- 8 — drop\_intensity\_below
- 9 — drop\_gps\_time\_above
- 10 — drop\_gps\_time\_below
- 11 — drop\_scan\_angle\_above
- 12 — drop\_scan\_angle\_below
- 13 — keep\_point\_source
- 14 — drop\_point\_source
- 15 — drop\_point\_source\_above
- 16 — drop\_point\_source\_below
- 17 — keep\_user\_data
- 18 — drop\_user\_data
- 19 — drop\_user\_data\_above
- 20 — drop\_user\_data\_below
- 21 — keep\_every\_nth
- 22 — keep\_random\_fraction
- 23 — thin\_with\_grid

Default: 0



**value for second filter (by coordinate, intensity, GPS time, ...)** [string] <put parameter description here>

Default: *(not set)*

## Outputs

**output LAS/LAZ file** [file] <put output description here>

## Console usage

```
processing.runalg('lidartools:las2lasfilter', verbose, input_laslaz, filter_return_class_flags1, ...)
```

## See also

### 18.2.2 las2las\_project

#### Description

<put algorithm description here>

#### Parameters

**verbose** [boolean] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [file] Optional.

<put parameter description here>

**source projection** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — utm
- 2 — sp83
- 3 — sp27
- 4 — longlat
- 5 — latlong

Default: *0*

**source utm zone** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — 1 (north)
- 2 — 2 (north)
- 3 — 3 (north)
- 4 — 4 (north)
- 5 — 5 (north)
- 6 — 6 (north)

- 7 — 7 (north)
- 8 — 8 (north)
- 9 — 9 (north)
- 10 — 10 (north)
- 11 — 11 (north)
- 12 — 12 (north)
- 13 — 13 (north)
- 14 — 14 (north)
- 15 — 15 (north)
- 16 — 16 (north)
- 17 — 17 (north)
- 18 — 18 (north)
- 19 — 19 (north)
- 20 — 20 (north)
- 21 — 21 (north)
- 22 — 22 (north)
- 23 — 23 (north)
- 24 — 24 (north)
- 25 — 25 (north)
- 26 — 26 (north)
- 27 — 27 (north)
- 28 — 28 (north)
- 29 — 29 (north)
- 30 — 30 (north)
- 31 — 31 (north)
- 32 — 32 (north)
- 33 — 33 (north)
- 34 — 34 (north)
- 35 — 35 (north)
- 36 — 36 (north)
- 37 — 37 (north)
- 38 — 38 (north)
- 39 — 39 (north)
- 40 — 40 (north)
- 41 — 41 (north)
- 42 — 42 (north)
- 43 — 43 (north)
- 44 — 44 (north)
- 45 — 45 (north)

- 46 — 46 (north)
- 47 — 47 (north)
- 48 — 48 (north)
- 49 — 49 (north)
- 50 — 50 (north)
- 51 — 51 (north)
- 52 — 52 (north)
- 53 — 53 (north)
- 54 — 54 (north)
- 55 — 55 (north)
- 56 — 56 (north)
- 57 — 57 (north)
- 58 — 58 (north)
- 59 — 59 (north)
- 60 — 60 (north)
- 61 — 1 (south)
- 62 — 2 (south)
- 63 — 3 (south)
- 64 — 4 (south)
- 65 — 5 (south)
- 66 — 6 (south)
- 67 — 7 (south)
- 68 — 8 (south)
- 69 — 9 (south)
- 70 — 10 (south)
- 71 — 11 (south)
- 72 — 12 (south)
- 73 — 13 (south)
- 74 — 14 (south)
- 75 — 15 (south)
- 76 — 16 (south)
- 77 — 17 (south)
- 78 — 18 (south)
- 79 — 19 (south)
- 80 — 20 (south)
- 81 — 21 (south)
- 82 — 22 (south)
- 83 — 23 (south)
- 84 — 24 (south)

- 85 — 25 (south)
- 86 — 26 (south)
- 87 — 27 (south)
- 88 — 28 (south)
- 89 — 29 (south)
- 90 — 30 (south)
- 91 — 31 (south)
- 92 — 32 (south)
- 93 — 33 (south)
- 94 — 34 (south)
- 95 — 35 (south)
- 96 — 36 (south)
- 97 — 37 (south)
- 98 — 38 (south)
- 99 — 39 (south)
- 100 — 40 (south)
- 101 — 41 (south)
- 102 — 42 (south)
- 103 — 43 (south)
- 104 — 44 (south)
- 105 — 45 (south)
- 106 — 46 (south)
- 107 — 47 (south)
- 108 — 48 (south)
- 109 — 49 (south)
- 110 — 50 (south)
- 111 — 51 (south)
- 112 — 52 (south)
- 113 — 53 (south)
- 114 — 54 (south)
- 115 — 55 (south)
- 116 — 56 (south)
- 117 — 57 (south)
- 118 — 58 (south)
- 119 — 59 (south)
- 120 — 60 (south)

Default: 0

**source state plane code [selection]** <put parameter description here>

Options:

- 0 — —
- 1 — AK\_10
- 2 — AK\_2
- 3 — AK\_3
- 4 — AK\_4
- 5 — AK\_5
- 6 — AK\_6
- 7 — AK\_7
- 8 — AK\_8
- 9 — AK\_9
- 10 — AL\_E
- 11 — AL\_W
- 12 — AR\_N
- 13 — AR\_S
- 14 — AZ\_C
- 15 — AZ\_E
- 16 — AZ\_W
- 17 — CA\_I
- 18 — CA\_II
- 19 — CA\_III
- 20 — CA\_IV
- 21 — CA\_V
- 22 — CA\_VI
- 23 — CA\_VII
- 24 — CO\_C
- 25 — CO\_N
- 26 — CO\_S
- 27 — CT
- 28 — DE
- 29 — FL\_E
- 30 — FL\_N
- 31 — FL\_W
- 32 — GA\_E
- 33 — GA\_W
- 34 — HI\_1
- 35 — HI\_2
- 36 — HI\_3
- 37 — HI\_4
- 38 — HI\_5

- 39 — IA\_N
- 40 — IA\_S
- 41 — ID\_C
- 42 — ID\_E
- 43 — ID\_W
- 44 — IL\_E
- 45 — IL\_W
- 46 — IN\_E
- 47 — IN\_W
- 48 — KS\_N
- 49 — KS\_S
- 50 — KY\_N
- 51 — KY\_S
- 52 — LA\_N
- 53 — LA\_S
- 54 — MA\_I
- 55 — MA\_M
- 56 — MD
- 57 — ME\_E
- 58 — ME\_W
- 59 — MI\_C
- 60 — MI\_N
- 61 — MI\_S
- 62 — MN\_C
- 63 — MN\_N
- 64 — MN\_S
- 65 — MO\_C
- 66 — MO\_E
- 67 — MO\_W
- 68 — MS\_E
- 69 — MS\_W
- 70 — MT\_C
- 71 — MT\_N
- 72 — MT\_S
- 73 — NC
- 74 — ND\_N
- 75 — ND\_S
- 76 — NE\_N
- 77 — NE\_S

- 78 — NH
- 79 — NJ
- 80 — NM\_C
- 81 — NM\_E
- 82 — NM\_W
- 83 — NV\_C
- 84 — NV\_E
- 85 — NV\_W
- 86 — NY\_C
- 87 — NY\_E
- 88 — NY\_LI
- 89 — NY\_W
- 90 — OH\_N
- 91 — OH\_S
- 92 — OK\_N
- 93 — OK\_S
- 94 — OR\_N
- 95 — OR\_S
- 96 — PA\_N
- 97 — PA\_S
- 98 — PR
- 99 — RI
- 100 — SC\_N
- 101 — SC\_S
- 102 — SD\_N
- 103 — SD\_S
- 104 — St.Croix
- 105 — TN
- 106 — TX\_C
- 107 — TX\_N
- 108 — TX\_NC
- 109 — TX\_S
- 110 — TX\_SC
- 111 — UT\_C
- 112 — UT\_N
- 113 — UT\_S
- 114 — VA\_N
- 115 — VA\_S
- 116 — VT

- 117 — WA\_N
- 118 — WA\_S
- 119 — WI\_C
- 120 — WI\_N
- 121 — WI\_S
- 122 — WV\_N
- 123 — WV\_S
- 124 — WY\_E
- 125 — WY\_EC
- 126 — WY\_W
- 127 — WY\_WC

Default: 0

**target projection** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — utm
- 2 — sp83
- 3 — sp27
- 4 — longlat
- 5 — latlong

Default: 0

**target utm zone** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — 1 (north)
- 2 — 2 (north)
- 3 — 3 (north)
- 4 — 4 (north)
- 5 — 5 (north)
- 6 — 6 (north)
- 7 — 7 (north)
- 8 — 8 (north)
- 9 — 9 (north)
- 10 — 10 (north)
- 11 — 11 (north)
- 12 — 12 (north)
- 13 — 13 (north)
- 14 — 14 (north)
- 15 — 15 (north)



- 16 — 16 (north)
- 17 — 17 (north)
- 18 — 18 (north)
- 19 — 19 (north)
- 20 — 20 (north)
- 21 — 21 (north)
- 22 — 22 (north)
- 23 — 23 (north)
- 24 — 24 (north)
- 25 — 25 (north)
- 26 — 26 (north)
- 27 — 27 (north)
- 28 — 28 (north)
- 29 — 29 (north)
- 30 — 30 (north)
- 31 — 31 (north)
- 32 — 32 (north)
- 33 — 33 (north)
- 34 — 34 (north)
- 35 — 35 (north)
- 36 — 36 (north)
- 37 — 37 (north)
- 38 — 38 (north)
- 39 — 39 (north)
- 40 — 40 (north)
- 41 — 41 (north)
- 42 — 42 (north)
- 43 — 43 (north)
- 44 — 44 (north)
- 45 — 45 (north)
- 46 — 46 (north)
- 47 — 47 (north)
- 48 — 48 (north)
- 49 — 49 (north)
- 50 — 50 (north)
- 51 — 51 (north)
- 52 — 52 (north)
- 53 — 53 (north)
- 54 — 54 (north)

- 55 — 55 (north)
- 56 — 56 (north)
- 57 — 57 (north)
- 58 — 58 (north)
- 59 — 59 (north)
- 60 — 60 (north)
- 61 — 1 (south)
- 62 — 2 (south)
- 63 — 3 (south)
- 64 — 4 (south)
- 65 — 5 (south)
- 66 — 6 (south)
- 67 — 7 (south)
- 68 — 8 (south)
- 69 — 9 (south)
- 70 — 10 (south)
- 71 — 11 (south)
- 72 — 12 (south)
- 73 — 13 (south)
- 74 — 14 (south)
- 75 — 15 (south)
- 76 — 16 (south)
- 77 — 17 (south)
- 78 — 18 (south)
- 79 — 19 (south)
- 80 — 20 (south)
- 81 — 21 (south)
- 82 — 22 (south)
- 83 — 23 (south)
- 84 — 24 (south)
- 85 — 25 (south)
- 86 — 26 (south)
- 87 — 27 (south)
- 88 — 28 (south)
- 89 — 29 (south)
- 90 — 30 (south)
- 91 — 31 (south)
- 92 — 32 (south)
- 93 — 33 (south)

- 94 — 34 (south)
- 95 — 35 (south)
- 96 — 36 (south)
- 97 — 37 (south)
- 98 — 38 (south)
- 99 — 39 (south)
- 100 — 40 (south)
- 101 — 41 (south)
- 102 — 42 (south)
- 103 — 43 (south)
- 104 — 44 (south)
- 105 — 45 (south)
- 106 — 46 (south)
- 107 — 47 (south)
- 108 — 48 (south)
- 109 — 49 (south)
- 110 — 50 (south)
- 111 — 51 (south)
- 112 — 52 (south)
- 113 — 53 (south)
- 114 — 54 (south)
- 115 — 55 (south)
- 116 — 56 (south)
- 117 — 57 (south)
- 118 — 58 (south)
- 119 — 59 (south)
- 120 — 60 (south)

Default: 0

**target state plane code [selection]** <put parameter description here>

Options:

- 0 — —
- 1 — AK\_10
- 2 — AK\_2
- 3 — AK\_3
- 4 — AK\_4
- 5 — AK\_5
- 6 — AK\_6
- 7 — AK\_7
- 8 — AK\_8

- 9 — AK\_9
- 10 — AL\_E
- 11 — AL\_W
- 12 — AR\_N
- 13 — AR\_S
- 14 — AZ\_C
- 15 — AZ\_E
- 16 — AZ\_W
- 17 — CA\_I
- 18 — CA\_II
- 19 — CA\_III
- 20 — CA\_IV
- 21 — CA\_V
- 22 — CA\_VI
- 23 — CA\_VII
- 24 — CO\_C
- 25 — CO\_N
- 26 — CO\_S
- 27 — CT
- 28 — DE
- 29 — FL\_E
- 30 — FL\_N
- 31 — FL\_W
- 32 — GA\_E
- 33 — GA\_W
- 34 — HI\_1
- 35 — HI\_2
- 36 — HI\_3
- 37 — HI\_4
- 38 — HI\_5
- 39 — IA\_N
- 40 — IA\_S
- 41 — ID\_C
- 42 — ID\_E
- 43 — ID\_W
- 44 — IL\_E
- 45 — IL\_W
- 46 — IN\_E
- 47 — IN\_W

- 48 — KS\_N
- 49 — KS\_S
- 50 — KY\_N
- 51 — KY\_S
- 52 — LA\_N
- 53 — LA\_S
- 54 — MA\_I
- 55 — MA\_M
- 56 — MD
- 57 — ME\_E
- 58 — ME\_W
- 59 — MI\_C
- 60 — MI\_N
- 61 — MI\_S
- 62 — MN\_C
- 63 — MN\_N
- 64 — MN\_S
- 65 — MO\_C
- 66 — MO\_E
- 67 — MO\_W
- 68 — MS\_E
- 69 — MS\_W
- 70 — MT\_C
- 71 — MT\_N
- 72 — MT\_S
- 73 — NC
- 74 — ND\_N
- 75 — ND\_S
- 76 — NE\_N
- 77 — NE\_S
- 78 — NH
- 79 — NJ
- 80 — NM\_C
- 81 — NM\_E
- 82 — NM\_W
- 83 — NV\_C
- 84 — NV\_E
- 85 — NV\_W
- 86 — NY\_C

- 87 — NY\_E
- 88 — NY\_LI
- 89 — NY\_W
- 90 — OH\_N
- 91 — OH\_S
- 92 — OK\_N
- 93 — OK\_S
- 94 — OR\_N
- 95 — OR\_S
- 96 — PA\_N
- 97 — PA\_S
- 98 — PR
- 99 — RI
- 100 — SC\_N
- 101 — SC\_S
- 102 — SD\_N
- 103 — SD\_S
- 104 — St.Croix
- 105 — TN
- 106 — TX\_C
- 107 — TX\_N
- 108 — TX\_NC
- 109 — TX\_S
- 110 — TX\_SC
- 111 — UT\_C
- 112 — UT\_N
- 113 — UT\_S
- 114 — VA\_N
- 115 — VA\_S
- 116 — VT
- 117 — WA\_N
- 118 — WA\_S
- 119 — WI\_C
- 120 — WI\_N
- 121 — WI\_S
- 122 — WV\_N
- 123 — WV\_S
- 124 — WY\_E
- 125 — WY\_EC

- 126 — WY\_W
- 127 — WY\_WC

Default: 0

## Outputs

**output LAS/LAZ file [file]** <put output description here>

## Console usage

```
processing.runalg('lidartools:las2lasproject', verbose, input_laslaz, source_projection, source_u
```

## See also

### 18.2.3 las2las\_transform

#### Description

<put algorithm description here>

#### Parameters

**verbose [boolean]** <put parameter description here>

Default: *False*

**input LAS/LAZ file [file]** Optional.

<put parameter description here>

**transform (coordinates) [selection]** <put parameter description here>

Options:

- 0 — —
- 1 — translate\_x
- 2 — translate\_y
- 3 — translate\_z
- 4 — scale\_x
- 5 — scale\_y
- 6 — scale\_z
- 7 — clamp\_z\_above
- 8 — clamp\_z\_below

Default: 0

**value for transform (coordinates) [string]** <put parameter description here>

Default: *(not set)*

**second transform (coordinates) [selection]** <put parameter description here>

Options:

- 0 — —

- 1 — translate\_x
- 2 — translate\_y
- 3 — translate\_z
- 4 — scale\_x
- 5 — scale\_y
- 6 — scale\_z
- 7 — clamp\_z\_above
- 8 — clamp\_z\_below

Default: 0

**value for second transform (coordinates) [string]** <put parameter description here>

Default: *(not set)*

**transform (intensities, scan angles, GPS times, ...)** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — scale\_intensity
- 2 — translate\_intensity
- 3 — clamp\_intensity\_above
- 4 — clamp\_intensity\_below
- 5 — scale\_scan\_angle
- 6 — translate\_scan\_angle
- 7 — translate\_gps\_time
- 8 — set\_classification
- 9 — set\_user\_data
- 10 — set\_point\_source
- 11 — scale\_rgb\_up
- 12 — scale\_rgb\_down
- 13 — repair\_zero\_returns

Default: 0

**value for transform (intensities, scan angles, GPS times, ...)** [string] <put parameter description here>

Default: *(not set)*

**second transform (intensities, scan angles, GPS times, ...)** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — scale\_intensity
- 2 — translate\_intensity
- 3 — clamp\_intensity\_above
- 4 — clamp\_intensity\_below



- 5 — scale\_scan\_angle
- 6 — translate\_scan\_angle
- 7 — translate\_gps\_time
- 8 — set\_classification
- 9 — set\_user\_data
- 10 — set\_point\_source
- 11 — scale\_rgb\_up
- 12 — scale\_rgb\_down
- 13 — repair\_zero\_returns

Default: 0

**value for second transform (intensities, scan angles, GPS times, ...)** [string]  
 <put parameter description here>

Default: (not set)

**operations (first 7 need an argument)** [selection] <put parameter description here>

Options:

- 0 — —
- 1 — set\_point\_type
- 2 — set\_point\_size
- 3 — set\_version\_minor
- 4 — set\_version\_major
- 5 — start\_at\_point
- 6 — stop\_at\_point
- 7 — remove\_vlr
- 8 — auto\_reoffset
- 9 — week\_to\_adjusted
- 10 — adjusted\_to\_week
- 11 — scale\_rgb\_up
- 12 — scale\_rgb\_down
- 13 — remove\_all\_vlrs
- 14 — remove\_extra
- 15 — clip\_to\_bounding\_box

Default: 0

**argument for operation** [string] <put parameter description here>

Default: (not set)

## Outputs

**output LAS/LAZ file** [file] <put output description here>

## Console usage

```
processing.runalg('lidartools:las2lastransform', verbose, input_laslaz, transform_coordinate1, tr
```

## See also

### 18.2.4 las2txt

#### Description

<put algorithm description here>

#### Parameters

**verbose** [boolean] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [file] Optional.

<put parameter description here>

**parse\_string** [string] <put parameter description here>

Default: *xyz*

#### Outputs

**Output ASCII file** [file] <put output description here>

## Console usage

```
processing.runalg('lidartools:las2txt', verbose, input_laslaz, parse_string, output)
```

## See also

### 18.2.5 lasindex

#### Description

<put algorithm description here>

#### Parameters

**verbose** [boolean] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [file] Optional.

<put parameter description here>

**is mobile or terrestrial LiDAR (not airborne)** [boolean] <put parameter description here>

Default: *False*

## Outputs

### Console usage

```
processing.runalg('lidartools:lasindex', verbose, input_laslaz, mobile_or_terrestrial)
```

### See also

## 18.2.6 lasinfo

### Description

<put algorithm description here>

### Parameters

**verbose** [boolean] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [file] Optional.

<put parameter description here>

### Outputs

**Output ASCII file** [file] <put output description here>

### Console usage

```
processing.runalg('lidartools:lasinfo', verbose, input_laslaz, output)
```

### See also

## 18.2.7 lasmerge

### Description

<put algorithm description here>

### Parameters

**verbose** [boolean] <put parameter description here>

Default: *False*

**files are flightlines** [boolean] <put parameter description here>

Default: *True*

**input LAS/LAZ file** [file] Optional.

<put parameter description here>

**2nd file** [file] Optional.

<put parameter description here>

**3rd file [file]** Optional.

<put parameter description here>

**4th file [file]** Optional.

<put parameter description here>

**5th file [file]** Optional.

<put parameter description here>

**6th file [file]** Optional.

<put parameter description here>

**7th file [file]** Optional.

<put parameter description here>

## Outputs

**output LAS/LAZ file [file]** <put output description here>

## Console usage

```
processing.runalg('lidartools:lasmerge', verbose, files_are_flightlines, input_laslaz, file2, file3)
```

## See also

### 18.2.8 lasprecision

#### Description

<put algorithm description here>

#### Parameters

**verbose [boolean]** <put parameter description here>

Default: *False*

**input LAS/LAZ file [file]** Optional.

<put parameter description here>

#### Outputs

**Output ASCII file [file]** <put output description here>

#### Console usage

```
processing.runalg('lidartools:lasprecision', verbose, input_laslaz, output)
```

See also

## 18.2.9 lasquery

### Description

<put algorithm description here>

### Parameters

**verbose** [**boolean**] <put parameter description here>

Default: *False*

**area of interest** [**extent**] <put parameter description here>

Default: *0,1,0,1*

### Outputs

### Console usage

```
processing.runalg('lidartools:lasquery', verbose, aoi)
```

See also

## 18.2.10 lasvalidate

### Description

<put algorithm description here>

### Parameters

**verbose** [**boolean**] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [**file**] Optional.

<put parameter description here>

### Outputs

**Output XML file** [**file**] <put output description here>

### Console usage

```
processing.runalg('lidartools:lasvalidate', verbose, input_laslaz, output)
```

See also

## 18.2.11 laszip

### Description

<put algorithm description here>

### Parameters

**verbose** [**boolean**] <put parameter description here>

Default: *False*

**input LAS/LAZ file** [**file**] Optional.

<put parameter description here>

**only report size** [**boolean**] <put parameter description here>

Default: *False*

### Outputs

**output LAS/LAZ file** [**file**] <put output description here>

### Console usage

```
processing.runalg('lidartools:laszip', verbose, input_laslaz, report_size, output_laslaz)
```

See also

## 18.2.12 txt2las

### Description

<put algorithm description here>

### Parameters

**verbose** [**boolean**] <put parameter description here>

Default: *False*

**Input ASCII file** [**file**] Optional.

<put parameter description here>

**parse lines as** [**string**] <put parameter description here>

Default: *xyz*

**skip the first n lines** [**number**] <put parameter description here>

Default: *0*

**resolution of x and y coordinate** [**number**] <put parameter description here>

Default: *0.01*

**resolution of z coordinate [number]** <put parameter description here>

Default: *0.01*

## Outputs

**output LAS/LAZ file [file]** <put output description here>

## Console usage

```
processing.runalg('lidartools:txt2las', verbose, input, parse_string, skip, scale_factor_xy, scale_factor_z)
```

## See also

.

# 18.3 Modeler Tools

## 18.3.1 Calculator

### Description

<put algorithm description here>

### Parameters

**Formula [string]** <put parameter description here>

Default: *(not set)*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy [number]** <put parameter description here>

Default: *0.0*

**dummy** [number] <put parameter description here>

Default: 0.0

**dummy** [number] <put parameter description here>

Default: 0.0

## Outputs

**Result** [number] <put output description here>

## Console usage

```
processing.runalg('modelertools:calculator', formula, number0, number1, number2, number3, number4,
```

## See also

### 18.3.2 Raster layer bounds

#### Description

<put algorithm description here>

#### Parameters

**Layer** [raster] <put parameter description here>

#### Outputs

**min X** [number] <put output description here>

**max X** [number] <put output description here>

**min Y** [number] <put output description here>

**max Y** [number] <put output description here>

**Extent** [extent] <put output description here>

#### Console usage

```
processing.runalg('modelertools:rasterlayerbounds', layer)
```

## See also

### 18.3.3 Vector layer bounds

#### Description

<put algorithm description here>

#### Parameters

**Layer** [vector: any] <put parameter description here>



## Outputs

**min X [number]** <put output description here>

**max X [number]** <put output description here>

**min Y [number]** <put output description here>

**max Y [number]** <put output description here>

**Extent [extent]** <put output description here>

## Console usage

```
processing.runalg('modelertools:vectorlayerbounds', layer)
```

## See also

.

## 18.4 OrfeoToolbox algorithm provider

Orfeo ToolBox (OTB) is an open source library of image processing algorithms. OTB is based on the medical image processing library ITK and offers particular functionalities for remote sensing image processing in general and for high spatial resolution images in particular. Targeted algorithms for high resolution optical images (Pleiades, SPOT, QuickBird, WorldView, Landsat, Ikonos), hyperspectral sensors (Hyperion) or SAR (TerraSarX, ERS, Palsar) are available.

---

**Bemerkung:** Please remember that Processing contains only the interface description, so you need to install OTB by yourself and configure Processing properly.

---

.

### 18.4.1 Calibration

#### Optical calibration

##### Description

<put algorithm description here>

##### Parameters

**Input [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Calibration Level [selection]** <put parameter description here>

Options:

- 0 — toa

Default: *0*

**Convert to milli reflectance [boolean]** <put parameter description here>

Default: *True*

**Clamp of reflectivity values between [0, 100] [boolean]** <put parameter description here>

Default: *True*

**Relative Spectral Response File [file]** Optional.

<put parameter description here>

## Outputs

**Output [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:opticalcalibration', -in, -ram, -level, -milli, -clamp, -rsr, -out)
```

## See also

.

## 18.4.2 Feature extrcation

### BinaryMorphologicalOperation (closing)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation [selection]** <put parameter description here>

Options:

- 0 — closing

Default: *0*

## Outputs

**Feature Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:binarymorphologicaloperationclosing', -in, -channel, -ram, -structype, -st
```

## See also

## BinaryMorphologicalOperation (dilate)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation [selection]** <put parameter description here>

Options:

- 0 — dilate

Default: *0*

**Foreground Value [number]** <put parameter description here>

Default: *1*

**Background Value [number]** <put parameter description here>

Default: *0*

## Outputs

**Feature Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:binarymorphologicaloperationdilate', -in, -channel, -ram, -structype, -stru
```

### See also

### BinaryMorphologicalOperation (erode)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation [selection]** <put parameter description here>

Options:

- 0 — erode

Default: *0*

#### Outputs

**Feature Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:binarymorphologicaloperationerode', -in, -channel, -ram, -structype, -stru
```

### See also

### BinaryMorphologicalOperation (opening)

#### Description

<put algorithm description here>

## Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation [selection]** <put parameter description here>

Options:

- 0 — opening

Default: *0*

## Outputs

**Feature Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:binarymorphologicaloperationopening', -in, -channel, -ram, -structype, -st
```

## See also

### EdgeExtraction (gradient)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Edge feature [selection]** <put parameter description here>

Options:

- 0 — gradient

Default: 0

### Outputs

**Feature Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:edgeextractiongradient', -in, -channel, -ram, -filter, -out)
```

### See also

## EdgeExtraction (sobel)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: 1

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Edge feature [selection]** <put parameter description here>

Options:

- 0 — sobel

Default: 0

### Outputs

**Feature Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:edgeextraction sobel', -in, -channel, -ram, -filter, -out)
```

### See also

## EdgeExtraction (touzi)

### Description

<put algorithm description here>

## Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Edge feature [selection]** <put parameter description here>

Options:

- 0 — touzi

Default: *0*

**The Radius [number]** <put parameter description here>

Default: *1*

## Outputs

**Feature Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:edgeextractiontouzi', -in, -channel, -ram, -filter, -filter.touzi.xradius,
```

## See also

### GrayScaleMorphologicalOperation (closing)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation** [selection] <put parameter description here>

Options:

- 0 — closing

Default: 0

## Outputs

**Feature Output Image** [raster] <put output description here>

## Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationclosing', -in, -channel, -ram, -structype, ...)
```

## See also

## GrayScaleMorphologicalOperation (dilate)

### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Selected Channel** [number] <put parameter description here>

Default: 1

**Available RAM (Mb)** [number] <put parameter description here>

Default: 128

**Structuring Element Type** [selection] <put parameter description here>

Options:

- 0 — ball

Default: 0

**The Structuring Element Radius** [number] <put parameter description here>

Default: 5

**Morphological Operation** [selection] <put parameter description here>

Options:

- 0 — dilate

Default: 0

## Outputs

**Feature Output Image** [raster] <put output description here>



### Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationdilate', -in, -channel, -ram, -structype, -s
```

### See also

## GrayScaleMorphologicalOperation (erode)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation [selection]** <put parameter description here>

Options:

- 0 — erode

Default: *0*

### Outputs

**Feature Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationerode', -in, -channel, -ram, -structype, -s
```

### See also

## GrayScaleMorphologicalOperation (opening)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Structuring Element Type [selection]** <put parameter description here>

Options:

- 0 — ball

Default: *0*

**The Structuring Element Radius [number]** <put parameter description here>

Default: *5*

**Morphological Operation [selection]** <put parameter description here>

Options:

- 0 — opening

Default: *0*

### Outputs

**Feature Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationopening', -in, -channel, -ram, -structype, ...
```

### See also

### Haralick Texture Extraction

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Selected Channel [number]** <put parameter description here>

Default: *1*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**X Radius [number]** <put parameter description here>

Default: *2*

**Y Radius [number]** <put parameter description here>

Default: 2

**X Offset [number]** <put parameter description here>

Default: 1

**Y Offset [number]** <put parameter description here>

Default: 1

**Image Minimum [number]** <put parameter description here>

Default: 0

**Image Maximum [number]** <put parameter description here>

Default: 255

**Histogram number of bin [number]** <put parameter description here>

Default: 8

**Texture Set Selection [selection]** <put parameter description here>

Options:

- 0 — simple
- 1 — advanced
- 2 — higher

Default: 0

## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:haralicktextureextraction', -in, -channel, -ram, -parameters.xrad, -parameter
```

## See also

## Line segment detection

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**No rescaling in [0, 255] [boolean]** <put parameter description here>

Default: *True*

### Outputs

**Output** **Detected lines** [**vector**] <put output description here>

### Console usage

```
processing.runalg('otb:linesegmentdetection', -in, -norescale, -out)
```

### See also

## Local Statistic Extraction

### Description

<put algorithm description here>

### Parameters

**Input Image** [**raster**] <put parameter description here>

**Selected Channel** [**number**] <put parameter description here>

Default: 1

**Available RAM (Mb)** [**number**] <put parameter description here>

Default: 128

**Neighborhood radius** [**number**] <put parameter description here>

Default: 3

### Outputs

**Feature Output Image** [**raster**] <put output description here>

### Console usage

```
processing.runalg('otb:localstatisticextraction', -in, -channel, -ram, -radius, -out)
```

### See also

## Multivariate alteration detector

### Description

<put algorithm description here>

### Parameters

**Input Image 1** [raster] <put parameter description here>

**Input Image 2** [raster] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

### Outputs

**Change Map** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:multivariatealterationdetector', -in1, -in2, -ram, -out)
```

### See also

## Radiometric Indices

### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

**Blue Channel** [number] <put parameter description here>

Default: *1*

**Green Channel** [number] <put parameter description here>

Default: *1*

**Red Channel** [number] <put parameter description here>

Default: *1*

**NIR Channel** [number] <put parameter description here>

Default: *1*

**Mir Channel** [number] <put parameter description here>

Default: *1*

**Available Radiometric Indices** [selection] <put parameter description here>

Options:

- 0 — ndvi
- 1 — tndvi
- 2 — rvi

- 3 — savi
- 4 — tsavi
- 5 — msavi
- 6 — msavi2
- 7 — gemi
- 8 — ipvi
- 9 — ndwi
- 10 — ndwi2
- 11 — mndwi
- 12 — ndpi
- 13 — ndti
- 14 — ri
- 15 — ci
- 16 — bi
- 17 — bi2

Default: 0

## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:radiometricindices', -in, -ram, -channels.blue, -channels.green, -channels
```

## See also

.

## 18.4.3 Geometry

### Image Envelope

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Sampling Rate [number]** <put parameter description here>

Default: 0

**Projection** [string] Optional.

<put parameter description here>

Default: *None*

## Outputs

**Output Vector Data** [vector] <put output description here>

## Console usage

```
processing.runalg('otb:imageenvelope', -in, -sr, -proj, -out)
```

## See also

## OrthoRectification (epsg)

### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Output Cartographic Map Projection** [selection] <put parameter description here>

Options:

- 0 — epsg

Default: *0*

**EPSG Code** [number] <put parameter description here>

Default: *4326*

**Parameters estimation modes** [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: *0*

**Default pixel value** [number] <put parameter description here>

Default: *0*

**Default elevation** [number] <put parameter description here>

Default: *0*

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

**Radius for bicubic interpolation [number]** <put parameter description here>

Default: 2

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Resampling grid spacing [number]** <put parameter description here>

Default: 4

## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:orthorectificationeps', -io.in, -map, -map.epsg.code, -outputs.mode, -outp
```

## See also

### OrthoRectification (fit-to-ortho)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Parameters estimation modes [selection]** <put parameter description here>

Options:

- 0 — orthofit

Default: 0

**Model ortho-image [raster]** Optional.

<put parameter description here>

**Default pixel value [number]** <put parameter description here>

Default: 0

**Default elevation [number]** <put parameter description here>

Default: 0

**Interpolation [selection]** <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0



**Radius for bicubic interpolation [number]** <put parameter description here>

Default: 2

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Resampling grid spacing [number]** <put parameter description here>

Default: 4

## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:orthorectificationfittoortho', -io.in, -outputs.mode, -outputs.ortho, -outp
```

## See also

### OrthoRectification (lambert-WGS84)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Output Cartographic Map Projection [selection]** <put parameter description here>

Options:

- 0 — lambert2
- 1 — lambert93
- 2 — wgs

Default: 0

**Parameters estimation modes [selection]** <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: 0

**Default pixel value [number]** <put parameter description here>

Default: 0

**Default elevation [number]** <put parameter description here>

Default: 0

**Interpolation [selection]** <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

**Radius for bicubic interpolation [number]** <put parameter description here>

Default: 2

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Resampling grid spacing [number]** <put parameter description here>

Default: 4

## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:orthorectificationlambertwgs84', -io.in, -map, -outputs.mode, -outputs.def
```

## See also

## OrthoRectification (utm)

### Beschreibung

<put algorithm description here>

### Parameter

**Input Image [raster]** <put parameter description here>

**Output Cartographic Map Projection [selection]** <put parameter description here>

Optionen:

- 0 — utm

Vorgabe: 0

**Zonennummer [Nummer]** <put parameter description here>

Vorgabe: 31

**Northern Hemisphere [boolean]** <put parameter description here>

Default: *True*

**Parameters estimation modes [selection]** <put parameter description here>

Optionen:

- 0 — autosize
- 1 — autospacing

Vorgabe: 0

**Default pixel value [number]** <put parameter description here>

Vorgabe: 0

**Default elevation [number]** <put parameter description here>

Vorgabe: 0

**Interpolation [Auswahl]** <put parameter description here>

Optionen:

- 0 — bco
- 1 — nn
- 2 — linear

Vorgabe: 0

**Radius for bicubic interpolation [number]** <put parameter description here>

Vorgabe: 2

**Available RAM (Mb) [number]** <put parameter description here>

Vorgabe: 128

**Resampling grid spacing [number]** <put parameter description here>

Vorgabe: 4

## Ausgaben

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:orthorectificationutm', -io.in, -map, -map.utm.zone, -map.utm.northhem, -o
```

## Siehe auch

## Pansharpening (bayes)

### Description

<put algorithm description here>

### Parameters

**Input PAN Image [raster]** <put parameter description here>

**Input XS Image [raster]** <put parameter description here>

**Algorithm [selection]** <put parameter description here>

Options:

- 0 — bayes

Default: 0

**Weight [number]** <put parameter description here>

Default: 0.9999

**S coefficient** [number] <put parameter description here>

Default: *1*

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

### Outputs

**Output image** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:pansharpeningbayes', -inp, -inxs, -method, -method.bayes.lambda, -method.b
```

### See also

### Pansharpening (lmvm)

#### Description

<put algorithm description here>

#### Parameters

**Input PAN Image** [raster] <put parameter description here>

**Input XS Image** [raster] <put parameter description here>

**Algorithm** [selection] <put parameter description here>

Options:

- 0 — lmvm

Default: *0*

**X radius** [number] <put parameter description here>

Default: *3*

**Y radius** [number] <put parameter description here>

Default: *3*

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

### Outputs

**Output image** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:pansharpeninglmvm', -inp, -inxs, -method, -method.lmvm.radiusx, -method.lm
```

**See also**

**Pansharpening (rcs)**

**Description**

<put algorithm description here>

**Parameters**

**Input PAN Image [raster]** <put parameter description here>

**Input XS Image [raster]** <put parameter description here>

**Algorithm [selection]** <put parameter description here>

Options:

- 0 — rcs

Default: 0

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Outputs**

**Output image [raster]** <put output description here>

**Console usage**

```
processing.runalg('otb:pansharpeningrcs', -inp, -inxs, -method, -ram, -out)
```

**See also**

**RigidTransformResample (id)**

**Description**

<put algorithm description here>

**Parameters**

**Input image [raster]** <put parameter description here>

**Type of transformation [selection]** <put parameter description here>

Options:

- 0 — id

Default: 0

**X scaling [number]** <put parameter description here>

Default: 1

**Y scaling** [number] <put parameter description here>

Default: *1*

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: *2*

**Radius for bicubic interpolation** [number] <put parameter description here>

Default: *2*

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

## Outputs

**Output image** [raster] <put output description here>

## Console usage

```
processing.runalg('otb:rigidtransformresampleid', -in, -transform.type, -transform.type.id.scalex
```

## See also

### RigidTransformResample (rotation)

#### Description

<put algorithm description here>

#### Parameters

**Input image** [raster] <put parameter description here>

**Type of transformation** [selection] <put parameter description here>

Options:

- 0 — rotation

Default: *0*

**Rotation angle** [number] <put parameter description here>

Default: *0*

**X scaling** [number] <put parameter description here>

Default: *1*

**Y scaling** [number] <put parameter description here>

Default: *1*

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

**Radius for bicubic interpolation** [number] <put parameter description here>

Default: 2

**Available RAM (Mb)** [number] <put parameter description here>

Default: 128

## Outputs

**Output image** [raster] <put output description here>

## Console usage

```
processing.runalg('otb:rigidtransformresamplerotation', -in, -transform.type, -transform.type.rot
```

## See also

### RigidTransformResample (translation)

#### Description

<put algorithm description here>

#### Parameters

**Input image** [raster] <put parameter description here>

**Type of transformation** [selection] <put parameter description here>

Options:

- 0 — translation

Default: 0

**The X translation (in physical units)** [number] <put parameter description here>

Default: 0

**The Y translation (in physical units)** [number] <put parameter description here>

Default: 0

**X scaling** [number] <put parameter description here>

Default: 1

**Y scaling** [number] <put parameter description here>

Default: 1

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

**Radius for bicubic interpolation** [number] <put parameter description here>

Default: 2

**Available RAM (Mb)** [number] <put parameter description here>

Default: 128

## Outputs

**Output image** [raster] <put output description here>

## Console usage

```
processing.runalg('otb:rigidtransformresampletranslation', -in, -transform.type, -transform.type.
```

## See also

## Superimpose sensor

### Description

<put algorithm description here>

### Parameters

**Reference input** [raster] <put parameter description here>

**The image to reproject** [raster] <put parameter description here>

**Default elevation** [number] <put parameter description here>

Default: 0

**Spacing of the deformation field** [number] <put parameter description here>

Default: 4

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

**Radius for bicubic interpolation** [number] <put parameter description here>

Default: 2



**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

### Outputs

**Output image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:superimposesensor', -inr, -inm, -elev.default, -lms, -interpolator, -inter
```

### See also

.

## 18.4.4 Image filtering

### DimensionalityReduction (ica)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Algorithm [selection]** <put parameter description here>

Options:

- 0 — ica

Default: *0*

**number of iterations [number]** <put parameter description here>

Default: *20*

**Give the increment weight of W in [0, 1] [number]** <put parameter description here>

Default: *1*

**Number of Components [number]** <put parameter description here>

Default: *0*

**Normalize [boolean]** <put parameter description here>

Default: *True*

### Outputs

**Output Image [raster]** <put output description here>

**“Inverse Output Image“ [raster]** <put output description here>

**Transformation matrix output [file]** <put output description here>

### Console usage

```
processing.runalg('otb:dimensionalityreductionica', -in, -method, -method.ica.iter, -method.ica.m
```

### See also

### DimensionalityReduction (maf)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Algorithm [selection]** <put parameter description here>

Options:

- 0 — maf

Default: 0

**Number of Components. [number]** <put parameter description here>

Default: 0

**Normalize. [boolean]** <put parameter description here>

Default: *True*

#### Outputs

**Output Image [raster]** <put output description here>

**Transformation matrix output [file]** <put output description here>

### Console usage

```
processing.runalg('otb:dimensionalityreductionmaf', -in, -method, -nbcomp, -normalize, -out, -out
```

### See also

### DimensionalityReduction (napca)

#### Description

<put algorithm description here>

## Parameters

**Input Image** [raster] <put parameter description here>

**Algorithm** [selection] <put parameter description here>

Options:

- 0 — napca

Default: 0

**Set the x radius of the sliding window.** [number] <put parameter description here>

Default: 1

**Set the y radius of the sliding window.** [number] <put parameter description here>

Default: 1

**Number of Components.** [number] <put parameter description here>

Default: 0

**Normalize.** [boolean] <put parameter description here>

Default: True

## Outputs

**Output Image** [raster] <put output description here>

“**Inverse Output Image**“ [raster] <put output description here>

**Transformation matrix output** [file] <put output description here>

## Console usage

```
processing.runalg('otb:dimensionalityreductionnapca', -in, -method, -method.napca.radiusx, -method
```

## See also

### DimensionalityReduction (pca)

#### Description

<put algorithm description here>

#### Parameters

**Input Image** [raster] <put parameter description here>

**Algorithm** [selection] <put parameter description here>

Options:

- 0 — pca

Default: 0

**Number of Components.** [number] <put parameter description here>

Default: 0

**Normalize** . [boolean] <put parameter description here>

Default: *True*

### Outputs

**Output Image** [raster] <put output description here>

“ **Inverse Output Image**“ [raster] <put output description here>

**Transformation matrix output** [file] <put output description here>

### Console usage

```
processing.runalg('otb:dimensionalityreductionpca', -in, -method, -nbcomp, -normalize, -out, -out.
```

### See also

**Mean Shift filtering (can be used as Exact Large-Scale Mean-Shift segmentation, step 1)**

### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Spatial radius** [number] <put parameter description here>

Default: *5*

**Range radius** [number] <put parameter description here>

Default: *15*

**Mode convergence threshold** [number] <put parameter description here>

Default: *0.1*

**Maximum number of iterations** [number] <put parameter description here>

Default: *100*

**Range radius coefficient** [number] <put parameter description here>

Default: *0*

**Mode search** . [boolean] <put parameter description here>

Default: *True*

### Outputs

**Filtered output** [raster] <put output description here>

**Spatial image** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:meanshiftfilteringcanbeusedasexactlargescalemeanshiftsegmentationstep1', -
```

### See also

### Smoothing (anidif)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Smoothing Type [selection]** <put parameter description here>

Options:

- 0 — anidif

Default: *2*

**Time Step [number]** <put parameter description here>

Default: *0.125*

**Nb Iterations [number]** <put parameter description here>

Default: *10*

#### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:smoothinganidif', -in, -ram, -type, -type.anidif.timestep, -type.anidif.nb
```

### See also

### Smoothing (gaussian)

#### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Smoothing Type [selection]** <put parameter description here>

Options:

- 0 — gaussian

Default: 2

**Radius [number]** <put parameter description here>

Default: 2

### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:smoothinggaussian', -in, -ram, -type, -type.gaussian.radius, -out)
```

### See also

### Smoothing (mean)

#### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Smoothing Type [selection]** <put parameter description here>

Options:

- 0 — mean

Default: 2

**Radius [number]** <put parameter description here>

Default: 2

### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:smoothingmean', -in, -ram, -type, -type.mean.radius, -out)
```

### See also

.

## 18.4.5 Image manipulation

### ColorMapping (continuous)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Operation [selection]** <put parameter description here>

Options:

- 0 — labeltocolour

Default: *0*

**Color mapping method [selection]** <put parameter description here>

Options:

- 0 — continuous

Default: *0*

**Look-up tables [selection]** <put parameter description here>

Options:

- 0 — red
- 1 — green
- 2 — blue
- 3 — grey
- 4 — hot
- 5 — cool
- 6 — spring
- 7 — summer
- 8 — autumn
- 9 — winter
- 10 — copper

- 11 — jet
- 12 — hsv
- 13 — overunder
- 14 — relief

Default: 0

**Mapping range lower value [number]** <put parameter description here>

Default: 0

**Mapping range higher value [number]** <put parameter description here>

Default: 255

## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:colormappingcontinuous', -in, -ram, -op, -method, -method.continuous.lut, ...
```

## See also

## ColorMapping (custom)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Operation [selection]** <put parameter description here>

Options:

- 0 — labeltcolor

Default: 0

**Color mapping method [selection]** <put parameter description here>

Options:

- 0 — custom

Default: 0

**Look-up table file [file]** <put parameter description here>



## Outputs

**Output Image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:colormappingcustom', -in, -ram, -op, -method, -method.custom.lut, -out)
```

## See also

## ColorMapping (image)

### Beschreibung

<put algorithm description here>

### Parameter

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Vorgabe: 128

**Operation [selection]** <put parameter description here>

Optionen:

- 0 — labeltcolor

Vorgabe: 0

**Color mapping method [selection]** <put parameter description here>

Optionen:

- 0 — Bild

Vorgabe: 0

**Support Image [raster]** <put parameter description here>

**NoData value [number]** <put parameter description here>

Vorgabe: 0

**lower quantile [number]** <put parameter description here>

Vorgabe: 2

**upper quantile [number]** <put parameter description here>

Vorgabe: 2

### Ausgaben

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:colormappingimage', -in, -ram, -op, -method, -method.image.in, -method.ima
```

### Siehe auch

### ColorMapping (optimal)

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Operation [selection]** <put parameter description here>

Options:

- 0 — labeltocolor

Default: *0*

**Color mapping method [selection]** <put parameter description here>

Options:

- 0 — optimal

Default: *0*

**Background label [number]** <put parameter description here>

Default: *0*

#### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:colormappingoptimal', -in, -ram, -op, -method, -method.optimal.background,
```

### See also

### ExtractROI (fit)

#### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

**Extraction mode** [selection] <put parameter description here>

Options:

- 0 — fit

Default: *0*

**Reference image** [raster] <put parameter description here>

**Default elevation** [number] <put parameter description here>

Default: *0*

### Outputs

**Output Image** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:extractroi', -in, -ram, -mode, -mode.fit.ref, -mode.fit.elev.default, -
```

### See also

#### ExtractROI (standard)

#### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

**Extraction mode** [selection] <put parameter description here>

Options:

- 0 — standard

Default: *0*

**Start X** [number] <put parameter description here>

Default: *0*

**Start Y** [number] <put parameter description here>

Default: *0*

**Size X [number]** <put parameter description here>

Default: 0

**Size Y [number]** <put parameter description here>

Default: 0

### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:extractroistandard', -in, -ram, -mode, -startx, -starty, -sizex, -sizey, -o
```

### See also

## Images Concatenation

### Description

<put algorithm description here>

### Parameters

**Input images list [multipleinput: rasters]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:imagesconcatenation', -il, -ram, -out)
```

### See also

## Image Tile Fusion

### Description

<put algorithm description here>

### Parameters

**Input Tile Images** [**multipleinput: rasters**] <put parameter description here>

**Number of tile columns** [**number**] <put parameter description here>

Default: *0*

**Number of tile rows** [**number**] <put parameter description here>

Default: *0*

### Outputs

**Output Image** [**raster**] <put output description here>

### Console usage

```
processing.runalg('otb:imagetilefusion', -il, -cols, -rows, -out)
```

### See also

### Read image information

#### Description

<put algorithm description here>

#### Parameters

**Input Image** [**raster**] <put parameter description here>

**Display the OSSIM keywordlist** [**boolean**] <put parameter description here>

Default: *True*

**GCPs Id** [**string**] <put parameter description here>

Default: *None*

**GCPs Info** [**string**] <put parameter description here>

Default: *None*

**GCPs Image Coordinates** [**string**] <put parameter description here>

Default: *None*

**GCPs Geographic Coordinates** [**string**] <put parameter description here>

Default: *None*

#### Outputs

#### Console usage

```
processing.runalg('otb:readimageinformation', -in, -keywordlist, -gcp.ids, -gcp.info, -gcp.imcoord
```

See also

## Rescale Image

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Output min value [number]** <put parameter description here>

Default: 0

**Output max value [number]** <put parameter description here>

Default: 255

### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:rescaleimage', -in, -ram, -outmin, -outmax, -out)
```

See also

## Split Image

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

### Outputs

**Output Image [file]** <put output description here>

**Console usage**

```
processing.runalg('otb:splitimage', -in, -ram, -out)
```

**See also**

.

## 18.4.6 Learning

### Classification Map Regularization

**Description**

<put algorithm description here>

**Parameters**

**Input classification image [raster]** <put parameter description here>

**Structuring element radius (in pixels) [number]** <put parameter description here>

Default: *1*

**Multiple majority: Undecided(X)/Original [boolean]** <put parameter description here>

Default: *True*

**Label for the NoData class [number]** <put parameter description here>

Default: *0*

**Label for the Undecided class [number]** <put parameter description here>

Default: *0*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

**Outputs**

**Output regularized image [raster]** <put output description here>

**Console usage**

```
processing.runalg('otb:classificationmapregularization', -io.in, -ip.radius, -ip.suvbool, -ip.nod
```

**See also**

**ComputeConfusionMatrix (raster)**

**Description**

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Ground truth [selection]** <put parameter description here>

Options:

- 0 — raster

Default: 0

**Input reference image [raster]** <put parameter description here>

**Value for nodata pixels [number]** <put parameter description here>

Default: 0

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

### Outputs

**Matrix output [file]** <put output description here>

### Console usage

```
processing.runalg('otb:computeconfusionmatrixraster', -in, -ref, -ref.raster.in, -nodatalabel, -r
```

### See also

#### ComputeConfusionMatrix (vector)

#### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Ground truth [selection]** <put parameter description here>

Options:

- 0 — vector

Default: 0

**Input reference vector data [file]** <put parameter description here>

**Field name [string]** Optional.

<put parameter description here>

Default: *Class*

**Value for nodata pixels [number]** <put parameter description here>

Default: 0



**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

### Outputs

**Matrix output [file]** <put output description here>

### Console usage

```
processing.runalg('otb:computeconfusionmatrixvector', -in, -ref, -ref.vector.in, -ref.vector.field)
```

### See also

## Compute Images second order statistics

### Description

<put algorithm description here>

### Parameters

**Input images [multipleinput: rasters]** <put parameter description here>

**Background Value [number]** <put parameter description here>

Default: *0.0*

### Outputs

**Output XML file [file]** <put output description here>

### Console usage

```
processing.runalg('otb:computeimagessecondorderstatistics', -il, -bv, -out)
```

### See also

## FusionOfClassifications (dempstershafer)

### Description

<put algorithm description here>

### Parameters

**Input classifications [multipleinput: rasters]** <put parameter description here>

**Fusion method [selection]** <put parameter description here>

Options:

- 0 — dempstershafer

Default: 0

**Confusion Matrices** [**multipleinput: files**] <put parameter description here>

**Mass of belief measurement** [**selection**] <put parameter description here>

Options:

- 0 — precision
- 1 — recall
- 2 — accuracy
- 3 — kappa

Default: 0

**Label for the NoData class** [**number**] <put parameter description here>

Default: 0

**Label for the Undecided class** [**number**] <put parameter description here>

Default: 0

## Outputs

**The output classification image** [**raster**] <put output description here>

## Console usage

```
processing.runalg('otb:fusionofclassificationsdempstershafer', -il, -method, -method.dempstershafer)
```

## See also

### FusionOfClassifications (majorityvoting)

#### Description

<put algorithm description here>

#### Parameters

**Input classifications** [**multipleinput: rasters**] <put parameter description here>

**Fusion method** [**selection**] <put parameter description here>

Options:

- 0 — majorityvoting

Default: 0

**Label for the NoData class** [**number**] <put parameter description here>

Default: 0

**Label for the Undecided class** [**number**] <put parameter description here>

Default: 0

## Outputs

**The output classification image [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:fusionofclassificationsmajorityvoting', -il, -method, -nodatalabel, -undec
```

## See also

## Image Classification

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Input Mask [raster]** Optional.

<put parameter description here>

**Model file [file]** <put parameter description here>

**Statistics file [file]** Optional.

<put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

### Outputs

**Output Image [raster]** <put output description here>

### Console usage

```
processing.runalg('otb:imageclassification', -in, -mask, -model, -imstat, -ram, -out)
```

## See also

## SOM Classification

### Description

<put algorithm description here>

## Parameters

**InputImage** [raster] <put parameter description here>

**ValidityMask** [raster] Optional.

<put parameter description here>

**TrainingProbability** [number] <put parameter description here>

Default: 1

**TrainingSetSize** [number] <put parameter description here>

Default: 0

**StreamingLines** [number] <put parameter description here>

Default: 0

**SizeX** [number] <put parameter description here>

Default: 32

**SizeY** [number] <put parameter description here>

Default: 32

**NeighborhoodX** [number] <put parameter description here>

Default: 10

**NeighborhoodY** [number] <put parameter description here>

Default: 10

**NumberIteration** [number] <put parameter description here>

Default: 5

**BetaInit** [number] <put parameter description here>

Default: 1

**BetaFinal** [number] <put parameter description here>

Default: 0.1

**InitialValue** [number] <put parameter description here>

Default: 0

**Available RAM (Mb)** [number] <put parameter description here>

Default: 128

**set user defined seed** [number] <put parameter description here>

Default: 0

## Outputs

**OutputImage** [raster] <put output description here>

**SOM Map** [raster] <put output description here>

## Console usage

```
processing.runalg('otb:somclassification', -in, -vm, -tp, -ts, -sl, -sx, -sy, -nx, -ny, -ni, -bi,
```

See also

**TrainImagesClassifier (ann)**

**Description**

<put algorithm description here>

**Parameters**

**Input Image List** [**multipleinput: rasters**] <put parameter description here>

**Input Vector Data List** [**multipleinput: any vectors**] <put parameter description here>

**Input XML image statistics file** [**file**] Optional.

<put parameter description here>

**Default elevation** [**number**] <put parameter description here>

Default: *0*

**Maximum training sample size per class** [**number**] <put parameter description here>

Default: *1000*

**Maximum validation sample size per class** [**number**] <put parameter description here>

Default: *1000*

**On edge pixel inclusion** [**boolean**] <put parameter description here>

Default: *True*

**Training and validation sample ratio** [**number**] <put parameter description here>

Default: *0.5*

**Name of the discrimination field** [**string**] <put parameter description here>

Default: *Class*

**Classifier to use for the training** [**selection**] <put parameter description here>

Options:

- 0 — ann

Default: *0*

**Train Method Type** [**selection**] <put parameter description here>

Options:

- 0 — reg
- 1 — back

Default: *0*

**Number of neurons in each intermediate layer** [**string**] <put parameter description here>

Default: *None*

**Neuron activation function type** [**selection**] <put parameter description here>

Options:

- 0 — ident
- 1 — sig

- 2 — gau

Default: 1

**Alpha parameter of the activation function [number]** <put parameter description here>

Default: 1

**Beta parameter of the activation function [number]** <put parameter description here>

Default: 1

**Strength of the weight gradient term in the BACKPROP method [number]** <put parameter description here>

Default: 0.1

**Strength of the momentum term (the difference between weights on the 2 previous iterations) [number]** <put parameter description here>

Default: 0.1

**Initial value Delta\_0 of update-values Delta\_{ij} in RPROP method [number]** <put parameter description here>

Default: 0.1

**Update-values lower limit Delta\_{min} in RPROP method [number]** <put parameter description here>

Default: 1e-07

**Termination criteria [selection]** <put parameter description here>

Options:

- 0 — iter
- 1 — eps
- 2 — all

Default: 2

**Epsilon value used in the Termination criteria [number]** <put parameter description here>

Default: 0.01

**Maximum number of iterations used in the Termination criteria [number]** <put parameter description here>

Default: 1000

**set user defined seed [number]** <put parameter description here>

Default: 0

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifierann', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

**See also**

**TrainImagesClassifier (bayes)**

**Description**

<put algorithm description here>

**Parameters**

**Input Image List** [**multipleinput: rasters**] <put parameter description here>

**Input Vector Data List** [**multipleinput: any vectors**] <put parameter description here>

**Input XML image statistics file** [**file**] Optional.

<put parameter description here>

**Default elevation** [**number**] <put parameter description here>

Default: 0

**Maximum training sample size per class** [**number**] <put parameter description here>

Default: 1000

**Maximum validation sample size per class** [**number**] <put parameter description here>

Default: 1000

**On edge pixel inclusion** [**boolean**] <put parameter description here>

Default: True

**Training and validation sample ratio** [**number**] <put parameter description here>

Default: 0.5

**Name of the discrimination field** [**string**] <put parameter description here>

Default: Class

**Classifier to use for the training** [**selection**] <put parameter description here>

Options:

- 0 — bayes

Default: 0

**set user defined seed** [**number**] <put parameter description here>

Default: 0

**Outputs**

**Output confusion matrix** [**file**] <put output description here>

**Output model** [**file**] <put output description here>

**Console usage**

```
processing.runalg('otb:trainimagesclassifierbayes', -io.il, -io.vd, -io.imstat, -elev.default, -s
```

See also

## TrainImagesClassifier (boost)

### Description

<put algorithm description here>

### Parameters

**Input Image List** [**multipleinput: rasters**] <put parameter description here>

**Input Vector Data List** [**multipleinput: any vectors**] <put parameter description here>

**Input XML image statistics file** [**file**] Optional.

<put parameter description here>

**Default elevation** [**number**] <put parameter description here>

Default: 0

**Maximum training sample size per class** [**number**] <put parameter description here>

Default: 1000

**Maximum validation sample size per class** [**number**] <put parameter description here>

Default: 1000

**On edge pixel inclusion** [**boolean**] <put parameter description here>

Default: True

**Training and validation sample ratio** [**number**] <put parameter description here>

Default: 0.5

**Name of the discrimination field** [**string**] <put parameter description here>

Default: Class

**Classifier to use for the training** [**selection**] <put parameter description here>

Options:

- 0 — boost

Default: 0

**Boost Type** [**selection**] <put parameter description here>

Options:

- 0 — discrete
- 1 — real
- 2 — logit
- 3 — gentle

Default: 1

**Weak count** [**number**] <put parameter description here>

Default: 100

**Weight Trim Rate** [**number**] <put parameter description here>

Default: 0.95



**Maximum depth of the tree [number]** <put parameter description here>

Default: *1*

**set user defined seed [number]** <put parameter description here>

Default: *0*

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifierboost', -io.il, -io.vd, -io.imstat, -elev.default, -s
```

## See also

### TrainImagesClassifier (dt)

#### Description

<put algorithm description here>

#### Parameters

**Input Image List [multipleinput: rasters]** <put parameter description here>

**Input Vector Data List [multipleinput: any vectors]** <put parameter description here>

**Input XML image statistics file [file]** Optional.

<put parameter description here>

**Default elevation [number]** <put parameter description here>

Default: *0*

**Maximum training sample size per class [number]** <put parameter description here>

Default: *1000*

**Maximum validation sample size per class [number]** <put parameter description here>

Default: *1000*

**On edge pixel inclusion [boolean]** <put parameter description here>

Default: *True*

**Training and validation sample ratio [number]** <put parameter description here>

Default: *0.5*

**Name of the discrimination field [string]** <put parameter description here>

Default: *Class*

**Classifier to use for the training [selection]** <put parameter description here>

Options:

- 0 — dt

Default: 0

**Maximum depth of the tree [number]** <put parameter description here>

Default: 65535

**Minimum number of samples in each node [number]** <put parameter description here>

Default: 10

**Termination criteria for regression tree [number]** <put parameter description here>

Default: 0.01

**Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal**  
<put parameter description here>

Default: 10

**K-fold cross-validations [number]** <put parameter description here>

Default: 10

**Set UseSelfRule flag to false [boolean]** <put parameter description here>

Default: True

**Set TruncatePrunedTree flag to false [boolean]** <put parameter description here>

Default: True

**set user defined seed [number]** <put parameter description here>

Default: 0

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifierdt', -io.il, -io.vd, -io.imstat, -elev.default, -samp
```

## See also

### TrainImagesClassifier (gbt)

#### Description

<put algorithm description here>

#### Parameters

**Input Image List [multipleinput: rasters]** <put parameter description here>

**Input Vector Data List [multipleinput: any vectors]** <put parameter description here>

**Input XML image statistics file [file]** Optional.

<put parameter description here>

**Default elevation [number]** <put parameter description here>

Default: *0*

**Maximum training sample size per class [number]** <put parameter description here>

Default: *1000*

**Maximum validation sample size per class [number]** <put parameter description here>

Default: *1000*

**On edge pixel inclusion [boolean]** <put parameter description here>

Default: *True*

**Training and validation sample ratio [number]** <put parameter description here>

Default: *0.5*

**Name of the discrimination field [string]** <put parameter description here>

Default: *Class*

**Classifier to use for the training [selection]** <put parameter description here>

Options:

- 0 — gbt

Default: *0*

**Number of boosting algorithm iterations [number]** <put parameter description here>

Default: *200*

**Regularization parameter [number]** <put parameter description here>

Default: *0.01*

**Portion of the whole training set used for each algorithm iteration [number]**

<put parameter description here>

Default: *0.8*

**Maximum depth of the tree [number]** <put parameter description here>

Default: *3*

**set user defined seed [number]** <put parameter description here>

Default: *0*

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifiergbt', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

## TrainImagesClassifier (knn)

### Description

<put algorithm description here>

### Parameters

**Input Image List** [**multipleinput: rasters**] <put parameter description here>

**Input Vector Data List** [**multipleinput: any vectors**] <put parameter description here>

**Input XML image statistics file** [**file**] Optional.

<put parameter description here>

**Default elevation** [**number**] <put parameter description here>

Default: 0

**Maximum training sample size per class** [**number**] <put parameter description here>

Default: 1000

**Maximum validation sample size per class** [**number**] <put parameter description here>

Default: 1000

**On edge pixel inclusion** [**boolean**] <put parameter description here>

Default: *True*

**Training and validation sample ratio** [**number**] <put parameter description here>

Default: 0.5

**Name of the discrimination field** [**string**] <put parameter description here>

Default: *Class*

**Classifier to use for the training** [**selection**] <put parameter description here>

Options:

- 0 — knn

Default: 0

**Number of Neighbors** [**number**] <put parameter description here>

Default: 32

**set user defined seed** [**number**] <put parameter description here>

Default: 0

### Outputs

**Output confusion matrix** [**file**] <put output description here>

**Output model** [**file**] <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifierknn', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

## See also

### TrainImagesClassifier (libsvm)

#### Description

<put algorithm description here>

#### Parameters

**Input Image List** [**multipleinput: rasters**] <put parameter description here>

**Input Vector Data List** [**multipleinput: any vectors**] <put parameter description here>

**Input XML image statistics file** [**file**] Optional.

<put parameter description here>

**Default elevation** [**number**] <put parameter description here>

Default: *0*

**Maximum training sample size per class** [**number**] <put parameter description here>

Default: *1000*

**Maximum validation sample size per class** [**number**] <put parameter description here>

Default: *1000*

**On edge pixel inclusion** [**boolean**] <put parameter description here>

Default: *True*

**Training and validation sample ratio** [**number**] <put parameter description here>

Default: *0.5*

**Name of the discrimination field** [**string**] <put parameter description here>

Default: *Class*

**Classifier to use for the training** [**selection**] <put parameter description here>

Options:

- 0 — libsvm

Default: *0*

**SVM Kernel Type** [**selection**] <put parameter description here>

Options:

- 0 — linear
- 1 — rbf
- 2 — poly
- 3 — sigmoid

Default: *0*

**Cost parameter C [number]** <put parameter description here>

Default: *1*

**Parameters optimization [boolean]** <put parameter description here>

Default: *True*

**set user defined seed [number]** <put parameter description here>

Default: *0*

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifierlibsvm', -io.il, -io.vd, -io.imstat, -elev.default, -
```

## See also

### TrainImagesClassifier (rf)

#### Description

<put algorithm description here>

#### Parameters

**Input Image List [multipleinput: rasters]** <put parameter description here>

**Input Vector Data List [multipleinput: any vectors]** <put parameter description here>

**Input XML image statistics file [file]** Optional.

<put parameter description here>

**Default elevation [number]** <put parameter description here>

Default: *0*

**Maximum training sample size per class [number]** <put parameter description here>

Default: *1000*

**Maximum validation sample size per class [number]** <put parameter description here>

Default: *1000*

**On edge pixel inclusion [boolean]** <put parameter description here>

Default: *True*

**Training and validation sample ratio [number]** <put parameter description here>

Default: *0.5*

**Name of the discrimination field [string]** <put parameter description here>

Default: *Class*

**Classifier to use for the training [selection]** <put parameter description here>

Options:

- 0 — rf

Default: 0

**Maximum depth of the tree [number]** <put parameter description here>

Default: 5

**Minimum number of samples in each node [number]** <put parameter description here>

Default: 10

**Termination Criteria for regression tree [number]** <put parameter description here>

Default: 0

**Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal** <put parameter description here>

Default: 10

**Size of the randomly selected subset of features at each tree node [number]** <put parameter description here>

Default: 0

**Maximum number of trees in the forest [number]** <put parameter description here>

Default: 100

**Sufficient accuracy (OOB error) [number]** <put parameter description here>

Default: 0.01

**set user defined seed [number]** <put parameter description here>

Default: 0

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifierrf', -io.il, -io.vd, -io.imstat, -elev.default, -samp
```

## See also

### TrainImagesClassifier (svm)

#### Description

<put algorithm description here>

## Parameters

**Input Image List** [**multipleinput: rasters**] <put parameter description here>

**Input Vector Data List** [**multipleinput: any vectors**] <put parameter description here>

**Input XML image statistics file** [**file**] Optional.

<put parameter description here>

**Default elevation** [**number**] <put parameter description here>

Default: *0*

**Maximum training sample size per class** [**number**] <put parameter description here>

Default: *1000*

**Maximum validation sample size per class** [**number**] <put parameter description here>

Default: *1000*

**On edge pixel inclusion** [**boolean**] <put parameter description here>

Default: *True*

**Training and validation sample ratio** [**number**] <put parameter description here>

Default: *0.5*

**Name of the discrimination field** [**string**] <put parameter description here>

Default: *Class*

**Classifier to use for the training** [**selection**] <put parameter description here>

Options:

- 0 — svm

Default: *0*

**SVM Model Type** [**selection**] <put parameter description here>

Options:

- 0 — csvc
- 1 — nusvc
- 2 — oneclass

Default: *0*

**SVM Kernel Type** [**selection**] <put parameter description here>

Options:

- 0 — linear
- 1 — rbf
- 2 — poly
- 3 — sigmoid

Default: *0*

**Cost parameter C** [**number**] <put parameter description here>

Default: *1*

**Parameter nu of a SVM optimization problem (NU\_SVC / ONE\_CLASS)** [**number**] <put parameter description here>

Default: *0*



**Parameter coef0 of a kernel function (POLY / SIGMOID) [number]** <put parameter description here>

Default: 0

**Parameter gamma of a kernel function (POLY / RBF / SIGMOID) [number]** <put parameter description here>

Default: 1

**Parameter degree of a kernel function (POLY) [number]** <put parameter description here>

Default: 1

**Parameters optimization [boolean]** <put parameter description here>

Default: *True*

**set user defined seed [number]** <put parameter description here>

Default: 0

## Outputs

**Output confusion matrix [file]** <put output description here>

**Output model [file]** <put output description here>

## Console usage

```
processing.runalg('otb:trainimagesclassifiersvm', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

## See also

## Unsupervised KMeans image classification

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Available RAM (Mb) [number]** <put parameter description here>

Default: 128

**Validity Mask [raster]** Optional.

<put parameter description here>

**Training set size [number]** <put parameter description here>

Default: 100

**Number of classes [number]** <put parameter description here>

Default: 5

**Maximum number of iterations [number]** <put parameter description here>

Default: 1000

**Convergence threshold** [number] <put parameter description here>

Default: *0.0001*

#### Outputs

**Output Image** [raster] <put output description here>

**Centroid filename** [file] <put output description here>

#### Console usage

```
processing.runalg('otb:unsupervisedkmeansimageclassification', -in, -ram, -vm, -ts, -nc, -maxit,
```

#### See also

.

## 18.4.7 Miscellaneous

### Band Math

#### Description

<put algorithm description here>

#### Parameters

**Input image list** [multipleinput: rasters] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: *128*

**Expression** [string] <put parameter description here>

Default: *None*

#### Outputs

**Output Image** [raster] <put output description here>

#### Console usage

```
processing.runalg('otb:bandmath', -il, -ram, -exp, -out)
```

#### See also

### ComputeModulusAndPhase-one (OneEntry)

#### Description

<put algorithm description here>

### Parameters

**Number Of inputs** [selection] <put parameter description here>

Options:

- 0 — one

Default: 0

**Input image** [raster] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: 128

### Outputs

**Modulus** [raster] <put output description here>

**Phase** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:computemodulusandphaseoneoneentry', -nbininput, -nbininput.one.in, -ram, -mod,
```

### See also

#### ComputeModulusAndPhase-two (TwoEntries)

### Description

<put algorithm description here>

### Parameters

**Number Of inputs** [selection] <put parameter description here>

Options:

- 0 — two

Default: 0

**Real part input** [raster] <put parameter description here>

**Imaginary part input** [raster] <put parameter description here>

**Available RAM (Mb)** [number] <put parameter description here>

Default: 128

### Outputs

**Modulus** [raster] <put output description here>

**Phase** [raster] <put output description here>

### Console usage

```
processing.runalg('otb:computemodulusandphasetwoentries', -nbinput, -nbinput.two.re, -nbinput.t
```

### See also

### Images comparaison

#### Description

<put algortithm description here>

#### Parameters

**Reference image [raster]** <put parameter description here>

**Reference image channel [number]** <put parameter description here>

Default: *1*

**Measured image [raster]** <put parameter description here>

**Measured image channel [number]** <put parameter description here>

Default: *1*

**Start X [number]** <put parameter description here>

Default: *0*

**Start Y [number]** <put parameter description here>

Default: *0*

**Size X [number]** <put parameter description here>

Default: *0*

**Size Y [number]** <put parameter description here>

Default: *0*

#### Outputs

### Console usage

```
processing.runalg('otb:imagescomparaison', -ref.in, -ref.channel, -meas.in, -meas.channel, -roi.st
```

### See also

### Image to KMZ Export

#### Description

<put algortithm description here>

## Parameters

**Input image [raster]** <put parameter description here>

**Tile Size [number]** <put parameter description here>

Default: *512*

**Image logo [raster]** Optional.

<put parameter description here>

**Image legend [raster]** Optional.

<put parameter description here>

**Default elevation [number]** <put parameter description here>

Default: *0*

## Outputs

**Output .kmz product [file]** <put output description here>

## Console usage

```
processing.runalg('otb:imagetokmzexport', -in, -tilesize, -logo, -legend, -elev.default, -out)
```

## See also

.

## 18.4.8 Segmentation

### Connected Component Segmentation

#### Description

<put algorithm description here>

#### Parameters

**Input Image [raster]** <put parameter description here>

**Mask expression [string]** Optional.

<put parameter description here>

Default: *None*

**Connected Component Expression [string]** <put parameter description here>

Default: *None*

**Minimum Object Size [number]** <put parameter description here>

Default: *2*

**OBIA Expression** [string] Optional.

<put parameter description here>

Default: *None*

**Default elevation** [number] <put parameter description here>

Default: *0*

## Outputs

**Output Shape** [vector] <put output description here>

## Console usage

```
processing.runalg('otb:connectedcomponentsegmentation', -in, -mask, -expr, -minsize, -obia, -elev
```

## See also

### Exact Large-Scale Mean-Shift segmentation, step 2

#### Description

<put algorithm description here>

#### Parameters

**Filtered image** [raster] <put parameter description here>

**Spatial image** [raster] Optional.

<put parameter description here>

**Range radius** [number] <put parameter description here>

Default: *15*

**Spatial radius** [number] <put parameter description here>

Default: *5*

**Minimum Region Size** [number] <put parameter description here>

Default: *0*

**Size of tiles in pixel (X-axis)** [number] <put parameter description here>

Default: *500*

**Size of tiles in pixel (Y-axis)** [number] <put parameter description here>

Default: *500*

**Directory where to write temporary files** [file] Optional.

<put parameter description here>

**Temporary files cleaning** [boolean] <put parameter description here>

Default: *True*

**Outputs**

**Output Image [raster]** <put output description here>

**Console usage**

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep2', -in, -inpos, -ranger, -spatial,
```

**See also**

**Exact Large-Scale Mean-Shift segmentation, step 3 (optional)**

**Description**

<put algorithm description here>

**Parameters**

**Input image [raster]** <put parameter description here>

**Segmented image [raster]** <put parameter description here>

**Minimum Region Size [number]** <put parameter description here>

Default: 50

**Size of tiles in pixel (X-axis) [number]** <put parameter description here>

Default: 500

**Size of tiles in pixel (Y-axis) [number]** <put parameter description here>

Default: 500

**Outputs**

**Output Image [raster]** <put output description here>

**Console usage**

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep3optional', -in, -inseg, -minsize,
```

**See also**

**Exact Large-Scale Mean-Shift segmentation, step 4**

**Description**

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Segmented image** [raster] <put parameter description here>

**Size of tiles in pixel (X-axis)** [number] <put parameter description here>

Default: *500*

**Size of tiles in pixel (Y-axis)** [number] <put parameter description here>

Default: *500*

### Outputs

**Output GIS vector file** [vector] <put output description here>

### Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep4', -in, -inseg, -tilesizex, -tile
```

### See also

### Hoover compare segmentation

#### Description

<put algorithm description here>

#### Parameters

**Input ground truth** [raster] <put parameter description here>

**Input machine segmentation** [raster] <put parameter description here>

**Background label** [number] <put parameter description here>

Default: *0*

**Overlapping threshold** [number] <put parameter description here>

Default: *0.75*

**Correct detection score** [number] <put parameter description here>

Default: *0.0*

**Over-segmentation score** [number] <put parameter description here>

Default: *0.0*

**Under-segmentation score** [number] <put parameter description here>

Default: *0.0*

**Missed detection score** [number] <put parameter description here>

Default: *0.0*



## Outputs

**Colored ground truth output [raster]** <put output description here>

**Colored machine segmentation output [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:hoovercomparesegmentation', -ingt, -inms, -bg, -th, -rc, -rf, -ra, -rm, -o
```

## See also

## Segmentation (cc)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Segmentation algorithm [selection]** <put parameter description here>

Options:

- 0 — cc

Default: 0

**Condition [string]** <put parameter description here>

Default: *None*

**Processing mode [selection]** <put parameter description here>

Options:

- 0 — vector

Default: 0

**Writing mode for the output vector file [selection]** <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

**Mask Image [raster]** Optional.

<put parameter description here>

**8-neighbor connectivity [boolean]** <put parameter description here>

Default: *True*

**Stitch polygons** [**boolean**] <put parameter description here>

Default: *True*

**Minimum object size** [**number**] <put parameter description here>

Default: *1*

**Simplify polygons** [**number**] <put parameter description here>

Default: *0.1*

**Layer name** [**string**] <put parameter description here>

Default: *layer*

**Geometry index field name** [**string**] <put parameter description here>

Default: *DN*

**Tiles size** [**number**] <put parameter description here>

Default: *1024*

**Starting geometry index** [**number**] <put parameter description here>

Default: *1*

**OGR options for layer creation** [**string**] Optional.

<put parameter description here>

Default: *None*

## Outputs

**Output vector file** [**vector**] <put output description here>

## Console usage

```
processing.runalg('otb:segmentationcc', -in, -filter, -filter.cc.expr, -mode, -mode.vector.outmode)
```

## See also

### Segmentation (edison)

#### Description

<put algorithm description here>

#### Parameters

**Input Image** [**raster**] <put parameter description here>

**Segmentation algorithm** [**selection**] <put parameter description here>

Options:

- 0 — edison

Default: *0*

**Spatial radius** [**number**] <put parameter description here>

Default: *5*

**Range radius [number]** <put parameter description here>

Default: *15*

**Minimum region size [number]** <put parameter description here>

Default: *100*

**Scale factor [number]** <put parameter description here>

Default: *1*

**Processing mode [selection]** <put parameter description here>

Options:

- 0 — vector

Default: *0*

**Writing mode for the output vector file [selection]** <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: *0*

**Mask Image [raster]** Optional.

<put parameter description here>

**8-neighbor connectivity [boolean]** <put parameter description here>

Default: *True*

**Stitch polygons [boolean]** <put parameter description here>

Default: *True*

**Minimum object size [number]** <put parameter description here>

Default: *1*

**Simplify polygons [number]** <put parameter description here>

Default: *0.1*

**Layer name [string]** <put parameter description here>

Default: *layer*

**Geometry index field name [string]** <put parameter description here>

Default: *DN*

**Tiles size [number]** <put parameter description here>

Default: *1024*

**Starting geometry index [number]** <put parameter description here>

Default: *1*

**OGR options for layer creation [string]** Optional.

<put parameter description here>

Default: *None*

## Outputs

**Output vector file [vector]** <put output description here>

## Console usage

```
processing.runalg('otb:segmentationedison', -in, -filter, -filter.edison.spatialr, -filter.edison
```

## See also

## Segmentation (meanshift)

### Description

<put algorithm description here>

### Parameters

**Input Image [raster]** <put parameter description here>

**Segmentation algorithm [selection]** <put parameter description here>

Options:

- 0 — meanshift

Default: 0

**Spatial radius [number]** <put parameter description here>

Default: 5

**Range radius [number]** <put parameter description here>

Default: 15

**Mode convergence threshold [number]** <put parameter description here>

Default: 0.1

**Maximum number of iterations [number]** <put parameter description here>

Default: 100

**Minimum region size [number]** <put parameter description here>

Default: 100

**Processing mode [selection]** <put parameter description here>

Options:

- 0 — vector

Default: 0

**Writing mode for the output vector file [selection]** <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw

- 3 — ulu

Default: *0*

**Mask Image [raster]** Optional.

<put parameter description here>

**8-neighbor connectivity [boolean]** <put parameter description here>

Default: *True*

**Stitch polygons [boolean]** <put parameter description here>

Default: *True*

**Minimum object size [number]** <put parameter description here>

Default: *1*

**Simplify polygons [number]** <put parameter description here>

Default: *0.1*

**Layer name [string]** <put parameter description here>

Default: *layer*

**Geometry index field name [string]** <put parameter description here>

Default: *DN*

**Tiles size [number]** <put parameter description here>

Default: *1024*

**Starting geometry index [number]** <put parameter description here>

Default: *1*

**OGR options for layer creation [string]** Optional.

<put parameter description here>

Default: *None*

## Outputs

**Output vector file [vector]** <put output description here>

## Console usage

```
processing.runalg('otb:segmentationmeanshift', -in, -filter, -filter.meanshift.spatialr, -filter.
```

## See also

### Segmentation (mprofiles)

#### Description

<put algorithm description here>

## Parameters

**Input Image** [raster] <put parameter description here>

**Segmentation algorithm** [selection] <put parameter description here>

Options:

- 0 — mprofiles

Default: 0

**Profile Size** [number] <put parameter description here>

Default: 5

**Initial radius** [number] <put parameter description here>

Default: 1

**Radius step.** [number] <put parameter description here>

Default: 1

**Threshold of the final decision rule** [number] <put parameter description here>

Default: 1

**Processing mode** [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

**Writing mode for the output vector file** [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

**Mask Image** [raster] Optional.

<put parameter description here>

**8-neighbor connectivity** [boolean] <put parameter description here>

Default: *True*

**Stitch polygons** [boolean] <put parameter description here>

Default: *True*

**Minimum object size** [number] <put parameter description here>

Default: 1

**Simplify polygons** [number] <put parameter description here>

Default: 0.1

**Layer name** [string] <put parameter description here>

Default: *layer*

**Geometry index field name** [string] <put parameter description here>

Default: *DN*

**Tiles size** [number] <put parameter description here>

Default: *1024*

**Starting geometry index** [number] <put parameter description here>

Default: *1*

**OGR options for layer creation** [string] Optional.

<put parameter description here>

Default: *None*

## Outputs

**Output vector file** [vector] <put output description here>

## Console usage

```
processing.runalg('otb:segmentationmprofiles', -in, -filter, -filter.mprofiles.size, -filter.mpro
```

## See also

## Segmentation (watershed)

### Description

<put algorithm description here>

### Parameters

**Input Image** [raster] <put parameter description here>

**Segmentation algorithm** [selection] <put parameter description here>

Options:

- 0 — watershed

Default: *0*

**Depth Threshold** [number] <put parameter description here>

Default: *0.01*

**Flood Level** [number] <put parameter description here>

Default: *0.1*

**Processing mode** [selection] <put parameter description here>

Options:

- 0 — vector

Default: *0*

**Writing mode for the output vector file** [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: *0*

**Mask Image** [raster] Optional.

<put parameter description here>

**8-neighbor connectivity** [boolean] <put parameter description here>

Default: *True*

**Stitch polygons** [boolean] <put parameter description here>

Default: *True*

**Minimum object size** [number] <put parameter description here>

Default: *1*

**Simplify polygons** [number] <put parameter description here>

Default: *0.1*

**Layer name** [string] <put parameter description here>

Default: *layer*

**Geometry index field name** [string] <put parameter description here>

Default: *DN*

**Tiles size** [number] <put parameter description here>

Default: *1024*

**Starting geometry index** [number] <put parameter description here>

Default: *1*

**OGR options for layer creation** [string] Optional.

<put parameter description here>

Default: *None*

## Outputs

**Output vector file** [vector] <put output description here>

## Console usage

```
processing.runalg('otb:segmentationwatershed', -in, -filter, -filter.watershed.threshold, -filter
```

## See also

.



## 18.4.9 Stereo

### Stereo Framework

#### Description

<put algorithm description here>

#### Parameters

**Input images list** [**multipleinput: rasters**] <put parameter description here>

**Couples list** [**string**] Optional.

<put parameter description here>

Default: *None*

**Image channel used for the block matching** [**number**] <put parameter description here>

Default: *1*

**Default elevation** [**number**] <put parameter description here>

Default: *0*

**Output resolution** [**number**] <put parameter description here>

Default: *1*

**NoData value** [**number**] <put parameter description here>

Default: *-32768*

**Method to fuse measures in each DSM cell** [**selection**] <put parameter description here>

Options:

- 0 — max
- 1 — min
- 2 — mean
- 3 — acc

Default: *0*

**Parameters estimation modes** [**selection**] <put parameter description here>

Options:

- 0 — fit
- 1 — user

Default: *0*

**Upper Left X** [**number**] <put parameter description here>

Default: *0.0*

**Upper Left Y** [**number**] <put parameter description here>

Default: *0.0*

**Size X** [**number**] <put parameter description here>

Default: *0*

**Size Y [number]** <put parameter description here>

Default: 0

**Pixel Size X [number]** <put parameter description here>

Default: 0.0

**Pixel Size Y [number]** <put parameter description here>

Default: 0.0

**Output Cartographic Map Projection [selection]** <put parameter description here>

Options:

- 0 — utm
- 1 — lambert2
- 2 — lambert93
- 3 — wgs
- 4 — epsg

Default: 3

**Zone number [number]** <put parameter description here>

Default: 31

**Northern Hemisphere [boolean]** <put parameter description here>

Default: *True*

**EPSG Code [number]** <put parameter description here>

Default: 4326

**Step of the deformation grid (in pixels) [number]** <put parameter description here>

Default: 16

**Sub-sampling rate for epipolar grid inversion [number]** <put parameter description here>

Default: 10

**Block-matching metric [selection]** <put parameter description here>

Options:

- 0 — ssdmean
- 1 — ssd
- 2 — ncc
- 3 — lp

Default: 0

**p value [number]** <put parameter description here>

Default: 1

**Radius of blocks for matching filter (in pixels) [number]** <put parameter description here>

Default: 2

**Minimum altitude offset (in meters) [number]** <put parameter description here>

Default: -20

**Maximum altitude offset (in meters) [number]** <put parameter description here>

Default: *20*

**Use bijection consistency in block matching strategy [boolean]** <put parameter description here>

Default: *True*

**Use median disparities filtering [boolean]** <put parameter description here>

Default: *True*

**Correlation metric threshold [number]** <put parameter description here>

Default: *0.6*

**Input left mask [raster]** Optional.

<put parameter description here>

**Input right mask [raster]** Optional.

<put parameter description here>

**Discard pixels with low local variance [number]** <put parameter description here>

Default: *50*

**Available RAM (Mb) [number]** <put parameter description here>

Default: *128*

## Outputs

**Output DSM [raster]** <put output description here>

## Console usage

```
processing.runalg('otb:stereoframework', -input.il, -input.co, -input.channel, -elev.default, -out
```

## See also

.

## 18.4.10 Vector

### Concatenate

#### Description

<put algorithm description here>

#### Parameters

**Input VectorDatas to concatenate [multipleinput: any vectors]** <put parameter description here>

## Outputs

**Concatenated VectorData** [**vector**] <put output description here>

## Console usage

```
processing.runalg('otb:concatenate', -vd, -out)
```

## See also

.

## 18.5 QGIS algorithm provider

QGIS algorithm provider implements various analysis and geoprocessing operations using mostly only QGIS API. So almost all algorithms from this provider will work “out of the box” without any additional configuration.

This provider incorporates fTools functionality, some algorithms from mmQGIS plugin and also adds its own algorithms.

.

### 18.5.1 Database

#### Import into PostGIS

##### Description

<put algorithm description here>

##### Parameters

**Layer to import** [**vector: any**] <put parameter description here>

**Database (connection name)** [**selection**] <put parameter description here>

Options:

- 0 — local

Default: 0

**Schema (schema name)** [**string**] <put parameter description here>

Default: *public*

**Table to import to (leave blank to use layer name)** [**string**] <put parameter description here>

Default: (*not set*)

**Primary key field** [**tablefield: any**] Optional.

<put parameter description here>

**Geometry column** [**string**] <put parameter description here>

Default: *geom*

**Overwrite [boolean]** <put parameter description here>

Default: *True*

**Create spatial index [boolean]** <put parameter description here>

Default: *True*

**Convert field names to lowercase [boolean]** <put parameter description here>

Default: *True*

**Drop length constraints on character fields [boolean]** <put parameter description here>

Default: *False*

## Outputs

### Console usage

```
processing.runalg('qgis:importintopostgis', input, database, schema, tablename, primary_key, geom)
```

## See also

### PostGIS execute SQL

#### Description

<put algorithm description here>

#### Parameters

**Database [string]** <put parameter description here>

Default: *(not set)*

**SQL query [string]** <put parameter description here>

Default: *(not set)*

## Outputs

### Console usage

```
processing.runalg('qgis:postgisexecutesql', database, sql)
```

## See also

.

## 18.5.2 Raster general

### Set style for raster layer

#### Description

<put algorithm description here>

#### Parameters

**Raster layer** [**raster**] <put parameter description here>

**Style file** [**file**] <put parameter description here>

#### Outputs

**Styled layer** [**raster**] <put output description here>

#### Console usage

```
processing.runalg('qgis:setstyleforrasterlayer', input, style)
```

#### See also

.

## 18.5.3 Raster

### Hypsometric curves

#### Description

Calculate hypsometric curves for features of polygon layer and save them as CSV file for further processing.

#### Parameters

**DEM to analyze** [**raster**] DEM to use for calculating altitudes.

**Boundary layer** [**vector: polygon**] Polygonal vector layer with boundaries of areas used to calculate hypsometric curves.

**Step** [**number**] Distance between curves.

Default: *100.0*

**Use % of area instead of absolute value** [**boolean**] Write area percentage to “Area” field of the CSV file instead of absolute area value.

Default: *False*

## Outputs

**Output directory [directory]** Directory where output will be saved. For each feature from input vector layer CSV file with area and altitude values will be created.

File name consists of prefix `hystogram_` followed by layer name and feature ID.

## Console usage

```
processing.runalg('qgis:hypsometriccurves', input_dem, boundary_layer, step, use_percentage, output_directory)
```

## See also

## Raster layer statistics

### Description

Calculates basic statistics of the raster layer.

### Parameters

**Input layer [raster]** Raster to analyze.

### Outputs

**Statistics [html]** Analysis results in HTML format.

**Minimum value [number]** Minimum cell value.

**Maximum value [number]** Maximum cell value.

**Sum [number]** Sum of all cells values.

**Mean value [number]** Mean cell value.

**valid cells count [number]** Number of cell with data.

**No-data cells count [number]** Number of NODATA cells.

**Standard deviation [number]** Standard deviation of cells values.

### Console usage

```
processing.runalg('qgis:rasterlayerstatistics', input, output_html_file)
```

## See also

## Zonal Statistics

### Description

Calculates some statistics values for pixels of input raster inside certain zones, defined as polygon layer.

Following values calculated for each zone:

- minimum

- maximum
- sum
- count
- mean
- standard deviation
- number of unique values
- range
- variance

### Parameters

**Raster layer** [**raster**] Raster to analyze.

**Raster band** [**number**] Number of raster band to analyze.

Default: *1*

**Vector layer containing zones** [**vector: polygon**] Layer with zones boundaries.

**Output column prefix** [**string**] Prefix for output fields.

Default: *\_*

**Load whole raster in memory** [**boolean**] Determines if raster band will be loaded in memory (**True**) or readed by chunks (**False**). Useful only when disk IO or raster scanning inefficiencies are your limiting factor.

Default: *True*

### Outputs

**Output layer** [**vector**] The resulting layer. Basically this is same layer as zones layer with new columns containing statistics added.

### Console usage

```
processing.runalg('qgis:zonalstatistics', input_raster, raster_band, input_vector, column_prefix,
```

### See also

.

## 18.5.4 Table

### Frequency analysis

#### Description

<put algorithm description here>



### Parameters

**input** [**vector: any**] <put parameter description here>

**fields** [**string**] <put parameter description here>

Default: *(not set)*

### Outputs

**output** [**table**] <put output description here>

### Console usage

```
processing.runalg('qgis:frequencyanalysis', input, fields, output)
```

### See also

.

## 18.5.5 Vector analysis

### Count points in polygon

#### Description

Counts the number of points present in each feature of a polygon layer.

#### Parameters

**Polygons** [**vector: polygon**] Polygons layer.

**Points** [**vector: point**] Points layer.

**Count field name** [**string**] The name of the attribute table column containing the points number.

Default: *NUMPOINTS*

#### Outputs

**Result** [**vector**] Resulting layer with the attribute table containing the new column of the points count.

#### Console usage

```
processing.runalg('qgis:countpointsinpolygon', polygons, points, field, output)
```

See also

### Count points in polygon (weighted)

#### Description

Counts the number of points in each feature of a polygon layer and calculates the mean of the selected field for each feature of the polygon layer. These values will be added to the attribute table of the resulting polygon layer.

#### Parameters

**Polygons** [vector: polygon] Polygons layer.

**Points** [vector: point] Points layer.

**Weight field** [tablefield: any] Weight field of the points attribute table.

**Count field name** [string] Name of the column for the new weighted field.

Default: *NUMPOINTS*

#### Outputs

**Result** [vector] The resulting polygons layer.

#### Console usage

```
processing.runalg('qgis:countpointsinpolygonweighted', polygons, points, weight, field, output)
```

See also

### Count unique points in polygon

#### Description

Counts the number of unique values of a points in a polygons layer. Creates a new polygons layer with an extra column in the attribute table containing the count of unique values for each feature.

#### Parameters

**Polygons** [vector: polygon] Polygons layer.

**Points** [vector: point] Points layer.

**Class field** [tablefield: any] Points layer column name of the unique value chosen.

**Count field name** [string] Column name containing the count of unique values in the resulting polygons layer.

Default: *NUMPOINTS*

#### Outputs

**Result** [vector] The resulting polygons layer.

### Console usage

```
processing.runalg('qgis:countuniquepointsinpolygon', polygons, points, classfield, field, output)
```

### See also

### Distance matrix

#### Description

<put algorithm description here>

#### Parameters

**Input point layer** [**vector: point**] <put parameter description here>

**Input unique ID field** [**tablefield: any**] <put parameter description here>

**Target point layer** [**vector: point**] <put parameter description here>

**Target unique ID field** [**tablefield: any**] <put parameter description here>

**Output matrix type** [**selection**] <put parameter description here>

Options:

- 0 — Linear ( $N \times k \times 3$ ) distance matrix
- 1 — Standard ( $N \times T$ ) distance matrix
- 2 — Summary distance matrix (mean, std. dev., min, max)

Default: 0

**Use only the nearest (k) target points** [**number**] <put parameter description here>

Default: 0

#### Outputs

**Distance matrix** [**table**] <put output description here>

### Console usage

```
processing.runalg('qgis:distancematrix', input_layer, input_field, target_layer, target_field, ma
```

### See also

### Distance to nearest hub

#### Description

<put algorithm description here>

### Parameters

**Source points layer** [vector: any] <put parameter description here>

**Destination hubs layer** [vector: any] <put parameter description here>

**Hub layer name attribute** [tablefield: any] <put parameter description here>

**Output shape type** [selection] <put parameter description here>

Options:

- 0 — Point
- 1 — Line to hub

Default: 0

**Measurement unit** [selection] <put parameter description here>

Options:

- 0 — Meters
- 1 — Feet
- 2 — Miles
- 3 — Kilometers
- 4 — Layer units

Default: 0

### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:distancetonearesthub', points, hubs, field, geometry, unit, output)
```

### See also

#### Generate points (pixel centroids) along line

##### Description

<put algorithm description here>

##### Parameters

**Raster layer** [raster] <put parameter description here>

**Vector layer** [vector: line] <put parameter description here>

##### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:generatepointspixelcentroidsalongline', input_raster, input_vector, output)
```

### See also

## Generate points (pixel centroids) inside polygons

### Description

<put algorithm description here>

### Parameters

**Raster layer** [raster] <put parameter description here>

**Vector layer** [vector: polygon] <put parameter description here>

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:generatepointspixelcentroidsinsidepolygons', input_raster, input_vector, output)
```

### See also

## Hub lines

### Description

Creates hub and spoke diagrams with lines drawn from points on the `Spoke Point` layer to matching points in the `Hub Point` layer. Determination of which hub goes with each point is based on a match between the `Hub ID field` on the hub points and the `Spoke ID field` on the spoke points.

### Parameters

**Hub point layer** [vector: any] <put parameter description here>

**Hub ID field** [tablefield: any] <put parameter description here>

**Spoke point layer** [vector: any] <put parameter description here>

**Spoke ID field** [tablefield: any] <put parameter description here>

### Outputs

**Output** [vector] The resulting layer.

### Console usage

```
processing.runalg('qgis:hublines', hubs, hub_field, spokes, spoke_field, output)
```

### See also

### Mean coordinate(s)

#### Description

Calculates the mean of the coordinates of a layer starting from a field of the attribute table.

#### Parameters

**Input layer** [**vector: any**] <put parameter description here>

**Weight field** [**tablefield: numeric**] Optional.

Field to use if you want to perform a weighted mean.

**Unique ID field** [**tablefield: numeric**] Optional.

Unique field on which the calculation of the mean will be made.

#### Outputs

**Result** [**vector**] The resulting points layer.

### Console usage

```
processing.runalg('qgis:meancoordinates', points, weight, uid, output)
```

### See also

### Nearest neighbour analysis

#### Description

<put algorithm description here>

#### Parameters

**Points** [**vector: point**] <put parameter description here>

#### Outputs

**Result** [**html**] <put output description here>

**Observed mean distance** [**number**] <put output description here>

**Expected mean distance** [**number**] <put output description here>

**Nearest neighbour index** [**number**] <put output description here>

**Number of points** [number] <put output description here>

**Z-Score** [number] <put output description here>

#### Console usage

```
processing.runalg('qgis:nearestneighbouranalysis', points, output)
```

#### See also

### Sum line lengths

#### Description

<put algorithm description here>

#### Parameters

**Lines** [vector: line] <put parameter description here>

**Polygons** [vector: polygon] <put parameter description here>

**Lines length field name** [string] <put parameter description here>

Default: *LENGTH*

**Lines count field name** [string] <put parameter description here>

Default: *COUNT*

#### Outputs

**Result** [vector] <put output description here>

#### Console usage

```
processing.runalg('qgis:sumlinelengths', lines, polygons, len_field, count_field, output)
```

#### See also

## 18.5.6 Vector creation

### Create grid

#### Description

Creates a grid.

## Parameters

**Grid type** [selection] Grid type.

Options:

- 0 — Rectangle (line)
- 1 — Rectangle (polygon)
- 2 — Diamond (polygon)
- 3 — Hexagon (polygon)

Default: *0*

**Width** [number] Horizontal extent of the grid.

Default: *360.0*

**Height** [number] Vertical extent of the grid.

Default: *180.0*

**Horizontal spacing** [number] X-axes spacing between the lines.

Default: *10.0*

**Vertical spacing** [number] Y-axes spacing between the lines.

Default: *10.0*

**Center X** [number] X-coordinate of the grid center.

Default: *0.0*

**Center Y** [number] Y-coordinate of the grid center.

Default: *0.0*

**Output CRS** [crs] Coordinate reference system for grid.

Default: *EPSG:4326*

## Outputs

**Output** [vector] The resulting grid layer (lines or polygons).

## Console usage

```
processing.runalg('qgis:creategrid', type, width, height, hspacing, vspacing, centerx, centery, c
```

## See also

### Points layer from table

#### Description

Creates points layer from geometryless table with columns that contain point coordinates.



### Parameters

**Input layer** [table] Input table

**X field** [tablefield: any] Table column containing the X coordinate.

**Y field** [tablefield: any] Table column containing the Y coordinate.

**Target CRS** [crs] Coordinate reference system to use for layer.

Default: *EPSG:4326*

### Outputs

**Output layer** [vector] The resulting layer.

### Console usage

```
processing.runalg('qgis:pointslayerfromtable', input, xfield, yfield, target_crs, output)
```

### See also

### Points to path

#### Description

<put algorithm description here>

#### Parameters

**Input point layer** [vector: point] <put parameter description here>

**Group field** [tablefield: any] <put parameter description here>

**Order field** [tablefield: any] <put parameter description here>

**Date format (if order field is DateTime)** [string] Optional.

<put parameter description here>

Default: *(not set)*

#### Outputs

**Paths** [vector] <put output description here>

**Directory** [directory] <put output description here>

#### Console usage

```
processing.runalg('qgis:pointstopath', vector, group_field, order_field, date_format, output_line)
```

**See also**

**Random points along line**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: line] <put parameter description here>

**Number of points** [number] <put parameter description here>

Default: 1

**Minimum distance** [number] <put parameter description here>

Default: 0.0

**Outputs**

**Random points** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:randompointsalongline', vector, point_number, min_distance, output)
```

**See also**

**Random points in extent**

**Description**

<put algorithm description here>

**Parameters**

**Input extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Points number** [number] <put parameter description here>

Default: 1

**Minimum distance** [number] <put parameter description here>

Default: 0.0

**Outputs**

**Random points** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:randompointsinextent', extent, point_number, min_distance, output)
```

### See also

## Random points in layer bounds

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: polygon] <put parameter description here>

**Points number** [number] <put parameter description here>

Default: 1

**Minimum distance** [number] <put parameter description here>

Default: 0.0

### Outputs

**Random points** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:randompointsinlayerbounds', vector, point_number, min_distance, output)
```

### See also

## Random points inside polygons (fixed)

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: polygon] <put parameter description here>

**Sampling strategy** [selection] <put parameter description here>

Options:

- 0 — Points count
- 1 — Points density

Default: 0

**Number or density of points [number]** <put parameter description here>

Default: *1.0*

**Minimum distance [number]** <put parameter description here>

Default: *0.0*

### Outputs

**Random points [vector]** <put output description here>

### Console usage

```
processing.runalg('qgis:randompointsinsidepolygonsfixed', vector, strategy, value, min_distance, ...)
```

### See also

## Random points inside polygons (variable)

### Description

<put algorithm description here>

### Parameters

**Input layer [vector: polygon]** <put parameter description here>

**Sampling strategy [selection]** <put parameter description here>

Options:

- 0 — Points count
- 1 — Points density

Default: *0*

**Number field [tablefield: numeric]** <put parameter description here>

**Minimum distance [number]** <put parameter description here>

Default: *0.0*

### Outputs

**Random points [vector]** <put output description here>

### Console usage

```
processing.runalg('qgis:randompointsinsidepolygonsvariable', vector, strategy, field, min_distance, ...)
```

**See also**

**Regular points**

**Description**

<put algorithm description here>

**Parameters**

**Input extent [extent]** <put parameter description here>

Default: *0,1,0,1*

**Point spacing/count [number]** <put parameter description here>

Default: *0.0001*

**Initial inset from corner (LH side) [number]** <put parameter description here>

Default: *0.0*

**Apply random offset to point spacing [boolean]** <put parameter description here>

Default: *False*

**Use point spacing [boolean]** <put parameter description here>

Default: *True*

**Outputs**

**Regular points [vector]** <put output description here>

**Console usage**

```
processing.runalg('qgis:regularpoints', extent, spacing, inset, randomize, is_spacing, output)
```

**See also**

**Vector grid**

**Description**

<put algorithm description here>

**Parameters**

**Grid extent [extent]** <put parameter description here>

Default: *0,1,0,1*

**X spacing [number]** <put parameter description here>

Default: *0.0001*

**Y spacing [number]** <put parameter description here>

Default: *0.0001*

**Grid type** [selection] <put parameter description here>

Options:

- 0 — Output grid as polygons
- 1 — Output grid as lines

Default: 0

### Outputs

**Grid** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:vectorgrid', extent, step_x, step_y, type, output)
```

### See also

.

## 18.5.7 Vector general

### Delete duplicate geometries

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

#### Outputs

**Output** [vector] <put output description here>

#### Console usage

```
processing.runalg('qgis:deleteduplicategeometries', input, output)
```

### See also

### Join attributes by location

#### Description

<put algorithm description here>

## Parameters

**Target vector layer** [**vector: any**] <put parameter description here>

**Join vector layer** [**vector: any**] <put parameter description here>

**Attribute summary** [**selection**] <put parameter description here>

Options:

- 0 — Take attributes of the first located feature
- 1 — Take summary of intersecting features

Default: 0

**Statistics for summary (comma separated)** [**string**] <put parameter description here>

Default: *sum,mean,min,max,median*

**Output table** [**selection**] <put parameter description here>

Options:

- 0 — Only keep matching records
- 1 — Keep all records (including non-matching target records)

Default: 0

## Outputs

**Output layer** [**vector**] <put output description here>

## Console usage

```
processing.runalg('qgis:joinattributesbylocation', target, join, summary, stats, keep, output)
```

## See also

### Join attributes table

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [**vector: any**] <put parameter description here>

**Input layer 2** [**table**] <put parameter description here>

**Table field** [**tablefield: any**] <put parameter description here>

**Table field 2** [**tablefield: any**] <put parameter description here>

#### Outputs

**Output layer** [**vector**] <put output description here>

### Console usage

```
processing.runalg('qgis:joinattributetable', input_layer, input_layer_2, table_field, table_fiel
```

### See also

### Merge vector layers

#### Description

<put algorithm description here>

#### Parameters

**Input layer 1** [vector: any] <put parameter description here>

**Input layer 2** [vector: any] <put parameter description here>

#### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:mergevectorlayers', layer1, layer2, output)
```

### See also

### Polygon from layer extent

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

**Calculate extent for each feature separately** [boolean] <put parameter description here>

Default: *False*

#### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:polygonfromlayerextent', input_layer, by_feature, output)
```



**See also****Reproject layer****Description**

Reprojects a vector layer in a different CRS.

**Parameters**

**Input layer** [vector: any] Layer to reproject.

**Target CRS** [crs] Destination coordinate reference system.

Default: *EPSG:4326*

**Outputs**

**Reprojected layer** [vector] The resulting layer.

**Console usage**

```
processing.runalg('qgis:reprojectlayer', input, target_crs, output)
```

**See also****Save selected features****Description**

Saves the selected features as a new layer.

**Parameters**

**Input layer** [vector: any] Layer to process.

**Outputs**

**Output layer with selected features** [vector] The resulting layer.

**Console usage**

```
processing.runalg('qgis:savesselectedfeatures', input_layer, output_layer)
```

**See also**

**Set style for vector layer**

**Description**

<put algorithm description here>

**Parameters**

**Vector layer** [vector: any] <put parameter description here>

**Style file** [file] <put parameter description here>

**Outputs**

**Styled layer** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:setstyleforvectorlayer', input, style)
```

**See also**

**Snap points to grid**

**Description**

<put algorithm description here>

**Parameters**

**Input Layer** [vector: any] <put parameter description here>

**Horizontal spacing** [number] <put parameter description here>

Default: *0.1*

**Vertical spacing** [number] <put parameter description here>

Default: *0.1*

**Outputs**

**Output** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:snappointstogrid', input, hspacing, vspacing, output)
```

See also

## Split vector layer

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Unique ID field** [tablefield: any] <put parameter description here>

### Outputs

**Output directory** [directory] <put output description here>

### Console usage

```
processing.runalg('qgis:splitvectorlayer', input, field, output)
```

See also

.

## 18.5.8 Vector geometry

### Concave hull

#### Description

<put algorithm description here>

#### Parameters

**Input point layer** [vector: point] <put parameter description here>

**Threshold (0-1, where 1 is equivalent with Convex Hull)** [number] <put parameter description here>

Default: *0.3*

**Allow holes** [boolean] <put parameter description here>

Default: *True*

**Split multipart geometry into singleparts geometries** [boolean] <put parameter description here>

Default: *False*

## Outputs

**Concave hull** [vector] <put output description here>

## Console usage

```
processing.runalg('qgis:concavehull', input, alpha, holes, no_multigeometry, output)
```

## See also

## Convert geometry type

### Description

Converts a geometry type to another one.

### Parameters

**Input layer** [vector: any] Layer in input.

**New geometry type** [selection] Type of conversion to perform.

Options:

- 0 — Centroids
- 1 — Nodes
- 2 — Linestrings
- 3 — Multilinestrings
- 4 — Polygons

Default: 0

## Outputs

**Output** [vector] The resulting layer.

## Console usage

```
processing.runalg('qgis:convertgeometrytype', input, type, output)
```

## See also

## Convex hull

### Description

<put algorithm description here>

### Parameters

**Input layer** [**vector: any**] <put parameter description here>

**Field (optional, only used if creating convex hulls by classes)** [**tablefield: any**]

Optional.

<put parameter description here>

**Method** [**selection**] <put parameter description here>

Options:

- 0 — Create single minimum convex hull
- 1 — Create convex hulls based on field

Default: 0

### Outputs

**Convex hull** [**vector**] <put output description here>

### Console usage

```
processing.runalg('qgis:convexhull', input, field, method, output)
```

### See also

#### Create points along lines

#### Description

<put algorithm description here>

### Parameters

**lines** [**vector: any**] <put parameter description here>

**distance** [**number**] <put parameter description here>

Default: 1

**startpoint** [**number**] <put parameter description here>

Default: 0

**endpoint** [**number**] <put parameter description here>

Default: 0

### Outputs

**output** [**vector**] <put output description here>

### Console usage

```
processing.runalg('qgis:createpointsalonglines', lines, distance, startpoint, endpoint, output)
```

### See also

### Delaunay triangulation

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: point] <put parameter description here>

#### Outputs

**Delaunay triangulation** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:delaunaytriangulation', input, output)
```

### See also

### Densify geometries given an interval

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: polygon, line] <put parameter description here>

**Interval between Vertices to add** [number] <put parameter description here>

Default: 1.0

#### Outputs

**Densified layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:densifygeometriesgivenaninterval', input, interval, output)
```

**See also**

**Densify geometries**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: polygon, line] <put parameter description here>

**Vertices to add** [number] <put parameter description here>

Default: *1*

**Outputs**

**Densified layer** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:densifygeometries', input, vertices, output)
```

**See also**

**Dissolve**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: polygon, line] <put parameter description here>

**Dissolve all (do not use field)** [boolean] <put parameter description here>

Default: *True*

**Unique ID field** [tablefield: any] Optional.

<put parameter description here>

**Outputs**

**Dissolved** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:dissolve', input, dissolve_all, field, output)
```

See also

## Eliminate sliver polygons

### Description

<put algorithm description here>

### Parameters

**Input layer** [**vector: polygon**] <put parameter description here>

**Use current selection in input layer (works only if called from toolbox)** [**boolean**]  
<put parameter description here>

Default: *False*

**Selection attribute** [**tablefield: any**] <put parameter description here>

**Comparison** [**selection**] <put parameter description here>

Options:

- 0 — ==
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with
- 7 — contains

Default: *0*

**Value** [**string**] <put parameter description here>

Default: *0*

**Merge selection with the neighbouring polygon with the** [**selection**] <put parameter description here>

Options:

- 0 — Largest area
- 1 — Smallest Area
- 2 — Largest common boundary

Default: *0*

### Outputs

**Cleaned layer** [**vector**] <put output description here>



### Console usage

```
processing.runalg('qgis:eliminatesliverpolygons', input, keepselection, attribute, comparison, co
```

### See also

### Explode lines

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: line] <put parameter description here>

#### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:explodelines', input, output)
```

### See also

### Extract nodes

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: polygon, line] <put parameter description here>

#### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:extractnodes', input, output)
```

**See also**

**Fill holes**

**Description**

<put algorithm description here>

**Parameters**

**Polygons** [vector: any] <put parameter description here>

**Max area** [number] <put parameter description here>

Default: *100000*

**Outputs**

**Results** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:fillholes', polygons, max_area, results)
```

**See also**

**Fixed distance buffer**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Distance** [number] <put parameter description here>

Default: *10.0*

**Segments** [number] <put parameter description here>

Default: *5*

**Dissolve result** [boolean] <put parameter description here>

Default: *False*

**Outputs**

**Buffer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:fixeddistancebuffer', input, distance, segments, dissolve, output)
```

### See also

### Keep n biggest parts

#### Description

<put algorithm description here>

#### Parameters

**Polygons** [vector: polygon] <put parameter description here>

**To keep** [number] <put parameter description here>

Default: 1

#### Outputs

**Results** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:keepnbiggestparts', polygons, to_keep, results)
```

### See also

### Lines to polygons

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: line] <put parameter description here>

#### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:linestopolygons', input, output)
```

## See also

## Multipart to singleparts

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:multiparttosingleparts', input, output)
```

## See also

## Points displacement

### Description

Moves overlapped points at small distance, that they all become visible. The result is very similar to the output of the “Point displacement” renderer but it is permanent.

### Parameters

**Input layer** [vector: point] Layer with overlapped points.

**Displacement distance** [number] Desired displacement distance **NOTE:** displacement distance should be in same units as layer.

Default: *0.00015*

**Horizontal distribution for two point case** [boolean] Controls distrobution direction in case of two overlapped points. If *True* points wwill be distributed horizontally, otherwise they will be distributed vertically.

Default: *True*

### Outputs

**Output layer** [vector] The resulting layer with shifted overlapped points.

### Console usage

```
processing.runalg('qgis:pointdisplacement', input_layer, distance, horizontal, output_layer)
```

See also

## Polygon centroids

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: polygon] <put parameter description here>

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:polygoncentroids', input_layer, output_layer)
```

See also

## Polygonize

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: line] <put parameter description here>

**Keep table structure of line layer** [boolean] <put parameter description here>

Default: *False*

**Create geometry columns** [boolean] <put parameter description here>

Default: *True*

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:polygonize', input, fields, geometry, output)
```

See also

## Polygons to lines

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: polygon] <put parameter description here>

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:polygonstolines', input, output)
```

See also

## Simplify geometries

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: polygon, line] <put parameter description here>

**Tolerance** [number] <put parameter description here>

Default: 1.0

### Outputs

**Simplified layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:simplifygeometries', input, tolerance, output)
```

**See also**

**Singleparts to multipart**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Unique ID field** [tablefield: any] <put parameter description here>

**Outputs**

**Output layer** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:singlepartstomultipart', input, field, output)
```

**See also**

**Variable distance buffer**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Distance field** [tablefield: any] <put parameter description here>

**Segments** [number] <put parameter description here>

Default: 5

**Dissolve result** [boolean] <put parameter description here>

Default: *False*

**Outputs**

**Buffer** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:variabledistancebuffer', input, field, segments, dissolve, output)
```

See also

## Voronoi polygons

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: point] <put parameter description here>

**Buffer region** [number] <put parameter description here>

Default: 0.0

### Outputs

**Voronoi polygons** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:voronoipolygons', input, buffer, output)
```

See also

.

## 18.5.9 Vector overlay

### Clip

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

**Clip layer** [vector: any] <put parameter description here>

#### Outputs

**Clipped** [vector] <put output description here>

#### Console usage

```
processing.runalg('qgis:clip', input, overlay, output)
```



**See also**

**Difference**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Difference layer** [vector: any] <put parameter description here>

**Outputs**

**Difference** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:difference', input, overlay, output)
```

**See also**

**Intersection**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Intersect layer** [vector: any] <put parameter description here>

**Outputs**

**Intersection** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:intersection', input, input2, output)
```

See also

## Line intersections

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: line] <put parameter description here>

**Intersect layer** [vector: line] <put parameter description here>

**Input unique ID field** [tablefield: any] <put parameter description here>

**Intersect unique ID field** [tablefield: any] <put parameter description here>

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:lineintersections', input_a, input_b, field_a, field_b, output)
```

See also

## Symmetrical difference

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Difference layer** [vector: any] <put parameter description here>

### Outputs

**Symmetrical difference** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:symmetricaldifference', input, overlay, output)
```

**See also**

**Union**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Input layer 2** [vector: any] <put parameter description here>

**Outputs**

**Union** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:union', input, input2, output)
```

**See also**

.

## 18.5.10 Vector selection

### Extract by attribute

**Description**

<put algorithm description here>

**Parameters**

**Input Layer** [vector: any] <put parameter description here>

**Selection attribute** [tablefield: any] <put parameter description here>

**Operator** [selection] <put parameter description here>

Options:

- 0 — =
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with

- 7 — contains

Default: 0

**Value** [string] <put parameter description here>

Default: *(not set)*

### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:extractbyattribute', input, field, operator, value, output)
```

### See also

### Extract by location

#### Description

<put algorithm description here>

#### Parameters

**Layer to select from** [vector: any] <put parameter description here>

**Additional layer (intersection layer)** [vector: any] <put parameter description here>

**Include input features that touch the selection features** [boolean] <put parameter description here>

Default: *False*

**Include input features that overlap/cross the selection features** [boolean] <put parameter description here>

Default: *False*

**Include input features completely within the selection features** [boolean] <put parameter description here>

Default: *False*

#### Outputs

**Selection** [vector] <put output description here>

#### Console usage

```
processing.runalg('qgis:extractbylocation', input, intersect, touches, overlaps, within, output)
```

**See also**

**Random extract**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

**Number/percentage of selected features** [number] <put parameter description here>

Default: 10

**Outputs**

**Selection** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:randomextract', input, method, number, output)
```

**See also**

**Random extract within subsets**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**ID Field** [tablefield: any] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

**Number/percentage of selected features [number]** <put parameter description here>

Default: *10*

### Outputs

**Selection [vector]** <put output description here>

### Console usage

```
processing.runalg('qgis:randomextractwithinsubsets', input, field, method, number, output)
```

### See also

### Random selection

#### Description

<put algorithm description here>

#### Parameters

**Input layer [vector: any]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: *0*

**Number/percentage of selected features [number]** <put parameter description here>

Default: *10*

### Outputs

**Selection [vector]** <put output description here>

### Console usage

```
processing.runalg('qgis:randomselection', input, method, number)
```

### See also

### Random selection within subsets

#### Description

<put algorithm description here>

## Parameters

**Input layer** [**vector: any**] <put parameter description here>

**ID Field** [**tablefield: any**] <put parameter description here>

**Method** [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

**Number/percentage of selected features** [**number**] <put parameter description here>

Default: 10

## Outputs

**Selection** [**vector**] <put output description here>

## Console usage

```
processing.runalg('qgis:randomselectionwithinsubsets', input, field, method, number)
```

## See also

### Select by attribute

### Description

Selects and saves as new layer all features from input layer that satisfy condition.

*NOTE:* algorithm is case-sensitive (“qgis” is different from “Qgis” and “QGIS”)

## Parameters

**Input Layer** [**vector: any**] Layer to process.

**Selection attribute** [**tablefield: any**] Field on which perform the selection.

**Operator** [**selection**] Comparison operator.

Options:

- 0 — =
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with
- 7 — contains

Default: *0*

**Value [string]** Value to compare.

Default: *(not set)*

### Outputs

**Output [vector]** The resulting layer.

### Console usage

```
processing.runalg('qgis:selectbyattribute', input, field, operator, value, output)
```

### See also

### Select by expression

#### Description

<put algorithm description here>

#### Parameters

**Input Layer [vector: any]** <put parameter description here>

**Expression [string]** <put parameter description here>

Default: *(not set)*

**Modify current selection by [selection]** <put parameter description here>

Options:

- 0 — creating new selection
- 1 — adding to current selection
- 2 — removing from current selection

Default: *0*

### Outputs

**Output [vector]** <put output description here>

### Console usage

```
processing.runalg('qgis:selectbyexpression', layername, expression, method)
```



**See also**

**Select by location**

**Description**

<put algorithm description here>

**Parameters**

**Layer to select from** [vector: any] <put parameter description here>

**Additional layer (intersection layer)** [vector: any] <put parameter description here>

**Include input features that touch the selection features** [boolean] <put parameter description here>

Default: *False*

**Include input features that overlap/cross the selection features** [boolean] <put parameter description here>

Default: *False*

**Include input features completely within the selection features** [boolean] <put parameter description here>

Default: *False*

**Modify current selection by** [selection] <put parameter description here>

Options:

- 0 — creating new selection
- 1 — adding to current selection
- 2 — removing from current selection

Default: *0*

**Outputs**

**Selection** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:selectbylocation', input, intersect, touches, overlaps, within, method)
```

**See also**

.

## 18.5.11 Vector table

### Add autoincremental field

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

#### Outputs

**Output layer** [vector] <put output description here>

#### Console usage

```
processing.runalg('qgis:addautoincrementalfield', input, output)
```

#### See also

### Add field to attributes table

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [vector: any] <put parameter description here>

**Field name** [string] <put parameter description here>

Default: *(not set)*

**Field type** [selection] <put parameter description here>

Options:

- 0 — Integer
- 1 — Float
- 2 — String

Default: *0*

**Field length** [number] <put parameter description here>

Default: *10*

**Field precision** [number] <put parameter description here>

Default: *0*

## Outputs

**Output layer** [vector] <put output description here>

## Console usage

```
processing.runalg('qgis:addfieldtoattributetable', input_layer, field_name, field_type, field_length)
```

## See also

## Advanced Python field calculator

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Result field name** [string] <put parameter description here>

Default: *NewField*

**Field type** [selection] <put parameter description here>

Options:

- 0 — Integer
- 1 — Float
- 2 — String

Default: *0*

**Field length** [number] <put parameter description here>

Default: *10*

**Field precision** [number] <put parameter description here>

Default: *0*

**Global expression** [string] Optional.

<put parameter description here>

Default: *(not set)*

**Formula** [string] <put parameter description here>

Default: *value =*

## Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:advancedpythonfieldcalculator', input_layer, field_name, field_type, field_name)
```

### See also

### Basic statistics for numeric fields

#### Description

<put algorithm description here>

#### Parameters

**Input vector layer** [**vector: any**] <put parameter description here>

**Field to calculate statistics on** [**tablefield: numeric**] <put parameter description here>

#### Outputs

**Statistics for numeric field** [**html**] <put output description here>

**Coefficient of Variation** [**number**] <put output description here>

**Minimum value** [**number**] <put output description here>

**Maximum value** [**number**] <put output description here>

**Sum** [**number**] <put output description here>

**Mean value** [**number**] <put output description here>

**Count** [**number**] <put output description here>

**Range** [**number**] <put output description here>

**Median** [**number**] <put output description here>

**Number of unique values** [**number**] <put output description here>

**Standard deviation** [**number**] <put output description here>

### Console usage

```
processing.runalg('qgis:basicstatisticsfornumericfields', input_layer, field_name, output_html_file)
```

### See also

### Basic statistics for text fields

#### Description

<put algorithm description here>

### Parameters

**Input vector layer** [vector: any] <put parameter description here>

**Field to calculate statistics on** [tablefield: string] <put parameter description here>

### Outputs

**Statistics for text field** [html] <put output description here>

**Minimum length** [number] <put output description here>

**Maximum length** [number] <put output description here>

**Mean length** [number] <put output description here>

**Count** [number] <put output description here>

**Number of empty values** [number] <put output description here>

**Number of non-empty values** [number] <put output description here>

**Number of unique values** [number] <put output description here>

### Console usage

```
processing.runalg('qgis:basicstatisticsfortextfields', input_layer, field_name, output_html_file)
```

### See also

#### Create equivalent numerical field

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Class field** [tablefield: any] <put parameter description here>

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:createequivalentnumericalfield', input, field, output)
```

See also

## Delete column

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Field to delete** [tablefield: any] <put parameter description here>

### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:deletecolumn', input, column, output)
```

See also

## Export/Add geometry columns

### Description

<put algorithm description here>

### Parameters

**Input layer** [vector: any] <put parameter description here>

**Calculate using [selection]** <put parameter description here>

Options:

- 0 — Layer CRS
- 1 — Project CRS
- 2 — Ellipsoidal

Default: 0

### Outputs

**Output layer** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:exportaddgeometrycolumns', input, calc_method, output)
```

**See also**

**Field calculator**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [vector: any] <put parameter description here>

**Result field name** [string] <put parameter description here>

Default: *(not set)*

**Field type** [selection] <put parameter description here>

Options:

- 0 — Float
- 1 — Integer
- 2 — String
- 3 — Date

Default: *0*

**Field length** [number] <put parameter description here>

Default: *10*

**Field precision** [number] <put parameter description here>

Default: *3*

**Create new field** [boolean] <put parameter description here>

Default: *True*

**Formula** [string] <put parameter description here>

Default: *(not set)*

**Outputs**

**Output layer** [vector] <put output description here>

**Console usage**

```
processing.runalg('qgis:fieldcalculator', input_layer, field_name, field_type, field_length, field...
```

See also

## List unique values

### Description

Lists unique values of an attribute table field and counts their number.

### Parameters

**Input layer** [**vector: any**] Layer to analyze.

**Target field** [**tablefield: any**] Field to analyze.

### Outputs

**Unique values** [**html**] Analysis results in HTML format.

**Total unique values** [**number**] Total number of unique values in given field.

**Unique values** [**string**] List of all unique values in given field.

### Console usage

```
processing.runalg('qgis:listuniquevalues', input_layer, field_name, output)
```

See also

## Number of unique values in classes

### Description

<put algorithm description here>

### Parameters

**input** [**vector: any**] <put parameter description here>

**class field** [**tablefield: any**] <put parameter description here>

**value field** [**tablefield: any**] <put parameter description here>

### Outputs

**output** [**vector**] <put output description here>

### Console usage

```
processing.runalg('qgis:numberofuniquevaluesinclasses', input, class_field, value_field, output)
```



See also

## Statistics by categories

### Description

<put algorithm description here>

### Parameters

**Input vector layer** [vector: any] <put parameter description here>

**Field to calculate statistics on** [tablefield: numeric] <put parameter description here>

**Field with categories** [tablefield: any] <put parameter description here>

### Outputs

**Statistics** [table] <put output description here>

### Console usage

```
processing.runalg('qgis:statisticsbycategories', input_layer, values_field_name, categories_field)
```

See also

## Text to float

### Description

<put algorithm description here>

### Parameters

**Input Layer** [vector: any] <put parameter description here>

**Text attribute to convert to float** [tablefield: string] <put parameter description here>

### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('qgis:texttofloat', input, field, output)
```

See also

.

## 18.6 R algorithm provider

R also called GNU S, is a strongly functional language and environment to statistically explore data sets, make many graphical displays of data from custom data sets

---

**Bemerkung:** Please remember that Processing contains only R scripts, so you need to install R by yourself and configure Processing properly.

---

### 18.6.1 Basic statistics

#### Frequency table

##### Description

<put algorithm description here>

##### Parameters

**Layer** [vector: any] <put parameter description here>

**Field** [tablefield: any] <put parameter description here>

##### Outputs

**R Console Output** [html] <put output description here>

##### Console usage

```
processing.runalg('r:frequencytable', layer, field, r_console_output)
```

##### See also

#### Kolmogrov-Smirnov test

##### Description

<put algorithm description here>

##### Parameters

**Layer** [vector: any] <put parameter description here>

**Field** [tablefield: any] <put parameter description here>

##### Outputs

**R Console Output** [html] <put output description here>

### Console usage

```
processing.runalg('r:kolmogrovsmirnovtest', layer, field, r_console_output)
```

### See also

### Summary statistics

#### Description

<put algorithm description here>

#### Parameters

**Layer** [vector: any] <put parameter description here>

**Field** [tablefield: any] <put parameter description here>

#### Outputs

**R Console Output** [html] <put output description here>

### Console usage

```
processing.runalg('r:summarystatistics', layer, field, r_console_output)
```

### See also

.

## 18.6.2 Home range

### Characteristic hull method

#### Description

<put algorithm description here>

#### Parameters

**Layer** [vector: any] <put parameter description here>

**Field** [tablefield: any] <put parameter description here>

#### Outputs

**Home\_ranges** [vector] <put output description here>

### Console usage

```
processing.runalg('r:characteristichullmethod', layer, field, home_ranges)
```

### See also

### Kernel h ref

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Field** [tablefield: any] <put parameter description here>

**Grid** [number] <put parameter description here>

Default: *10.0*

**Percentage** [number] <put parameter description here>

Default: *10.0*

**Folder** [directory] Optional.

<put parameter description here>

### Outputs

**Home\_ranges** [vector] <put output description here>

### Console usage

```
processing.runalg('r:kernelhref', layer, field, grid, percentage, folder, home_ranges)
```

### See also

### Minimum convex polygon

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Percentage** [number] <put parameter description here>

Default: *10.0*

**Field** [tablefield: any] <put parameter description here>

## Outputs

**Home\_ranges** [vector] <put output description here>

## Console usage

```
processing.runalg('r:minimumconvexpolygon', layer, percentage, field, home_ranges)
```

## See also

## Single-linkage cluster analysis

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Field** [tablefield: any] <put parameter description here>

**Percentage** [number] <put parameter description here>

Default: *10.0*

### Outputs

**R Plots** [html] <put output description here>

**Home\_ranges** [vector] <put output description here>

### Console usage

```
processing.runalg('r:singlelinkageclusteranalysis', layer, field, percentage, rplots, home_ranges)
```

## See also

.

## 18.6.3 Point pattern

### F function

#### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Nsim** [number] <put parameter description here>

Default: *10.0*

### Outputs

**R Plots** [html] <put output description here>

### Console usage

```
processing.runalg('r:ffunction', layer, nsim, rplots)
```

### See also

### G function

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Nsim** [number] <put parameter description here>

Default: *10.0*

### Outputs

**R Plots** [html] <put output description here>

### Console usage

```
processing.runalg('r:gfunction', layer, nsim, rplots)
```

### See also

### Monte-Carlo spatial randomness

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Simulations** [number] <put parameter description here>

Default: *100.0*

**Optional plot name** [string] <put parameter description here>

Default: *(not set)*

### Outputs

**R Plots** [html] <put output description here>

**R Console Output** [html] <put output description here>

### Console usage

```
processing.runalg('r:montecarlospatialrandomness', layer, simulations, optional_plot_name, rplots)
```

### See also

## Quadrat analysis

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

### Outputs

**R Plots** [html] <put output description here>

**R Console Output** [html] <put output description here>

### Console usage

```
processing.runalg('r:quadratanalysis', layer, rplots, r_console_output)
```

### See also

## Random sampling grid

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Size** [number] <put parameter description here>

Default: *10.0*

### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('r:randomsamplinggrid', layer, size, output)
```

### See also

## Regular sampling grid

### Description

<put algorithm description here>

### Parameters

**Layer** [vector: any] <put parameter description here>

**Size** [number] <put parameter description here>

Default: *10.0*

### Outputs

**Output** [vector] <put output description here>

### Console usage

```
processing.runalg('r:regularsamplinggrid', layer, size, output)
```

### See also

## Relative distribution (distance covariate)

### Description

<put algorithm description here>



## Parameters

**Layer** [vector: any] <put parameter description here>

**Covariate** [vector: any] <put parameter description here>

**Covariate name** [string] <put parameter description here>

Default: *mandatory\_covariate\_name\_(no\_spaces)*

**x label** [string] <put parameter description here>

Default: *(not set)*

**Plot name** [string] <put parameter description here>

Default: *(not set)*

**Legend position** [string] <put parameter description here>

Default: *float*

## Outputs

**R Plots** [html] <put output description here>

## Console usage

```
processing.runalg('r:relativedistributiondistancecovariate', layer, covariate, covariate_name, x_
```

## See also

### Relative distribution (raster covariate)

#### Description

<put algorithm description here>

#### Parameters

**points** [vector: any] <put parameter description here>

**covariate** [raster] <put parameter description here>

**covariate name** [string] <put parameter description here>

Default: *mandatory\_covariate\_name\_(no\_spaces)*

**x label** [string] <put parameter description here>

Default: *(not set)*

**plot name** [string] <put parameter description here>

Default: *(not set)*

**legend position** [string] <put parameter description here>

Default: *float*

### Outputs

**R Plots [html]** <put output description here>

### Console usage

```
processing.runalg('r:relativedistributionrastercovariate', points, covariate, covariate_name, x_1,
```

### See also

### Ripley - Rasson spatial domain

#### Description

<put algorithm description here>

#### Parameters

**Layer [vector: any]** <put parameter description here>

#### Outputs

**Output [vector]** <put output description here>

#### Console usage

```
processing.runalg('r:ripleyrassonspatialdomain', layer, output)
```

### See also

.

## 18.6.4 Raster processing

### Advanced raster histogram

#### Description

<put algorithm description here>

#### Parameters

**Layer [raster]** <put parameter description here>

**Dens or Hist [string]** <put parameter description here>

Default: *Hist*

### Outputs

**R Plots [html]** <put output description here>

### Console usage

```
processing.runalg('r:advancedrasterhistogram', layer, dens_or_hist, rplots)
```

### See also

## Raster histogram

### Description

<put algorithm description here>

### Parameters

**Layer [raster]** <put parameter description here>

### Outputs

**R Plots [html]** <put output description here>

### Console usage

```
processing.runalg('r:rasterhistogram', layer, rplots)
```

### See also

.

## 18.6.5 Vector processing

### Histogram

### Description

<put algorithm description here>

### Parameters

**Layer [vector: any]** <put parameter description here>

**Field [tablefield: any]** <put parameter description here>

### Outputs

**R Plots [html]** <put output description here>

### Console usage

```
processing.runalg('r:histogram', layer, field, rplots)
```

### See also

.

## 18.7 SAGA algorithm provider

**SAGA** (System for Automated Geoscientific Analyses) is a free, hybrid, cross-platform GIS software. SAGA provides many geoscientific methods which are bundled in so-called module libraries.

---

**Bemerkung:** Please remember that Processing contains only the interface description, so you need to install SAGA by yourself and configure Processing properly.

---

### 18.7.1 Geostatistics

#### Directional statistics for single grid

##### Description

<put algorithm description here>

##### Parameters

**Grid** [**raster**] <put parameter description here>

**Points** [**vector: any**] Optional.

<put parameter description here>

**Direction** [**Degree**] [**number**] <put parameter description here>

Default: *0.0*

**Tolerance** [**Degree**] [**number**] <put parameter description here>

Default: *0.0*

**Maximum Distance** [**Cells**] [**number**] <put parameter description here>

Default: *0*

**Distance Weighting** [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: *1*

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: *1.0*

## Outputs

**Arithmetic Mean [raster]** <put output description here>

**Difference from Arithmetic Mean [raster]** <put output description here>

**Minimum [raster]** <put output description here>

**Maximum [raster]** <put output description here>

**Range [raster]** <put output description here>

**Variance [raster]** <put output description here>

**Standard Deviation [raster]** <put output description here>

**Mean less Standard Deviation [raster]** <put output description here>

**Mean plus Standard Deviation [raster]** <put output description here>

**Deviation from Arithmetic Mean [raster]** <put output description here>

**Percentile [raster]** <put output description here>

**Directional Statistics for Points [vector]** <put output description here>

## Console usage

```
processing.runalg('saga:directionalstatisticsforsinglegrid', grid, points, direction, tolerance, n)
```

## See also

### Fast representativeness

#### Description

<put algorithm description here>

#### Parameters

**Input [raster]** <put parameter description here>

**Level of Generalisation [number]** <put parameter description here>

Default: *16*

## Outputs

**Output [raster]** <put output description here>

**Output Lod [raster]** <put output description here>

**Output Seeds [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:fastrepresentativeness', input, lod, result, result_lod, seeds)
```

## See also

### Geographically weighted multiple regression (points/grids)

#### Description

<put algorithm description here>

#### Parameters

**Predictors [multipleinput: rasters]** <put parameter description here>

**Output of Regression Parameters [boolean]** <put parameter description here>

Default: *True*

**Points [vector: point]** <put parameter description here>

**Dependent Variable [tablefield: any]** <put parameter description here>

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: *1*

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: *1.0*

**Search Range [selection]** <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

**Search Radius [number]** <put parameter description here>

Default: 100

**Search Mode [selection]** <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

**Number of Points [selection]** <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

**Maximum Number of Observations [number]** <put parameter description here>

Default: 10

**Minimum Number of Observations [number]** <put parameter description here>

Default: 4

## Outputs

**Regression [raster]** <put output description here>

**Coefficient of Determination [raster]** <put output description here>

**Regression Parameters [raster]** <put output description here>

**Residuals [vector]** <put output description here>

## Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregressionpointsgrids', predictors, parameter)
```

## See also

### Geographically weighted multiple regression (points)

#### Description

<put algorithm description here>

#### Parameters

**Points [vector: any]** <put parameter description here>

**Dependent Variable [tablefield: any]** <put parameter description here>

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: 1

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: 1.0

**Search Range [selection]** <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

**Search Radius [number]** <put parameter description here>

Default: 100

**Search Mode [selection]** <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

**Number of Points [selection]** <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

**Maximum Number of Observations [number]** <put parameter description here>

Default: 10

**Minimum Number of Observations [number]** <put parameter description here>

Default: 4

## Outputs

**Regression [vector]** <put output description here>



## Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregressionpoints', points, dependent, distan
```

## See also

## Geographically weighted multiple regression

### Description

<put algorithm description here>

### Parameters

**Points** [vector: point] <put parameter description here>

**Dependent Variable** [tablefield: any] <put parameter description here>

**Target Grids** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Distance Weighting** [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

**Inverse Distance Weighting Power** [number] <put parameter description here>

Default: 1

**Inverse Distance Offset** [boolean] <put parameter description here>

Default: True

**Gaussian and Exponential Weighting Bandwidth** [number] <put parameter description here>

Default: 1

**Search Range** [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

**Search Radius** [number] <put parameter description here>

Default: 100

**Search Mode** [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

**Number of Points [selection]** <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

**Maximum Number of Observations [number]** <put parameter description here>

Default: 10

**Minimum Number of Observations [number]** <put parameter description here>

Default: 4

**Output extent [extent]** <put parameter description here>

Default: 0,1,0,1

**Cellsize [number]** <put parameter description here>

Default: 100.0

## Outputs

**Quality [raster]** <put output description here>

**Intercept [raster]** <put output description here>

**Quality [raster]** <put output description here>

**Intercept [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregression', points, dependent, target, dist
```

## See also

### Geographically weighted regression (points/grid)

#### Description

<put algorithm description here>

#### Parameters

**Predictor [raster]** <put parameter description here>

**Points [vector: point]** <put parameter description here>

**Dependent Variable [tablefield: any]** <put parameter description here>

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: *1*

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: *1.0*

**Search Range [selection]** <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: *0*

**Search Radius [number]** <put parameter description here>

Default: *0*

**Search Mode [selection]** <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: *0*

**Number of Points [selection]** <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: *0*

**Maximum Number of Observations [number]** <put parameter description here>

Default: *10*

**Minimum Number of Observations [number]** <put parameter description here>

Default: *4*

## Outputs

**Regression [raster]** <put output description here>

**Coefficient of Determination [raster]** <put output description here>

**Intercept [raster]** <put output description here>

**Slope [raster]** <put output description here>

**Residuals [vector]** <put output description here>

## Console usage

```
processing.runalg('saga:geographicallyweightedregressionpointsgrid', predictor, points, dependent,
```

## See also

## Geographically weighted regression

### Description

<put algorithm description here>

### Parameters

**Points** [**vector: point**] <put parameter description here>

**Dependent Variable** [**tablefield: any**] <put parameter description here>

**Predictor** [**tablefield: any**] <put parameter description here>

**Target Grids** [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Distance Weighting** [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power** [**number**] <put parameter description here>

Default: *0*

**Inverse Distance Offset** [**boolean**] <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth** [**number**] <put parameter description here>

Default: *0.0*

**Search Range** [**selection**] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: *0*

**Search Radius** [**number**] <put parameter description here>

Default: *100*

**Search Mode** [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

**Number of Points** [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

**Maximum Number of Observations** [number] <put parameter description here>

Default: 10

**Minimum Number of Observations** [number] <put parameter description here>

Default: 4

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

## Outputs

**Grid** [raster] <put output description here>

**Quality** [raster] <put output description here>

**Intercept** [raster] <put output description here>

**Slope** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:geographicallyweightedregression', points, dependent, predictor, target, 0)
```

## See also

### Global moran's i for grids

#### Description

<put algorithm description here>

#### Parameters

**Grid** [raster] <put parameter description here>

**Case of contiguity** [selection] <put parameter description here>

Options:

- 0 — [0] Rook
  - 1 — [1] Queen
- Default: 0

### Outputs

**Result** [table] <put output description here>

### Console usage

```
processing.runalg('saga:globalmoransiforgrids', grid, contiguity, result)
```

### See also

## Minimum distance analysis

### Description

Performs a complete distance analysis of a point layer:

- minimum distance of points
- maximum distance of points
- average distance of all the points
- standard deviation of the distance
- duplicated points

### Parameters

**Points** [vector: point] Layer to analyze.

### Outputs

**Minimum Distance Analysis** [table] The resulting table.

### Console usage

```
processing.runalg('saga:minimumdistanceanalysis', points, table)
```

### See also

## Multi-band variation

### Description

<put algorithm description here>

## Parameters

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Radius** [**Cells**] [**number**] <put parameter description here>

Default: *1*

**Distance Weighting** [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power** [**number**] <put parameter description here>

Default: *1*

**Inverse Distance Offset** [**boolean**] <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth** [**number**] <put parameter description here>

Default: *1.0*

## Outputs

**Mean Distance** [**raster**] <put output description here>

**Standard Deviation** [**raster**] <put output description here>

**Distance** [**raster**] <put output description here>

## Console usage

```
processing.runalg('saga:multibandvariation', bands, radius, distance_weighting_weighting, distance_offset)
```

## See also

### Multiple regression analysis (grid/grids)

#### Description

<put algorithm description here>

#### Parameters

**Dependent** [**raster**] <put parameter description here>

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Grid Interpolation** [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

**Include X Coordinate [boolean]** <put parameter description here>

Default: *True*

**Include Y Coordinate [boolean]** <put parameter description here>

Default: *True*

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] include all
- 1 — [1] forward
- 2 — [2] backward
- 3 — [3] stepwise

Default: 0

**P in [number]** <put parameter description here>

Default: 5

**P out [number]** <put parameter description here>

Default: 5

## Outputs

**Regression [raster]** <put output description here>

**Residuals [raster]** <put output description here>

**Details: Coefficients [table]** <put output description here>

**Details: Model [table]** <put output description here>

**Details: Steps [table]** <put output description here>

## Console usage

```
processing.runalg('saga:multipleregressionanalysisgridgrids', dependent, grids, interpol, coord_x,
```

## See also

### Multiple regression analysis (points/grids)

#### Description

<put algorithm description here>



## Parameters

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Shapes** [**vector: any**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Grid Interpolation** [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

**Include X Coordinate** [**boolean**] <put parameter description here>

Default: *True*

**Include Y Coordinate** [**boolean**] <put parameter description here>

Default: *True*

**Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] include all
- 1 — [1] forward
- 2 — [2] backward
- 3 — [3] stepwise

Default: 0

**P in** [**number**] <put parameter description here>

Default: 5

**P out** [**number**] <put parameter description here>

Default: 5

## Outputs

**Details: Coefficients** [**table**] <put output description here>

**Details: Model** [**table**] <put output description here>

**Details: Steps** [**table**] <put output description here>

**Residuals** [**vector**] <put output description here>

**Regression** [**raster**] <put output description here>

## Console usage

```
processing.runalg('saga:multipleregressionanalysispointsgrids', grids, shapes, attribute, interpo
```

**See also**

**Polynomial regression**

**Description**

<put algorithm description here>

**Parameters**

**Points** [**vector: any**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Polynom** [**selection**] <put parameter description here>

Options:

- 0 — [0] simple planar surface
- 1 — [1] bi-linear saddle
- 2 — [2] quadratic surface
- 3 — [3] cubic surface
- 4 — [4] user defined

Default: 0

**Maximum X Order** [**number**] <put parameter description here>

Default: 4

**Maximum Y Order** [**number**] <put parameter description here>

Default: 4

**Maximum Total Order** [**number**] <put parameter description here>

Default: 4

**Trend Surface** [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Output extent** [**extent**] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [**number**] <put parameter description here>

Default: 100.0

**Outputs**

**Residuals** [**vector**] <put output description here>

**Grid** [**raster**] <put output description here>

### Console usage

```
processing.runalg('saga:polynomialregression', points, attribute, polynom, xorder, yorder, torder)
```

### See also

### Radius of variance (grid)

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Standard Deviation [number]** <put parameter description here>

Default: 1.0

**Maximum Search Radius (cells) [number]** <put parameter description here>

Default: 20

**Type of Output [selection]** <put parameter description here>

Options:

- 0 — [0] Cells
- 1 — [1] Map Units

Default: 0

#### Outputs

**Variance Radius [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:radiusofvariancegrid', input, variance, radius, output, result)
```

### See also

### Regression analysis

#### Description

<put algorithm description here>

## Parameters

**Grid [raster]** <put parameter description here>

**Shapes [vector: any]** <put parameter description here>

**Attribute [tablefield: any]** <put parameter description here>

**Grid Interpolation [selection]** <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

**Regression Function [selection]** <put parameter description here>

Options:

- 0 — [0]  $Y = a + b * X$  (linear)
- 1 — [1]  $Y = a + b / X$
- 2 — [2]  $Y = a / (b - X)$
- 3 — [3]  $Y = a * X^b$  (power)
- 4 — [4]  $Y = a e^{(b * X)}$  (exponential)
- 5 — [5]  $Y = a + b * \ln(X)$  (logarithmic)

Default: 0

## Outputs

**Regression [raster]** <put output description here>

**Residuals [vector]** <put output description here>

## Console usage

```
processing.runalg('saga:regressionanalysis', grid, shapes, attribute, interpol, method, regression)
```

## See also

## Representativeness

## Description

<put algorithm description here>

**Parameters****Grid [raster]** <put parameter description here>**Radius (Cells) [number]** <put parameter description here>Default: *10***Exponent [number]** <put parameter description here>Default: *1***Outputs****Representativeness [raster]** <put output description here>**Console usage**

```
processing.runalg('saga:representativeness', input, radius, exponent, result)
```

**See also****Residual analysis****Description**

&lt;put algorithm description here&gt;

**Parameters****Grid [raster]** <put parameter description here>**Radius (Cells) [number]** <put parameter description here>Default: *7***Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0***Inverse Distance Weighting Power [number]** <put parameter description here>Default: *1***Inverse Distance Offset [boolean]** <put parameter description here>Default: *True***Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>Default: *1.0*

## Outputs

**Mean Value** [raster] <put output description here>  
**Difference from Mean Value** [raster] <put output description here>  
**Standard Deviation** [raster] <put output description here>  
**Value Range** [raster] <put output description here>  
**Minimum Value** [raster] <put output description here>  
**Maximum Value** [raster] <put output description here>  
**Deviation from Mean Value** [raster] <put output description here>  
**Percentile** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:residualanalysis', grid, radius, distance_weighting_weighting, distance_w
```

## See also

### Spatial point pattern analysis

#### Description

<put algorithm description here>

#### Parameters

**Points** [vector: point] <put parameter description here>  
**Vertex Distance** [Degree] [number] <put parameter description here>  
Default: 5

#### Outputs

**Mean Centre** [vector] <put output description here>  
**Standard Distance** [vector] <put output description here>  
**Bounding Box** [vector] <put output description here>

#### Console usage

```
processing.runalg('saga:spatialpointpatternanalysis', points, step, centre, stddist, bbox)
```

## See also

### Statistics for grids

#### Description

<put algorithm description here>

**Parameters**

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Outputs**

**Arithmetic Mean** [**raster**] <put output description here>

**Minimum** [**raster**] <put output description here>

**Maximum** [**raster**] <put output description here>

**Variance** [**raster**] <put output description here>

**Standard Deviation** [**raster**] <put output description here>

**Mean less Standard Deviation** [**raster**] <put output description here>

**Mean plus Standard Deviation** [**raster**] <put output description here>

**Console usage**

```
processing.runalg('saga:statisticsforgrids', grids, mean, min, max, var, stddev, stddevlo, stddevhi)
```

**See also**

**Variogram cloud**

**Description**

<put algorithm description here>

**Parameters**

**Points** [**vector: point**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Maximum Distance** [**number**] <put parameter description here>

Default: 0.0

**Skip Number** [**number**] <put parameter description here>

Default: 1

**Outputs**

**Variogram Cloud** [**table**] <put output description here>

**Console usage**

```
processing.runalg('saga:variogramcloud', points, field, distmax, nskip, result)
```

**See also**

**Variogram surface**

**Description**

<put algorithm description here>

**Parameters**

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Number of Distance Classes** [number] <put parameter description here>

Default: 10

**Skip Number** [number] <put parameter description here>

Default: 1

**Outputs**

**Number of Pairs** [raster] <put output description here>

**Variogram Surface** [raster] <put output description here>

**Covariance Surface** [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:variogramsurface', points, field, distcount, nskip, count, variance, covar
```

**See also**

**Zonal grid statistics**

**Description**

<put algorithm description here>

**Parameters**

**Zone Grid** [raster] <put parameter description here>

**Categorical Grids** [multipleinput: rasters] Optional.

<put parameter description here>

**Grids to analyse** [multipleinput: rasters] Optional.

<put parameter description here>

**Aspect** [raster] Optional.

<put parameter description here>



**Short Field Names** [boolean] <put parameter description here>

Default: *True*

### Outputs

**Zonal Statistics** [table] <put output description here>

### Console usage

```
processing.runalg('saga:zonalgridstatistics', zones, catlist, statlist, aspect, shortnames, outtal
```

### See also

.

## 18.7.2 Grid analysis

### Accumulated cost (anisotropic)

#### Description

<put algorithm description here>

#### Parameters

**Cost Grid** [raster] <put parameter description here>

**Direction of max cost** [raster] <put parameter description here>

**Destination Points** [raster] <put parameter description here>

**k factor** [number] <put parameter description here>

Default: *1*

**Threshold for different route** [number] <put parameter description here>

Default: *0*

#### Outputs

**Accumulated Cost** [raster] <put output description here>

#### Console usage

```
processing.runalg('saga:accumulatedcostanisotropic', cost, direction, points, k, threshold, accco
```

**See also**

**Accumulated cost (isotropic)**

**Description**

<put algorithm description here>

**Parameters**

**Cost Grid** [raster] <put parameter description here>

**Destination Points** [raster] <put parameter description here>

**Threshold for different route** [number] <put parameter description here>

Default: 0.0

**Outputs**

**Accumulated Cost** [raster] <put output description here>

**Closest Point** [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:accumulatedcostisotropic', cost, points, threshold, acccost, closestpt)
```

**See also**

**Aggregation index**

**Description**

<put algorithm description here>

**Parameters**

**Input Grid** [raster] <put parameter description here>

**Max. Number of Classes** [number] <put parameter description here>

Default: 5

**Outputs**

**Result** [table] <put output description here>

**Console usage**

```
processing.runalg('saga:aggregationindex', input, maxnumclass, result)
```

**See also**

**Analytical hierarchy process**

**Description**

<put algorithm description here>

**Parameters**

**Input Grids** [**multipleinput: rasters**] <put parameter description here>

**Pairwise Comparisons Table** [**table**] <put parameter description here>

**Outputs**

**Output Grid** [**raster**] <put output description here>

**Console usage**

```
processing.runalg('saga:analyticalhierarchyprocess', grids, table, output)
```

**See also**

**Cross-classification and tabulation**

**Description**

<put algorithm description here>

**Parameters**

**Input Grid 1** [**raster**] <put parameter description here>

**Input Grid 2** [**raster**] <put parameter description here>

**Max. Number of Classes** [**number**] <put parameter description here>

Default: 5

**Outputs**

**Cross-Classification Grid** [**raster**] <put output description here>

**Cross-Tabulation Table** [**table**] <put output description here>

**Console usage**

```
processing.runalg('saga:crossclassificationandtabulation', input, input2, maxnumclass, resultgrid)
```

**See also**

**Fragmentation (alternative)**

**Description**

<put algorithm description here>

**Parameters**

**Classification** [raster] <put parameter description here>

**Class Identifier** [number] <put parameter description here>

Default: 1

**Neighborhood Min** [number] <put parameter description here>

Default: 1

**Neighborhood Max** [number] <put parameter description here>

Default: 1

**Level Aggregation** [selection] <put parameter description here>

Options:

- 0 — [0] average
- 1 — [1] multiplicative

Default: 0

**Add Border** [boolean] <put parameter description here>

Default: *True*

**Connectivity Weighting** [number] <put parameter description here>

Default: 1.1

**Minimum Density** [Percent] [number] <put parameter description here>

Default: 10

**Minimum Density for Interior Forest** [Percent] [number] <put parameter description here>

Default: 99

**Search Distance Increment** [number] <put parameter description here>

Default: 0.0

**Density from Neighbourhood** [boolean] <put parameter description here>

Default: *True*

**Outputs**

**Density** [Percent] [raster] <put output description here>

**Connectivity** [Percent] [raster] <put output description here>

**Fragmentation** [raster] <put output description here>

**Summary** [table] <put output description here>

### Console usage

```
processing.runalg('saga:fragmentationalternative', classes, class, neighborhood_min, neighborhood_max)
```

### See also

### Fragmentation classes from density and connectivity

#### Description

<put algorithm description here>

#### Parameters

**Density** [**Percent**] [**raster**] <put parameter description here>

**Connectivity** [**Percent**] [**raster**] <put parameter description here>

**Add Border** [**boolean**] <put parameter description here>

Default: *True*

**Connectivity Weighting** [**number**] <put parameter description here>

Default: *0*

**Minimum Density** [**Percent**] [**number**] <put parameter description here>

Default: *10*

**Minimum Density for Interior Forest** [**Percent**] [**number**] <put parameter description here>

Default: *99*

#### Outputs

**Fragmentation** [**raster**] <put output description here>

### Console usage

```
processing.runalg('saga:fragmentationclassesfromdensityandconnectivity', density, connectivity, border)
```

### See also

### Fragmentation (standard)

#### Description

<put algorithm description here>

## Parameters

**Classification** [raster] <put parameter description here>

**Class Identifier** [number] <put parameter description here>

Default: 1

**Neighborhood Min** [number] <put parameter description here>

Default: 1

**Neighborhood Max** [number] <put parameter description here>

Default: 3

**Level Aggregation** [selection] <put parameter description here>

Options:

- 0 — [0] average
- 1 — [1] multiplicative

Default: 0

**Add Border** [boolean] <put parameter description here>

Default: *True*

**Connectivity Weighting** [number] <put parameter description here>

Default: 1.1

**Minimum Density [Percent] [number]** <put parameter description here>

Default: 10

**Minimum Density for Interior Forest [Percent] [number]** <put parameter description here>

Default: 99

**Neighborhood Type** [selection] <put parameter description here>

Options:

- 0 — [0] square
- 1 — [1] circle

Default: 0

**Include diagonal neighbour relations** [boolean] <put parameter description here>

Default: *True*

## Outputs

**Density [Percent] [raster]** <put output description here>

**Connectivity [Percent] [raster]** <put output description here>

**Fragmentation [raster]** <put output description here>

**Summary [table]** <put output description here>

## Console usage

```
processing.runalg('saga:fragmentationstandard', classes, class, neighborhood_min, neighborhood_max)
```

**See also**

**Layer of extreme value**

**Description**

<put algorithm description here>

**Parameters**

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] Maximum
- 1 — [1] Minimum

Default: 0

**Outputs**

**Result** [**raster**] <put output description here>

**Console usage**

```
processing.runalg('saga:layerofextremevalue', grids, criteria, result)
```

**See also**

**Least cost paths**

**Description**

<put algorithm description here>

**Parameters**

**Source Point (s)** [**vector: point**] <put parameter description here>

**Accumulated cost** [**raster**] <put parameter description here>

**Values** [**multipleinput: rasters**] Optional.

<put parameter description here>

**Outputs**

**Profile (points)** [**vector**] <put output description here>

**Profile (lines)** [**vector**] <put output description here>

### Console usage

```
processing.runalg('saga:leastcostpaths', source, dem, values, points, line)
```

### See also

## Ordered Weighted Averaging

### Description

<put algorithm description here>

### Parameters

**Input Grids** [**multipleinput: rasters**] <put parameter description here>

**Weights** [**fixedtable**] <put parameter description here>

### Outputs

**Output Grid** [**raster**] <put output description here>

### Console usage

```
processing.runalg('saga:orderedweightedaveraging', grids, weights, output)
```

### See also

## Pattern analysis

### Description

<put algorithm description here>

### Parameters

**Input Grid** [**raster**] <put parameter description here>

**Size of Analysis Window** [**selection**] <put parameter description here>

Options:

- 0 — [0] 3 X 3
- 1 — [1] 5 X 5
- 2 — [2] 7 X 7

Default: 0

**Max. Number of Classes** [**number**] <put parameter description here>

Default: 0



## Outputs

**Relative Richness [raster]** <put output description here>

**Diversity [raster]** <put output description here>

**Dominance [raster]** <put output description here>

**Fragmentation [raster]** <put output description here>

**Number of Different Classes [raster]** <put output description here>

**Center Versus Neighbours [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:patternanalysis', input, winsize, maxnumclass, relative, diversity, domin
```

## See also

### Soil texture classification

#### Description

<put algorithm description here>

#### Parameters

**Sand [raster]** Optional.

<put parameter description here>

**Silt [raster]** Optional.

<put parameter description here>

**Clay [raster]** Optional.

<put parameter description here>

## Outputs

**Soil Texture [raster]** <put output description here>

**Sum [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:soiltextureclassification', sand, silt, clay, texture, sum)
```

## See also

.

### 18.7.3 Grid calculus

#### Function

#### Description

<put algorithm description here>

#### Parameters

**xmin** [number] <put parameter description here>

Default: 0.0

**xmax** [number] <put parameter description here>

Default: 0.0

**ymin** [number] <put parameter description here>

Default: 0.0

**ymax** [number] <put parameter description here>

Default: 0.0

**Formula** [string] <put parameter description here>

Default: (not set)

#### Outputs

**Function** [raster] <put output description here>

#### Console usage

```
processing.runalg('saga:function', xmin, xmax, ymin, ymax, formul, result)
```

#### See also

#### Fuzzify

#### Description

<put algorithm description here>

#### Parameters

**Grid** [raster] <put parameter description here>

**A** [number] <put parameter description here>

Default: 0.0

**B** [number] <put parameter description here>

Default: 0.0

**C [number]** <put parameter description here>

Default: *0.0*

**D [number]** <put parameter description here>

Default: *0.0*

**Membership Function Type [selection]** <put parameter description here>

Options:

- 0 — [0] linear
- 1 — [1] sigmoidal
- 2 — [2] j-shaped

Default: *0*

**Adjust to Grid [boolean]** <put parameter description here>

Default: *True*

## Outputs

**Fuzzified Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:fuzzify', input, a, b, c, d, type, autofit, output)
```

## See also

## Fuzzy intersection (and)

### Description

<put algorithm description here>

### Parameters

**Grids [multipleinput: rasters]** <put parameter description here>

**Operator Type [selection]** <put parameter description here>

Options:

- 0 — [0] min(a, b) (non-interactive)
- 1 — [1] a \* b
- 2 — [2] max(0, a + b - 1)

Default: *0*

## Outputs

**Intersection [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:fuzzyintersectionand', grids, type, and)
```

### See also

### Fuzzy union (or)

#### Description

<put algorithm description here>

#### Parameters

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Operator Type** [**selection**] <put parameter description here>

Options:

- 0 — [0]  $\max(a, b)$  (non-interactive)
- 1 — [1]  $a + b - a * b$
- 2 — [2]  $\min(1, a + b)$

Default: 0

#### Outputs

**Union** [**raster**] <put output description here>

### Console usage

```
processing.runalg('saga:fuzzyunionor', grids, type, or)
```

### See also

### Geometric figures

#### Description

Draws simple geometric figures.

#### Parameters

**Cell Count** [**number**] Number of cells to use.

Default: 0

**Cell Size** [**number**] Size of the single cell.

Default: 0

**Figure [selection]** Type of the figure.

Options:

- 0 — [0] Cone (up)
- 1 — [1] Cone (down)
- 2 — [2] Plane

Default: 0

**Direction of Plane [Degree] [number]** Rotation factor in degrees.

Default: 0

## Outputs

**Result [raster]** The resulting layer.

## Console usage

```
processing.runalg('saga:geometricfigures', cell_count, cell_size, figure, plane, result)
```

## See also

## Gradient vector from cartesian to polar coordinates

### Description

<put algorithm description here>

### Parameters

**X Component [raster]** <put parameter description here>

**Y Component [raster]** <put parameter description here>

**Polar Angle Units [selection]** <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: 0

**Polar Coordinate System [selection]** <put parameter description here>

Options:

- 0 — [0] mathematical
- 1 — [1] geographical
- 2 — [2] user defined

Default: 0

**User defined Zero Direction [number]** <put parameter description here>

Default: 0.0

**User defined Orientation [selection]** <put parameter description here>

Options:

- 0 — [0] clockwise
- 1 — [1] counterclockwise

Default: 0

## Outputs

**Direction [raster]** <put output description here>

**Length [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:gradientvectorfromcartesiantopolarcoordinates', dx, dy, units, system, sy
```

## See also

## Gradient vector from polar to cartesian coordinates

### Description

<put algorithm description here>

### Parameters

**Direction [raster]** <put parameter description here>

**Length [raster]** <put parameter description here>

**Polar Angle Units [selection]** <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: 0

**Polar Coordinate System [selection]** <put parameter description here>

Options:

- 0 — [0] mathematical
- 1 — [1] geographical
- 2 — [2] user defined

Default: 0

**User defined Zero Direction [number]** <put parameter description here>

Default: 0.0

**User defined Orientation [selection]** <put parameter description here>

Options:

- 0 — [0] clockwise

- 1 — [1] counterclockwise

Default: 0

### Outputs

**X Component [raster]** <put output description here>

**Y Component [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:gradientvectorfrompolarcoordinates', dir, len, units, system, ...)
```

### See also

### Grid difference

#### Description

Creates a new grid layer as the result of the difference between two other grid layers.

#### Parameters

**A [raster]** First layer.

**B [raster]** Second layer.

#### Outputs

**Difference (A - B) [raster]** The resulting layer.

#### Console usage

```
processing.runalg('saga:griddifference', a, b, c)
```

### See also

### Grid division

#### Description

Creates a new grid layer as the result of the division between two other grid layers.

#### Parameters

**Dividend [raster]** First layer.

**Divisor [raster]** Second layer.

## Outputs

**Quotient [raster]** The resulting layer.

## Console usage

```
processing.runalg('saga:griddivision', a, b, c)
```

## See also

## Grid normalisation

### Description

Normalises the grid values according to minimum and maximum values chosen.

### Parameters

**Grid [raster]** Grid to normalize.

**Target Range (min) [number]** Minimum value.

Default: *0*

**Target Range (max) [number]** Maximum value.

Default: *1*

### Outputs

**Normalised Grid [raster]** The resulting layer.

## Console usage

```
processing.runalg('saga:gridnormalisation', input, range_min, range_max, output)
```

## See also

## Grids product

### Description

<put algorithm description here>

### Parameters

**Grids [multipleinput: rasters]** <put parameter description here>

### Outputs

**Product [raster]** <put output description here>



### Console usage

```
processing.runalg('saga:gridsproduct', grids, result)
```

### See also

### Grids sum

#### Description

Creates a new grid layer as the result of the sum of two or more grid layers.

#### Parameters

**Grids [multipleinput: rasters]** Grid layers to sum

#### Outputs

**Sum [raster]** The resulting layer.

### Console usage

```
processing.runalg('saga:gridssum', grids, result)
```

### See also

### Grid standardisation

#### Description

Standardises the grid layer values.

#### Parameters

**Grid [raster]** Grid to process.

**Stretch Factor [number]** stretching factor.

Default: *1.0*

#### Outputs

**Standardised Grid [raster]** The resulting layer.

### Console usage

```
processing.runalg('saga:gridstandardisation', input, stretch, output)
```

**See also**

**Grid volume**

**Description**

<put algorithm description here>

**Parameters**

**Grid [raster]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Count Only Above Base Level
- 1 — [1] Count Only Below Base Level
- 2 — [2] Subtract Volumes Below Base Level
- 3 — [3] Add Volumes Below Base Level

Default: 0

**Base Level [number]** <put parameter description here>

Default: 0.0

**Outputs**

**Console usage**

```
processing.runalg('saga:gridvolume', grid, method, level)
```

**See also**

**Metric conversions**

**Description**

Performs numerical conversions of the grid values.

**Parameters**

**Grid [raster]** Grid to process.

**Conversion [selection]** Conversion type.

Options:

- 0 — [0] radians to degree
- 1 — [1] degree to radians
- 2 — [2] Celsius to Fahrenheit
- 3 — [3] Fahrenheit to Celsius

Default: 0

## Outputs

**Converted Grid [raster]** The resulting layer.

## Console usage

```
processing.runalg('saga:metricconversions', grid, conversion, conv)
```

## See also

## Polynomial trend from grids

### Description

<put algorithm description here>

### Parameters

**Dependent Variables [multipleinput: rasters]** <put parameter description here>

**Independent Variable (per Grid and Cell) [multipleinput: rasters]** Optional.

<put parameter description here>

**Independent Variable (per Grid) [fixedtable]** <put parameter description here>

**Type of Approximated Function [selection]** <put parameter description here>

Options:

- 0 — [0] first order polynom (linear regression)
- 1 — [1] second order polynom
- 2 — [2] third order polynom
- 3 — [3] fourth order polynom
- 4 — [4] fifth order polynom

Default: 0

## Outputs

**Polynomial Coefficients [raster]** <put output description here>

**Coefficient of Determination [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:polynomialtrendfromgrids', grids, y_grids, y_table, polynom, parms, quality)
```

## See also

## Random field

### Description

Generates a random grid layer.

### Parameters

**Width (Cells) [number]** Width of the layer in cells.

Default: *100*

**Height (Cells) [number]** Height of the layer in cells.

Default: *100*

**Cellsize [number]** Cell size to use.

Default: *100.0*

**West [number]** West coordinate of the bottom-left corner of the grid.

Default: *0.0*

**South [number]** South coordinate of the bottom-left corner of the grid.

Default: *0.0*

**Method [selection]** Statistical method used for the calculation.

Options:

- 0 — [0] Uniform
- 1 — [1] Gaussian

Default: *0*

**Range Min [number]** Minimum cell value to use.

Default: *0.0*

**Range Max [number]** Maximum cell value to use.

Default: *1.0*

**Arithmetic Mean [number]** Mean of all the cell values to use.

Default: *0.0*

**Standard Deviation [number]** Standard deviation of all the cell values to use.

Default: *1.0*

### Outputs

**Random Field [raster]** The resulting layer.

### Console usage

```
processing.runalg('saga:randomfield', nx, ny, cellsize, xmin, ymin, method, range_min, range_max,
```

**See also**

**Random terrain generation**

**Description**

<put algorithm description here>

**Parameters**

**Radius (cells) [number]** <put parameter description here>

Default: *10*

**Iterations [number]** <put parameter description here>

Default: *10*

**Target Dimensions [selection]** <put parameter description here>

Options:

- 0 — [0] User defined

Default: *0*

**Grid Size [number]** <put parameter description here>

Default: *1.0*

**Cols [number]** <put parameter description here>

Default: *100*

**Rows [number]** <put parameter description here>

Default: *100*

**Outputs**

**Grid [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:randomterraingeneration', radius, iterations, target_type, user_cell_size)
```

**See also**

**Raster calculator**

**Description**

<put algorithm description here>

### Parameters

**Main input layer** [raster] <put parameter description here>

**Additional layers** [multipleinput: rasters] Optional.

<put parameter description here>

**Formula** [string] <put parameter description here>

Default: *(not set)*

### Outputs

**Result** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:rastercalculator', grids, xgrids, formula, result)
```

### See also

.

## 18.7.4 Grid filter

### Dtm filter (slope-based)

#### Description

<put algorithm description here>

#### Parameters

**Grid to filter** [raster] <put parameter description here>

**Search Radius** [number] <put parameter description here>

Default: 2

**Approx. Terrain Slope** [number] <put parameter description here>

Default: 30.0

**Use Confidence Interval** [boolean] <put parameter description here>

Default: *True*

#### Outputs

**Bare Earth** [raster] <put output description here>

**Removed Objects** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:dtmfilterslopebased', input, radius, terrainslope, stddev, ground, nongro
```

### See also

### Filter clumps

#### Description

<put algorithm description here>

#### Parameters

**Input Grid** [raster] <put parameter description here>

**Min. Size** [number] <put parameter description here>

Default: 10

#### Outputs

**Filtered Grid** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:filterclumps', grid, threshold, output)
```

### See also

### Gaussian filter

#### Description

<put algorithm description here>

#### Parameters

**Grid** [raster] <put parameter description here>

**Standard Deviation** [number] <put parameter description here>

Default: 1

**Search Mode** [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

**Search Radius** [number] <put parameter description here>

Default: 3

## Outputs

**Filtered Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:gaussianfilter', input, sigma, mode, radius, result)
```

## See also

## Laplacian filter

### Description

<put algorithm description here>

### Parameters

**Grid [raster]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] standard kernel 1
- 1 — [1] standard kernel 2
- 2 — [2] Standard kernel 3
- 3 — [3] user defined kernel

Default: 0

**Standard Deviation (Percent of Radius) [number]** <put parameter description here>

Default: 0

**Radius [number]** <put parameter description here>

Default: 1

**Search Mode [selection]** <put parameter description here>

Options:

- 0 — [0] square
- 1 — [1] circle

Default: 0

## Outputs

**Filtered Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:laplacianfilter', input, method, sigma, radius, mode, result)
```



**See also**

**Majority filter**

**Description**

<put algorithm description here>

**Parameters**

**Grid** [**raster**] <put parameter description here>

**Search Mode** [**selection**] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

**Radius** [**number**] <put parameter description here>

Default: 1

**Threshold** [**Percent**] [**number**] <put parameter description here>

Default: 0

**Outputs**

**Filtered Grid** [**raster**] <put output description here>

**Console usage**

```
processing.runalg('saga:majorityfilter', input, mode, radius, threshold, result)
```

**See also**

**Morphological filter**

**Description**

<put algorithm description here>

**Parameters**

**Grid** [**raster**] <put parameter description here>

**Search Mode** [**selection**] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

**Radius [number]** <put parameter description here>

Default: *1*

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Dilation
- 1 — [1] Erosion
- 2 — [2] Opening
- 3 — [3] Closing

Default: *0*

## Outputs

**Filtered Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:morphologicalfilter', input, mode, radius, method, result)
```

## See also

### Multi direction lee filter

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Estimated Noise (absolute) [number]** <put parameter description here>

Default: *1.0*

**Estimated Noise (relative) [number]** <put parameter description here>

Default: *1.0*

**Weighted [boolean]** <put parameter description here>

Default: *True*

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] noise variance given as absolute value
- 1 — [1] noise variance given relative to mean standard deviation
- 2 — [2] original calculation (Ringeler)

Default: *0*

## Outputs

**Filtered Grid [raster]** <put output description here>

**Minimum Standard Deviation [raster]** <put output description here>

**Direction of Minimum Standard Deviation [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:multidirectionleefilter', input, noise_abs, noise_rel, weighted, method, ...)
```

## See also

### Rank filter

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Search Mode [selection]** <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

**Radius [number]** <put parameter description here>

Default: 1

**Rank [Percent] [number]** <put parameter description here>

Default: 50

## Outputs

**Filtered Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:rankfilter', input, mode, radius, rank, result)
```

## See also

### Simple filter

#### Description

<put algorithm description here>

### Parameters

**Grid [raster]** <put parameter description here>

**Search Mode [selection]** <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

**Filter [selection]** <put parameter description here>

Options:

- 0 — [0] Smooth
- 1 — [1] Sharpen
- 2 — [2] Edge

Default: 0

**Radius [number]** <put parameter description here>

Default: 2

### Outputs

**Filtered Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:simplefilter', input, mode, method, radius, result)
```

### See also

#### User defined filter

#### Description

<put algorithm description here>

### Parameters

**Grid [raster]** <put parameter description here>

**Filter Matrix [table]** Optional.

<put parameter description here>

**Default Filter Matrix (3x3) [fixedtable]** <put parameter description here>

### Outputs

**Filtered Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:userdefinedfilter', input, filter, filter_3x3, result)
```

## See also

.

## 18.7.5 Grid gridding

### Inverse distance weighted

#### Description

Inverse distance grid interpolation from irregular distributed points.

#### Parameters

**Points** [**vector:** **point**] <put parameter description here>

**Attribute** [**tablefield:** **any**] <put parameter description here>

**Target Grid** [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Distance Weighting** [**selection**] <put parameter description here>

Options:

- 0 — [0] inverse distance to a power
- 1 — [1] linearly decreasing within search radius
- 2 — [2] exponential weighting scheme
- 3 — [3] gaussian weighting scheme

Default: 0

**Inverse Distance Power** [**number**] <put parameter description here>

Default: 2

**Exponential and Gaussian Weighting Bandwidth** [**number**] <put parameter description here>

Default: 1

**Search Range** [**selection**] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

**Search Radius** [**number**] <put parameter description here>

Default: 100.0

**Search Mode** [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

**Number of Points** [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of points
- 1 — [1] all points

Default: 0

**Maximum Number of Points** [number] <put parameter description here>

Default: 10

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

## Outputs

**Grid** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:inversedistanceweighted', shapes, field, target, weighting, power, bandwi
```

## See also

## Kernel density estimation

### Description

<put algorithm description here>

### Parameters

**Points** [vector: point] <put parameter description here>

**Weight** [tablefield: any] <put parameter description here>

**Radius** [number] <put parameter description here>

Default: 10

**Kernel** [selection] <put parameter description here>

Options:

- 0 — [0] quartic kernel
- 1 — [1] gaussian kernel

Default: 0

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

## Outputs

**Grid** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:kerneldensityestimation', points, population, radius, kernel, target, outp
```

## See also

### Modified quadratic shepard

#### Description

<put algorithm description here>

#### Parameters

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Quadratic Neighbors** [number] <put parameter description here>

Default: 13

**Weighting Neighbors** [number] <put parameter description here>

Default: 19

**Left** [number] <put parameter description here>

Default: 0.0

**Right** [number] <put parameter description here>

Default: 0.0

**Bottom** [number] <put parameter description here>

Default: *0.0*

**Top** [number] <put parameter description here>

Default: *0.0*

**Cellsize** [number] <put parameter description here>

Default: *100.0*

## Outputs

**Grid** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:modifiedquadraticshepard', shapes, field, target, quadratic_neighbors, wei
```

## See also

## Natural neighbour

### Description

<put algorithm description here>

### Parameters

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Sibson** [boolean] <put parameter description here>

Default: *True*

**Output extent** [extent] <put parameter description here>

Default: *0,1,0,1*

**Cellsize** [number] <put parameter description here>

Default: *100.0*

## Outputs

**Grid** [raster] <put output description here>



**Console usage**

```
processing.runalg('saga:naturalneighbour', shapes, field, target, sibson, output_extent, user_size)
```

**See also**

**Nearest neighbour**

**Description**

<put algorithm description here>

**Parameters**

**Points** [**vector: point**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Target Grid** [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Output extent** [**extent**] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [**number**] <put parameter description here>

Default: 100.0

**Outputs**

**Grid** [**raster**] <put output description here>

**Console usage**

```
processing.runalg('saga:nearestneighbour', shapes, field, target, output_extent, user_size, user_size)
```

**See also**

**Shapes to grid**

**Description**

<put algorithm description here>

## Parameters

**Shapes** [vector: any] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Method for Multiple Values** [selection] <put parameter description here>

Options:

- 0 — [0] first
- 1 — [1] last
- 2 — [2] minimum
- 3 — [3] maximum
- 4 — [4] mean

Default: 0

**Method for Lines** [selection] <put parameter description here>

Options:

- 0 — [0] thin
- 1 — [1] thick

Default: 0

**Preferred Target Grid Type** [selection] <put parameter description here>

Options:

- 0 — [0] Integer (1 byte)
- 1 — [1] Integer (2 byte)
- 2 — [2] Integer (4 byte)
- 3 — [3] Floating Point (4 byte)
- 4 — [4] Floating Point (8 byte)

Default: 0

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

## Outputs

**Grid** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:shapestogrid', input, field, multiple, line_type, grid_type, output_extent)
```

**See also**

**Triangulation**

**Description**

<put algorithm description here>

**Parameters**

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

**Outputs**

**Grid** [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:triangulation', shapes, field, target, output_extent, user_size, user_grid)
```

**See also**

.

**18.7.6 Grid spline**

**B-spline approximation**

**Description**

<put algorithm description here>

### Parameters

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Resolution** [number] <put parameter description here>

Default: 1.0

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

### Outputs

**Grid** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:bsplineapproximation', shapes, field, target, level, output_extent, user_
```

### See also

## Cubic spline approximation

### Description

<put algorithm description here>

### Parameters

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Minimal Number of Points** [number] <put parameter description here>

Default: 3

**Maximal Number of Points** [number] <put parameter description here>

Default: 20

**Points per Square [number]** <put parameter description here>

Default: 5

**Tolerance [number]** <put parameter description here>

Default: 140.0

**Output extent [extent]** <put parameter description here>

Default: 0,1,0,1

**Cellsize [number]** <put parameter description here>

Default: 100.0

## Outputs

**Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:cubicsplineapproximation', shapes, field, target, npmin, npmax, nppc, k, c
```

## See also

### Multilevel b-spline interpolation (from grid)

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Target Grid [selection]** <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] without B-spline refinement
- 1 — [1] with B-spline refinement

Default: 0

**Threshold Error [number]** <put parameter description here>

Default: 0.0001

**Maximum Level [number]** <put parameter description here>

Default: 11.0

**Data Type [selection]** <put parameter description here>

Options:

- 0 — [0] same as input grid
- 1 — [1] floating point

Default: 0

**Output extent [extent]** <put parameter description here>

Default: 0,1,0,1

**Cellsize [number]** <put parameter description here>

Default: 100.0

## Outputs

**Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:multilevelbsplineinterpolationfromgrid', gridpoints, target, method, epsi
```

## See also

## Multilevel b-spline interpolation

### Description

<put algorithm description here>

### Parameters

**Points [vector: point]** <put parameter description here>

**Attribute [tablefield: any]** <put parameter description here>

**Target Grid [selection]** <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] without B-spline refinement
- 1 — [1] with B-spline refinement

Default: 0

**Threshold Error [number]** <put parameter description here>

Default: 0.0001

**Maximum Level [number]** <put parameter description here>

Default: 11.0

**Output extent [extent]** <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: *100.0*

### Outputs

**Grid** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:multilevelbsplineinterpolation', shapes, field, target, method, epsilon, .
```

### See also

### Thin plate spline (global)

#### Description

<put algorithm description here>

#### Parameters

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Regularisation** [number] <put parameter description here>

Default: *0.0*

**Output extent** [extent] <put parameter description here>

Default: *0,1,0,1*

**Cellsize** [number] <put parameter description here>

Default: *100.0*

### Outputs

**Grid** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:thinplatesplineglobal', shapes, field, target, regul, output_extent, user.
```

## See also

## Thin plate spline (local)

### Description

<put algorithm description here>

### Parameters

**Points** [**vector: point**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Target Grid** [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Regularisation** [**number**] <put parameter description here>

Default: 0.0001

**Search Radius** [**number**] <put parameter description here>

Default: 100.0

**Search Mode** [**selection**] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

**Points Selection** [**selection**] <put parameter description here>

Options:

- 0 — [0] all points in search radius
- 1 — [1] maximum number of points

Default: 0

**Maximum Number of Points** [**number**] <put parameter description here>

Default: 10

**Output extent** [**extent**] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [**number**] <put parameter description here>

Default: 100.0

### Outputs

**Grid** [**raster**] <put output description here>



**Console usage**

```
processing.runalg('saga:thinplatesplinelocal', shapes, field, target, regul, radius, mode, select,
```

**See also**

**Thin plate spline (tin)**

**Description**

<put algorithm description here>

**Parameters**

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Regularisation** [number] <put parameter description here>

Default: 0.0

**Neighbourhood** [selection] <put parameter description here>

Options:

- 0 — [0] immediate
- 1 — [1] level 1
- 2 — [2] level 2

Default: 0

**Add Frame** [boolean] <put parameter description here>

Default: True

**Output extent** [extent] <put parameter description here>

Default: 0,1,0,1

**Cellsize** [number] <put parameter description here>

Default: 100.0

**Outputs**

**Grid** [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:thinplatesplinetin', shapes, field, target, regul, level, frame, output_e
```

See also

.

## 18.7.7 Grid tools

### Aggregate

#### Description

<put algorithm description here>

#### Parameters

**Grid** [**raster**] <put parameter description here>

**Aggregation Size** [**number**] <put parameter description here>

Default: 3

**Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] Sum
- 1 — [1] Min
- 2 — [2] Max

Default: 0

#### Outputs

#### Console usage

```
processing.runalg('saga:aggregate', input, size, method)
```

See also

### Change grid values

#### Description

<put algorithm description here>

#### Parameters

**Grid** [**raster**] <put parameter description here>

**Replace Condition** [**selection**] <put parameter description here>

Options:

- 0 — [0] Grid value equals low value
- 1 — [1] Low value < grid value < high value

- 2 — [2] Low value  $\leq$  grid value  $<$  high value

Default: 0

**Lookup Table [fixedtable]** <put parameter description here>

### Outputs

**Changed Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:changegridvalues', grid_in, method, lookup, grid_out)
```

### See also

### Close gaps

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Mask [raster]** Optional.

<put parameter description here>

**Tension Threshold [number]** <put parameter description here>

Default: 0.1

### Outputs

**Changed Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:closegaps', input, mask, threshold, result)
```

### See also

### Close gaps with spline

#### Description

<put algorithm description here>

### Parameters

**Grid [raster]** <put parameter description here>

**Mask [raster]** Optional.

<put parameter description here>

**Only Process Gaps with Less Cells [number]** <put parameter description here>

Default: *0*

**Maximum Points [number]** <put parameter description here>

Default: *1000*

**Number of Points for Local Interpolation [number]** <put parameter description here>

Default: *10*

**Extended Neighbourhood [boolean]** <put parameter description here>

Default: *True*

**Neighbourhood [selection]** <put parameter description here>

Options:

- 0 — [0] Neumann
- 1 — [1] Moore

Default: *0*

**Radius (Cells) [number]** <put parameter description here>

Default: *0*

**Relaxation [number]** <put parameter description here>

Default: *0.0*

### Outputs

**Closed Gaps Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:closegapswithspline', grid, mask, maxgapcells, maxpoints, localpoints, ex
```

### See also

#### Close one cell gaps

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

## Outputs

**Changed Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:closeonecellgaps', input, result)
```

## See also

## Convert data storage type

### Description

<put algorithm description here>

### Parameters

**Grid [raster]** <put parameter description here>

**Data storage type [selection]** <put parameter description here>

Options:

- 0 — [0] bit
- 1 — [1] unsigned 1 byte integer
- 2 — [2] signed 1 byte integer
- 3 — [3] unsigned 2 byte integer
- 4 — [4] signed 2 byte integer
- 5 — [5] unsigned 4 byte integer
- 6 — [6] signed 4 byte integer
- 7 — [7] 4 byte floating point number
- 8 — [8] 8 byte floating point number

Default: 0

## Outputs

**Converted Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:convertdatastoragetype', input, type, output)
```

**See also**

**Crop to data**

**Description**

<put algorithm description here>

**Parameters**

**Input layer** [raster] <put parameter description here>

**Outputs**

**Cropped layer** [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:croptodata', input, output)
```

**See also**

**Grid buffer**

**Description**

<put algorithm description here>

**Parameters**

**Features Grid** [raster] <put parameter description here>

**Distance** [number] <put parameter description here>

Default: *1000*

**Buffer Distance** [selection] <put parameter description here>

Options:

- 0 — [0] Fixed
- 1 — [1] Cell value

Default: *0*

**Outputs**

**Buffer Grid** [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:gridbuffer', features, dist, buffertype, buffer)
```

**See also**

**Grid masking**

**Description**

<put algorithm description here>

**Parameters**

**Grid [raster]** <put parameter description here>

**Mask [raster]** <put parameter description here>

**Outputs**

**Masked Grid [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:gridmasking', grid, mask, masked)
```

**See also**

**Grid orientation**

**Description**

<put algorithm description here>

**Parameters**

**Grid [raster]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Copy
- 1 — [1] Flip
- 2 — [2] Mirror
- 3 — [3] Invert

Default: 0

**Outputs**

**Changed Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:gridorientation', input, method, result)
```

### See also

### Grid proximity buffer

#### Description

<put algorithm description here>

#### Parameters

**Source Grid** [raster] <put parameter description here>

**Buffer distance** [number] <put parameter description here>

Default: *500.0*

**Equidistance** [number] <put parameter description here>

Default: *100.0*

#### Outputs

**Distance Grid** [raster] <put output description here>

**Allocation Grid** [raster] <put output description here>

**Buffer Grid** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:gridproximitybuffer', source, dist, ival, distance, alloc, buffer)
```

### See also

### Grid shrink/expand

#### Description

<put algorithm description here>

#### Parameters

**Grid** [raster] <put parameter description here>

**Operation** [selection] <put parameter description here>

Options:

- 0 — [0] Shrink
- 1 — [1] Expand



Default: 0

**Search Mode** [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

**Radius** [number] <put parameter description here>

Default: 1

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] min
- 1 — [1] max
- 2 — [2] mean
- 3 — [3] majority

Default: 0

## Outputs

**Result Grid** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:gridshrinkexpand', input, operation, mode, radius, method_expand, result)
```

## See also

## Invert data/no-data

### Description

<put algorithm description here>

### Parameters

**Grid** [raster] <put parameter description here>

### Outputs

**Result** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:invertdatanodata', input, output)
```

See also

## Merge raster layers

### Description

<put algorithm description here>

### Parameters

**Grids to Merge** [**multipleinput: rasters**] <put parameter description here>

**Preferred data storage type** [**selection**] <put parameter description here>

Options:

- 0 — [0] 1 bit
- 1 — [1] 1 byte unsigned integer
- 2 — [2] 1 byte signed integer
- 3 — [3] 2 byte unsigned integer
- 4 — [4] 2 byte signed integer
- 5 — [5] 4 byte unsigned integer
- 6 — [6] 4 byte signed integer
- 7 — [7] 4 byte floating point
- 8 — [8] 8 byte floating point

Default: 0

**Interpolation** [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

**Overlapping Cells** [**selection**] <put parameter description here>

Options:

- 0 — [0] mean value
- 1 — [1] first value in order of grid list

Default: 0

### Outputs

**Merged Grid** [**raster**] <put output description here>

### Console usage

```
processing.runalg('saga:mergerasterlayers', grids, type, interpol, overlap, merged)
```

### See also

### Patching

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Patch Grid [raster]** <put parameter description here>

**Interpolation Method [selection]** <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

#### Outputs

**Completed Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:patching', original, additional, interpolation, completed)
```

### See also

### Proximity grid

#### Description

<put algorithm description here>

#### Parameters

**Features [raster]** <put parameter description here>

## Outputs

**Distance** [raster] <put output description here>

**Direction** [raster] <put output description here>

**Allocation** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:proximitygrid', features, distance, direction, allocation)
```

## See also

### Reclassify grid values

#### Description

<put algorithm description here>

#### Parameters

**Grid** [raster] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] single
- 1 — [1] range
- 2 — [2] simple table

Default: 0

**old value (for single value change)** [number] <put parameter description here>

Default: 0.0

**new value (for single value change)** [number] <put parameter description here>

Default: 1.0

**operator (for single value change)** [selection] <put parameter description here>

Options:

- 0 — [0] =
- 1 — [1] <
- 2 — [2] <=
- 3 — [3] >=
- 4 — [4] >

Default: 0

**minimum value (for range)** [number] <put parameter description here>

Default: 0.0

**maximum value (for range) [number]** <put parameter description here>

Default: *1.0*

**new value (for range) [number]** <put parameter description here>

Default: *2.0*

**operator (for range) [selection]** <put parameter description here>

Options:

- 0 — [0] <=
- 1 — [1] <

Default: *0*

**Lookup Table [fixedtable]** <put parameter description here>

**operator (for table) [selection]** <put parameter description here>

Options:

- 0 — [0] min <= value < max
- 1 — [1] min <= value <= max
- 2 — [2] min < value <= max
- 3 — [3] min < value < max

Default: *0*

**replace no data values [boolean]** <put parameter description here>

Default: *True*

**new value for no data values [number]** <put parameter description here>

Default: *0.0*

**replace other values [boolean]** <put parameter description here>

Default: *True*

**new value for other values [number]** <put parameter description here>

Default: *0.0*

## Outputs

**Reclassified Grid [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:reclassifygridvalues', input, method, old, new, soperator, min, max, rnew,
```

## See also

## Resampling

## Description

<put algorithm description here>

## Parameters

**Grid [raster]** <put parameter description here>

**Preserve Data Type [boolean]** <put parameter description here>

Default: *True*

**Target Grid [selection]** <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Interpolation Method (Scale Up) [selection]** <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation
- 5 — [5] Mean Value
- 6 — [6] Mean Value (cell area weighted)
- 7 — [7] Minimum Value
- 8 — [8] Maximum Value
- 9 — [9] Majority

Default: *0*

**Interpolation Method (Scale Down) [selection]** <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

**Output extent [extent]** <put parameter description here>

Default: *0,1,0,1*

**Cellsize [number]** <put parameter description here>

Default: *100.0*

## Outputs

**Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:resampling', input, keep_type, target, scale_up_method, scale_down_method)
```

### See also

### Sort grid

#### Description

<put algorithm description here>

#### Parameters

**Input Grid** [raster] <put parameter description here>

**Down sort** [boolean] <put parameter description here>

Default: *True*

#### Outputs

**Sorted Grid** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:sortgrid', grid, down, output)
```

### See also

### Split RGB bands

#### Description

<put algorithm description here>

#### Parameters

**Input layer** [raster] <put parameter description here>

#### Outputs

**Output R band layer** [raster] <put output description here>

**Output G band layer** [raster] <put output description here>

**Output B band layer** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:splitrgbbands', input, r, g, b)
```

### See also

### Threshold buffer

#### Description

<put algorithm description here>

#### Parameters

**Features Grid [raster]** <put parameter description here>

**Value Grid [raster]** <put parameter description here>

**Threshold Grid [raster]** Optional.

<put parameter description here>

**Threshold [number]** <put parameter description here>

Default: *0.0*

**Threshold Type [selection]** <put parameter description here>

Options:

- 0 — [0] Absolute
- 1 — [1] Relative from cell value

Default: *0*

#### Outputs

**Buffer Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:thresholdbuffer', features, value, thresholdgrid, threshold, thresholdtype)
```

### See also

.

## 18.7.8 Grid visualization

### Histogram surface

#### Description

<put algorithm description here>



## Parameters

**Grid** [**raster**] <put parameter description here>

**Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] rows
- 1 — [1] columns
- 2 — [2] circle

Default: 0

## Outputs

**Histogram** [**raster**] <put output description here>

## Console usage

```
processing.runalg('saga:histogramsurface', grid, method, hist)
```

## See also

## Rgb composite

### Description

<put algorithm description here>

## Parameters

**R** [**raster**] <put parameter description here>

**G** [**raster**] <put parameter description here>

**B** [**raster**] <put parameter description here>

**Method for R value** [**selection**] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255
- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

**Method for G value** [**selection**] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255

- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

**Method for B value [selection]** <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255
- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

**Rescale Range for RED min [number]** <put parameter description here>

Default: 0

**Rescale Range for RED max [number]** <put parameter description here>

Default: 255

**Percentiles Range for RED max [number]** <put parameter description here>

Default: 1

**Percentiles Range for RED max [number]** <put parameter description here>

Default: 99

**Percentage of standard deviation for RED [number]** <put parameter description here>

Default: 150.0

**Rescale Range for GREEN min [number]** <put parameter description here>

Default: 0

**Rescale Range for GREEN max [number]** <put parameter description here>

Default: 255

**Percentiles Range for GREEN max [number]** <put parameter description here>

Default: 1

**Percentiles Range for GREEN max [number]** <put parameter description here>

Default: 99

**Percentage of standard deviation for GREEN [number]** <put parameter description here>

Default: 150.0

**Rescale Range for BLUE min [number]** <put parameter description here>

Default: 0

**Rescale Range for BLUE max [number]** <put parameter description here>

Default: 255

**Percentiles Range for BLUE max [number]** <put parameter description here>

Default: 1

**Percentiles Range for BLUE max [number]** <put parameter description here>

Default: *99*

**Percentage of standard deviation for BLUE [number]** <put parameter description here>

Default: *150.0*

## Outputs

**Output RGB [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:rgbcomposite', grid_r, grid_g, grid_b, r_method, g_method, b_method, r_ra
```

## See also

.

## 18.7.9 Imagery classification

### Change detection

#### Description

<put algorithm description here>

#### Parameters

**Initial State [raster]** <put parameter description here>

**Look-up Table [table]** Optional.

<put parameter description here>

**Value [tablefield: any]** <put parameter description here>

**Value (Maximum) [tablefield: any]** <put parameter description here>

**Name [tablefield: any]** <put parameter description here>

**Final State [raster]** <put parameter description here>

**Look-up Table [table]** Optional.

<put parameter description here>

**Value [tablefield: any]** <put parameter description here>

**Value (Maximum) [tablefield: any]** <put parameter description here>

**Name [tablefield: any]** <put parameter description here>

**Report Unchanged Classes [boolean]** <put parameter description here>

Default: *True*

**Output as... [selection]** <put parameter description here>

Options:

- 0 — [0] cells
- 1 — [1] percent
- 2 — [2] area

Default: 0

### Outputs

**Changes [raster]** <put output description here>

**Changes [table]** <put output description here>

### Console usage

```
processing.runalg('saga:changedetection', initial, ini_lut, ini_lut_min, ini_lut_max, ini_lut_name,
```

### See also

## Cluster analysis for grids

### Description

<put algorithm description here>

### Parameters

**Grids [multipleinput: rasters]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Iterative Minimum Distance (Forgy 1965)
- 1 — [1] Hill-Climbing (Rubin 1967)
- 2 — [2] Combined Minimum Distance / Hillclimbing

Default: 0

**Clusters [number]** <put parameter description here>

Default: 5

**Normalise [boolean]** <put parameter description here>

Default: *True*

**Old Version [boolean]** <put parameter description here>

Default: *True*

### Outputs

**Clusters [raster]** <put output description here>

**Statistics [table]** <put output description here>

## Console usage

```
processing.runalg('saga:clusteranalysisforgrids', grids, method, ncluster, normalise, oldversion,
```

## See also

## Supervised classification

### Description

<put algorithm description here>

### Parameters

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Training Areas** [**vector: polygon**] <put parameter description here>

**Class Identifier** [**tablefield: any**] <put parameter description here>

**Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] Binary Encoding
- 1 — [1] Parallelepiped
- 2 — [2] Minimum Distance
- 3 — [3] Mahalanobis Distance
- 4 — [4] Maximum Likelihood
- 5 — [5] Spectral Angle Mapping
- 6 — [6] Winner Takes All

Default: *0*

**Normalise** [**boolean**] <put parameter description here>

Default: *True*

**Distance Threshold** [**number**] <put parameter description here>

Default: *0.0*

**Probability Threshold (Percent)** [**number**] <put parameter description here>

Default: *0.0*

**Probability Reference** [**selection**] <put parameter description here>

Options:

- 0 — [0] absolute
- 1 — [1] relative

Default: *0*

**Spectral Angle Threshold (Degree)** [**number**] <put parameter description here>

Default: *0.0*

### Outputs

**Class Information** [table] <put output description here>

**Classification** [raster] <put output description here>

**Quality** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:supervisedclassification', grids, roi, roi_id, method, normalise, thresho
```

### See also

.

## 18.7.10 Imagery RGA

### Fast region growing algorithm

#### Description

<put algorithm description here>

#### Parameters

**Input Grids** [multipleinput: rasters] <put parameter description here>

**Seeds Grid** [raster] <put parameter description here>

**Smooth Rep** [raster] Optional.

<put parameter description here>

#### Outputs

**Segmente** [raster] <put output description here>

**Mean** [raster] <put output description here>

#### Console usage

```
processing.runalg('saga:fastregiongrowingalgorithm', input, start, rep, result, mean)
```

### See also

.

## 18.7.11 Imagery segmentation

### Grid skeletonization

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] Hilditch's Algorithm
- 2 — [2] Channel Skeleton

Default: 0

**Initialisation [selection]** <put parameter description here>

Options:

- 0 — [0] Less than
- 1 — [1] Greater than

Default: 0

**Threshold (Init.) [number]** <put parameter description here>

Default: 0.0

**Convergence [number]** <put parameter description here>

Default: 3.0

#### Outputs

**Skeleton [raster]** <put output description here>

**Skeleton [vector]** <put output description here>

#### Console usage

```
processing.runalg('saga:gridskeletonization', input, method, init_method, init_threshold, converg
```

#### See also

### Seed generation

#### Description

<put algorithm description here>

## Parameters

**Features** [**multipleinput: rasters**] <put parameter description here>

**Bandwidth (Cells)** [**number**] <put parameter description here>

Default: 2

**Type of Surface** [**selection**] <put parameter description here>

Options:

- 0 — [0] smoothed surface
- 1 — [1] variance (a)
- 2 — [2] variance (b)

Default: 0

**Extraction of...** [**selection**] <put parameter description here>

Options:

- 0 — [0] minima
- 1 — [1] maxima
- 2 — [2] minima and maxima

Default: 0

**Feature Aggregation** [**selection**] <put parameter description here>

Options:

- 0 — [0] additive
- 1 — [1] multiplicative

Default: 0

**Normalized** [**boolean**] <put parameter description here>

Default: *True*

## Outputs

**Surface** [**raster**] <put output description here>

**Seeds Grid** [**raster**] <put output description here>

**Seeds** [**vector**] <put output description here>

## Console usage

```
processing.runalg('saga:seedgeneration', grids, factor, type_surface, type_seeds, type_merge, norm
```

## See also

### Simple region growing

#### Description

<put algorithm description here>



## Parameters

**Seeds** [raster] <put parameter description here>

**Features** [multipleinput: rasters] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] feature space and position
- 1 — [1] feature space

Default: 0

**Neighbourhood** [selection] <put parameter description here>

Options:

- 0 — [0] 4 (von Neumann)
- 1 — [1] 8 (Moore)

Default: 0

**Variance in Feature Space** [number] <put parameter description here>

Default: 1.0

**Variance in Position Space** [number] <put parameter description here>

Default: 1.0

**Threshold - Similarity** [number] <put parameter description here>

Default: 0.0

**Refresh** [boolean] <put parameter description here>

Default: *True*

**Leaf Size (for Speed Optimisation)** [number] <put parameter description here>

Default: 256

## Outputs

**Segments** [raster] <put output description here>

**Similarity** [raster] <put output description here>

**Seeds** [table] <put output description here>

## Console usage

```
processing.runalg('saga:simpleregiongrowing', seeds, features, method, neighbour, sig_1, sig_2, t
```

## See also

### Watershed segmentation

#### Description

<put algorithm description here>

## Parameters

**Grid** [raster] <put parameter description here>

**Output** [selection] <put parameter description here>

Options:

- 0 — [0] Seed Value
- 1 — [1] Segment ID

Default: 0

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Minima
- 1 — [1] Maxima

Default: 0

**Join Segments based on Threshold Value** [selection] <put parameter description here>

Options:

- 0 — [0] do not join
- 1 — [1] seed to saddle difference
- 2 — [2] seeds difference

Default: 0

**Threshold** [number] <put parameter description here>

Default: 0

**Allow Edge Pixels to be Seeds** [boolean] <put parameter description here>

Default: *True*

**Borders** [boolean] <put parameter description here>

Default: *True*

## Outputs

**Segments** [raster] <put output description here>

**Seed Points** [vector] <put output description here>

**Borders** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:watershedsegmentation', grid, output, down, join, threshold, edge, borders)
```

## See also

.

## 18.7.12 Imagery tools

### Vegetation index[distance based]

#### Description

<put algorithm description here>

#### Parameters

**Near Infrared Band** [raster] <put parameter description here>

**Red Band** [raster] <put parameter description here>

**Slope of the soil line** [number] <put parameter description here>

Default: *0.0*

**Intercept of the soil line** [number] <put parameter description here>

Default: *0.0*

#### Outputs

**PVI (Richardson and Wiegand)** [raster] <put output description here>

**PVI (Perry & Lautenschlager)** [raster] <put output description here>

**PVI (Walther & Shabaani)** [raster] <put output description here>

**PVI (Qi, et al)** [raster] <put output description here>

#### Console usage

```
processing.runalg('saga:vegetationindexdistancebased', nir, red, slope, intercept, pvi, pvi1, pvi2)
```

#### See also

### Vegetation index[slope based]

#### Description

<put algorithm description here>

#### Parameters

**Near Infrared Band** [raster] <put parameter description here>

**Red Band** [raster] <put parameter description here>

## Outputs

**Normalized Difference Vegetation Index [raster]** <put output description here>

**Ratio Vegetation Index [raster]** <put output description here>

**Transformed Vegetation Index [raster]** <put output description here>

**Corrected Transformed Vegetation Index [raster]** <put output description here>

**Thiam's Transformed Vegetation Index [raster]** <put output description here>

**Normalized Ratio Vegetation Index [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:vegetationindexslopebased', nir, red, ndvi, ratio, tvi, ctvi, ttvi, nratio)
```

## See also

.

## 18.7.13 Kriging

### Ordinary kriging (global)

#### Description

<put algorithm description here>

#### Parameters

**Points [vector: point]** <put parameter description here>

**Attribute [tablefield: any]** <put parameter description here>

**Create Variance Grid [boolean]** <put parameter description here>

Default: *True*

**Target Grid [selection]** <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Variogram Model [selection]** <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

**Block Kriging [boolean]** <put parameter description here>

Default: *True*

**Block Size [number]** <put parameter description here>

Default: 100

**Logarithmic Transformation [boolean]** <put parameter description here>

Default: *True*

**Nugget [number]** <put parameter description here>

Default: 0.0

**Sill [number]** <put parameter description here>

Default: 0.0

**Range [number]** <put parameter description here>

Default: 0.0

**Linear Regression [number]** <put parameter description here>

Default: 1.0

**Exponential Regression [number]** <put parameter description here>

Default: 0.1

**Power Function - A [number]** <put parameter description here>

Default: 1.0

**Power Function - B [number]** <put parameter description here>

Default: 0.5

**Grid Size [number]** <put parameter description here>

Default: 1.0

**Fit Extent [boolean]** <put parameter description here>

Default: *True*

**Output extent [extent]** <put parameter description here>

Default: 0,1,0,1

### Outputs

**Grid [raster]** <put output description here>

**Variance [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:ordinarykrigingglobal', shapes, field, bvariance, target, model, block, d
```

See also

## Ordinary kriging

### Description

<put algorithm description here>

### Parameters

**Points** [**vector: point**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Create Variance Grid** [**boolean**] <put parameter description here>

Default: *True*

**Target Grid** [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Variogram Model** [**selection**] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: *0*

**Block Kriging** [**boolean**] <put parameter description here>

Default: *True*

**Block Size** [**number**] <put parameter description here>

Default: *100*

**Logarithmic Transformation** [**boolean**] <put parameter description here>

Default: *True*

**Nugget** [**number**] <put parameter description here>

Default: *0.0*

**Sill** [**number**] <put parameter description here>

Default: *10.0*

**Range** [**number**] <put parameter description here>

Default: *100.0*

**Linear Regression** [**number**] <put parameter description here>

Default: *1.0*

**Exponential Regression [number]** <put parameter description here>

Default: *0.1*

**Power Function - A [number]** <put parameter description here>

Default: *1*

**Power Function - B [number]** <put parameter description here>

Default: *0.5*

**Maximum Search Radius (map units) [number]** <put parameter description here>

Default: *1000.0*

**Min. Number of m\_Points [number]** <put parameter description here>

Default: *4*

**Max. Number of m\_Points [number]** <put parameter description here>

Default: *20*

**Grid Size [number]** <put parameter description here>

Default: *1.0*

**Fit Extent [boolean]** <put parameter description here>

Default: *True*

**Output extent [extent]** <put parameter description here>

Default: *0,1,0,1*

## Outputs

**Grid [raster]** <put output description here>

**Variance [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:ordinarykriging', shapes, field, bvariance, target, model, block, dblock,
```

## See also

### Universal kriging (global)

#### Description

<put algorithm description here>

#### Parameters

**Points [vector: point]** <put parameter description here>

**Attribute [tablefield: any]** <put parameter description here>

**Create Variance Grid [boolean]** <put parameter description here>

Default: *True*

**Target Grid [selection]** <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

**Variogram Model [selection]** <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

**Block Kriging [boolean]** <put parameter description here>

Default: *True*

**Block Size [number]** <put parameter description here>

Default: 100

**Logarithmic Transformation [boolean]** <put parameter description here>

Default: *True*

**Nugget [number]** <put parameter description here>

Default: 0.0

**Sill [number]** <put parameter description here>

Default: 0.0

**Range [number]** <put parameter description here>

Default: 0.0

**Linear Regression [number]** <put parameter description here>

Default: 1

**Exponential Regression [number]** <put parameter description here>

Default: 0.5

**Power Function - A [number]** <put parameter description here>

Default: 1.0

**Power Function - B [number]** <put parameter description here>

Default: 0.1

**Grids [multipleinput: rasters]** <put parameter description here>

**Grid Interpolation [selection]** <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation



- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

**Grid Size** [number] <put parameter description here>

Default: *1.0*

**Fit Extent** [boolean] <put parameter description here>

Default: *True*

**Output extent** [extent] <put parameter description here>

Default: *0,1,0,1*

## Outputs

**Grid** [raster] <put output description here>

**Variance** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:universalkrigingglobal', shapes, field, bvariance, target, model, block, ...)
```

## See also

## Universal kriging

### Description

<put algorithm description here>

### Parameters

**Points** [vector: point] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Create Variance Grid** [boolean] <put parameter description here>

Default: *True*

**Target Grid** [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

**Variogram Model** [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression

- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

**Block Kriging [boolean]** <put parameter description here>

Default: *True*

**Block Size [number]** <put parameter description here>

Default: 100

**Logarithmic Transformation [boolean]** <put parameter description here>

Default: *True*

**Nugget [number]** <put parameter description here>

Default: 0.0

**Sill [number]** <put parameter description here>

Default: 0.0

**Range [number]** <put parameter description here>

Default: 0.0

**Linear Regression [number]** <put parameter description here>

Default: 1.0

**Exponential Regression [number]** <put parameter description here>

Default: 0.1

**Power Function - A [number]** <put parameter description here>

Default: 1

**Power Function - B [number]** <put parameter description here>

Default: 0.5

**Grids [multipleinput: rasters]** <put parameter description here>

**Grid Interpolation [selection]** <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

**Min.Number of m\_Points [number]** <put parameter description here>

Default: 4

**Max. Number of m\_Points [number]** <put parameter description here>

Default: 20

**Maximum Search Radius (map units) [number]** <put parameter description here>

Default: 1000.0

**Grid Size** [**number**] <put parameter description here>

Default: *1.0*

**Fit Extent** [**boolean**] <put parameter description here>

Default: *True*

**Output extent** [**extent**] <put parameter description here>

Default: *0,1,0,1*

## Outputs

**Grid** [**raster**] <put output description here>

**Variance** [**raster**] <put output description here>

## Console usage

```
processing.runalg('saga:universalkriging', shapes, field, bvariance, target, model, block, dblock,
```

## See also

.

## 18.7.14 Shapes grid

### Add grid values to points

#### Description

Creates a new vector layer as a result of the union of a points layer with the interpolated value of one or more base background grid layer(s). This way, the new layer created will have a new column in the attribute table that reflects the interpolated value of the background grid.

#### Parameters

**Points** [**vector: point**] Input layer.

**Grids** [**multipleinput: rasters**] Background grid layer(s)

**Interpolation** [**selection**] interpolation method to use.

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

## Outputs

**Result** [vector] The resulting layer.

## Console usage

```
processing.runalg('saga:addgridvaluestopoints', shapes, grids, interpol, result)
```

## See also

## Add grid values to shapes

### Description

<put algorithm description here>

### Parameters

**Shapes** [vector: any] <put parameter description here>

**Grids** [multipleinput: rasters] <put parameter description here>

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

## Outputs

**Result** [vector] <put output description here>

## Console usage

```
processing.runalg('saga:addgridvaluestoshapes', shapes, grids, interpol, result)
```

## See also

## Clip grid with polygon

### Description

<put algorithm description here>

### Parameters

**Input** [raster] <put parameter description here>

**Polygons** [vector: polygon] <put parameter description here>

### Outputs

**Output** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:clipgridwithpolygon', input, polygons, output)
```

### See also

#### Contour lines from grid

##### Description

<put algorithm description here>

##### Parameters

**Grid** [raster] <put parameter description here>

**Minimum Contour Value** [number] <put parameter description here>

Default: *0.0*

**Maximum Contour Value** [number] <put parameter description here>

Default: *10000.0*

**Equidistance** [number] <put parameter description here>

Default: *100.0*

##### Outputs

**Contour Lines** [vector] <put output description here>

##### Console usage

```
processing.runalg('saga:contourlinesfromgrid', input, zmin, zmax, zstep, contour)
```

### See also

#### Gradient vectors from directional components

##### Description

<put algorithm description here>

### Parameters

**X Component** [raster] <put parameter description here>

**Y Component** [raster] <put parameter description here>

**Step** [number] <put parameter description here>

Default: *1*

**Size Range Min** [number] <put parameter description here>

Default: *25.0*

**Size Range Max** [number] <put parameter description here>

Default: *100.0*

**Aggregation** [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: *0*

**Style** [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: *0*

### Outputs

**Gradient Vectors** [vector] <put output description here>

### Console usage

```
processing.runalg('saga:gradientvectorsfromdirectionalcomponents', x, y, step, size_min, size_max)
```

### See also

#### Gradient vectors from direction and length

#### Description

<put algorithm description here>

#### Parameters

**Direction** [raster] <put parameter description here>

**Length** [raster] <put parameter description here>

**Step** [number] <put parameter description here>

Default: 1

**Size Range Min** [number] <put parameter description here>

Default: 25.0

**Size Range Max** [number] <put parameter description here>

Default: 100.0

**Aggregation** [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: 0

**Style** [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: 0

## Outputs

**Gradient Vectors** [vector] <put output description here>

## Console usage

```
processing.runalg('saga:gradientvectorsfromdirectionandlength', dir, len, step, size_min, size_max)
```

## See also

### Gradient vectors from surface

#### Description

<put algorithm description here>

#### Parameters

**Surface** [raster] <put parameter description here>

**Step** [number] <put parameter description here>

Default: 1

**Size Range Min** [number] <put parameter description here>

Default: 25.0

**Size Range Max** [number] <put parameter description here>

Default: 100.0

**Aggregation** [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: 0

**Style** [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: 0

## Outputs

**Gradient Vectors** [vector] <put output description here>

## Console usage

```
processing.runalg('saga:gradientvectorsfromsurface', surface, step, size_min, size_max, aggr, sty
```

## See also

## Grid statistics for polygons

### Description

<put algorithm description here>

### Parameters

**Grids** [multipleinput: rasters] <put parameter description here>

**Polygons** [vector: polygon] <put parameter description here>

**Number of Cells** [boolean] <put parameter description here>

Default: *True*

**Minimum** [boolean] <put parameter description here>

Default: *True*

**Maximum** [boolean] <put parameter description here>

Default: *True*

**Range** [boolean] <put parameter description here>

Default: *True*

**Sum** [boolean] <put parameter description here>

Default: *True*



**Mean [boolean]** <put parameter description here>

Default: *True*

**Variance [boolean]** <put parameter description here>

Default: *True*

**Standard Deviation [boolean]** <put parameter description here>

Default: *True*

**Quantiles [number]** <put parameter description here>

Default: *0*

### Outputs

**Statistics [vector]** <put output description here>

### Console usage

```
processing.runalg('saga:gridstatisticsforpolygons', grids, polygons, count, min, max, range, sum,
```

### See also

#### Grid values to points (randomly)

##### Description

<put algorithm description here>

##### Parameters

**Grid [raster]** <put parameter description here>

**Frequency [number]** <put parameter description here>

Default: *100*

##### Outputs

**Points [vector]** <put output description here>

##### Console usage

```
processing.runalg('saga:gridvaluestopointsrandomly', grid, freq, points)
```

### See also

#### Grid values to points

##### Description

<put algorithm description here>

### Parameters

**Grids** [**multipleinput: rasters**] <put parameter description here>

**Polygons** [**vector: any**] Optional.

<put parameter description here>

**Exclude NoData Cells** [**boolean**] <put parameter description here>

Default: *True*

**Type** [**selection**] <put parameter description here>

Options:

- 0 — [0] nodes
- 1 — [1] cells

Default: *0*

### Outputs

**Shapes** [**vector**] <put output description here>

### Console usage

```
processing.runalg('saga:gridvaluestopoints', grids, polygons, nodata, type, shapes)
```

### See also

### Local minima and maxima

#### Description

<put algorithm description here>

#### Parameters

**Grid** [**raster**] <put parameter description here>

#### Outputs

**Minima** [**vector**] <put output description here>

**Maxima** [**vector**] <put output description here>

#### Console usage

```
processing.runalg('saga:localminimaandmaxima', grid, minima, maxima)
```

**See also**

**Vectorising grid classes**

**Description**

<put algorithm description here>

**Parameters**

**Grid [raster]** <put parameter description here>

**Class Selection [selection]** <put parameter description here>

Options:

- 0 — [0] one single class specified by class identifier
- 1 — [1] all classes

Default: 0

**Class Identifier [number]** <put parameter description here>

Default: 0

**Vectorised class as... [selection]** <put parameter description here>

Options:

- 0 — [0] one single (multi-)polygon object
- 1 — [1] each island as separated polygon

Default: 0

**Outputs**

**Polygons [vector]** <put output description here>

**Console usage**

```
processing.runalg('saga:vectorisinggridclasses', grid, class_all, class_id, split, polygons)
```

**See also**

.

**18.7.15 Shapes lines**

**Convert points to line(s)**

**Description**

Converts points to lines.

### Parameters

**Points** [vector: point] Points to convert.

**Order by . . .** [tablefield: any] Lines will be ordered following this field.

**Separate by . . .** [tablefield: any] Lines will be grouped according to this field.

### Outputs

**Lines** [vector] The resulting layer.

### Console usage

```
processing.runalg('saga:convertpointstolines', points, order, separate, lines)
```

### See also

## Convert polygons to lines

### Description

Creates lines from polygons.

### Parameters

**Polygons** [vector: polygon] Layer to process.

### Outputs

**Lines** [vector] The resulting layer.

### Console usage

```
processing.runalg('saga:convertpolygonstolines', polygons, lines)
```

### See also

## Line dissolve

### Description

<put algorithm description here>

### Parameters

**Lines** [vector: any] <put parameter description here>

1. **Attribute** [tablefield: any] <put parameter description here>

2. **Attribute** [tablefield: any] <put parameter description here>

**3. Attribute [tablefield: any]** <put parameter description here>

**Dissolve...** [selection] <put parameter description here>

Options:

- 0 — [0] lines with same attribute value(s)
- 1 — [1] all lines

Default: 0

### Outputs

**Dissolved Lines [vector]** <put output description here>

### Console usage

```
processing.runalg('saga:linedissolve', lines, field_1, field_2, field_3, all, dissolved)
```

### See also

## Line-polygon intersection

### Description

<put algorithm description here>

### Parameters

**Lines [vector: line]** <put parameter description here>

**Polygons [vector: polygon]** <put parameter description here>

**Output [selection]** <put parameter description here>

Options:

- 0 — [0] one multi-line per polygon
- 1 — [1] keep original line attributes

Default: 0

### Outputs

**Intersection [vector]** <put output description here>

### Console usage

```
processing.runalg('saga:linepolygonintersection', lines, polygons, method, intersect)
```

See also

## Line properties

### Description

Calculates some information on each line of the layer.

### Parameters

**Lines** [**vector: line**] Layer to analyze.

**Number of Parts** [**boolean**] Determines whether to calculate number of segments in line.

Default: *True*

**Number of Vertices** [**boolean**] Determines whether to calculate number of vertices in line.

Default: *True*

**Length** [**boolean**] Determines whether to calculate total line length.

Default: *True*

### Outputs

**Lines with Property Attributes** [**vector**] The resulting layer.

### Console usage

```
processing.runalg('saga:lineproperties', lines, bparts, bpoints, blength, output)
```

See also

## Line simplification

### Description

Simplifies the geometry of a lines layer.

### Parameters

**Lines** [**vector: line**] Layer to process.

**Tolerance** [**number**] Simplification tolerance.

Default: *1.0*

### Outputs

**Simplified Lines** [**vector**] The resulting layer.

### Console usage

```
processing.runalg('saga:linesimplification', lines, tolerance, output)
```

### See also

.

## 18.7.16 Shapes points

### Add coordinates to points

#### Description

Adds the X and Y coordinates of feature in the attribute table of input layer.

#### Parameters

**Points** [vector: point] Input layer.

#### Outputs

**Output** [vector] Resulting layer with the updated attribute table.

#### Console usage

```
processing.runalg('saga:addcoordinatestopoints', input, output)
```

### See also

### Add polygon attributes to points

#### Description

Adds the specified field of the polygons layer to the attribute table of the points layer. The new attributes added for each point depend on the value of the background polygon layer.

#### Parameters

**Points** [vector: point] Points layer.

**Polygons** [vector: polygon] Background polygons layer.

**Attribute** [tablefield: any] Attribute of the polygons layer that will be added to the points layer.

#### Outputs

**Result** [vector] The resulting layer.

### Console usage

```
processing.runalg('saga:addpolygonattributestopoints', input, polygons, field, output)
```

### See also

### Aggregate point observations

#### Description

<put algorithm description here>

#### Parameters

**Reference Points** [**vector: any**] <put parameter description here>

**ID** [**tablefield: any**] <put parameter description here>

**Observations** [**table**] <put parameter description here>

**X** [**tablefield: any**] <put parameter description here>

**Y** [**tablefield: any**] <put parameter description here>

**Track** [**tablefield: any**] <put parameter description here>

**Date** [**tablefield: any**] <put parameter description here>

**Time** [**tablefield: any**] <put parameter description here>

**Parameter** [**tablefield: any**] <put parameter description here>

**Maximum Time Span (Seconds)** [**number**] <put parameter description here>

Default: *60.0*

**Maximum Distance** [**number**] <put parameter description here>

Default: *0.002*

#### Outputs

**Aggregated** [**table**] <put output description here>

### Console usage

```
processing.runalg('saga:aggregatepointobservations', reference, reference_id, observations, x, y,
```

### See also

### Clip points with polygons

#### Description

<put algorithm description here>



## Parameters

**Points** [**vector: point**] <put parameter description here>

**Polygons** [**vector: polygon**] <put parameter description here>

**Add Attribute to Clipped Points** [**tablefield: any**] <put parameter description here>

**Clipping Options** [**selection**] <put parameter description here>

Options:

- 0 — [0] one layer for all points
- 1 — [1] separate layer for each polygon

Default: 0

## Outputs

**Clipped Points** [**vector**] <put output description here>

## Console usage

```
processing.runalg('saga:clippointswithpolygons', points, polygons, field, method, clips)
```

## See also

### Convert lines to points

#### Description

Converts lines layer into a points.

#### Parameters

**Lines** [**vector: line**] Lines layer to convert.

**Insert Additional Points** [**boolean**] Determines whether to add additional nodes or not.

Default: *True*

**Insert Distance** [**number**] Distance between the additional points.

Default: *1.0*

#### Outputs

**Points** [**vector**] The resulting layer.

#### Console usage

```
processing.runalg('saga:convertlinestopoints', lines, add, dist, points)
```

**See also**

**Convert multipoints to points**

**Description**

<put algorithm description here>

**Parameters**

**Multipoints** [**vector: point**] <put parameter description here>

**Outputs**

**Points** [**vector**] <put output description here>

**Console usage**

```
processing.runalg('saga:convertmultipointstopoints', multipoints, points)
```

**See also**

**Convex hull**

**Description**

<put algorithm description here>

**Parameters**

**Points** [**vector: point**] <put parameter description here>

**Hull Construction** [**selection**] <put parameter description here>

Options:

- 0 — [0] one hull for all shapes
- 1 — [1] one hull per shape
- 2 — [2] one hull per shape part

Default: 0

**Outputs**

**Convex Hull** [**vector**] <put output description here>

**Minimum Bounding Box** [**vector**] <put output description here>

**Console usage**

```
processing.runalg('saga:convexhull', shapes, polypoints, hulls, boxes)
```

**See also**

**Distance matrix**

**Description**

Generates a distance matrix between each point of the input layer. A unique ID will be created in the first row of the resulting matrix (symmetric matrix), while every other cell reflects the distance between the points.

**Parameters**

**Points** [vector: point] Input layer.

**Outputs**

**Distance Matrix Table** [table] The resulting table.

**Console usage**

```
processing.runalg('saga:distancematrix', points, table)
```

**See also**

**Fit n points to shape**

**Description**

<put algorithm description here>

**Parameters**

**Shapes** [vector: polygon] <put parameter description here>

**Number of points** [number] <put parameter description here>

Default: 10

**Outputs**

**Points** [vector] <put output description here>

**Console usage**

```
processing.runalg('saga:fitnpointstoshape', shapes, numpoints, points)
```

## See also

## Points filter

### Description

<put algorithm description here>

### Parameters

**Points** [**vector: point**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Radius** [**number**] <put parameter description here>

Default: *1*

**Minimum Number of Points** [**number**] <put parameter description here>

Default: *0*

**Maximum Number of Points** [**number**] <put parameter description here>

Default: *0*

**Quadrants** [**boolean**] <put parameter description here>

Default: *True*

**Filter Criterion** [**selection**] <put parameter description here>

Options:

- 0 — [0] keep maxima (with tolerance)
- 1 — [1] keep minima (with tolerance)
- 2 — [2] remove maxima (with tolerance)
- 3 — [3] remove minima (with tolerance)
- 4 — [4] remove below percentile
- 5 — [5] remove above percentile

Default: *0*

**Tolerance** [**number**] <put parameter description here>

Default: *0.0*

**Percentile** [**number**] <put parameter description here>

Default: *50*

### Outputs

**Filtered Points** [**vector**] <put output description here>

### Console usage

```
processing.runalg('saga:pointsfilter', points, field, radius, minnum, maxnum, quadrants, method, t
```

**See also**

**Points thinning**

**Description**

<put algorithm description here>

**Parameters**

**Points** [**vector: point**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Resolution** [**number**] <put parameter description here>

Default: 1.0

**Outputs**

**Thinned Points** [**vector**] <put output description here>

**Console usage**

```
processing.runalg('saga:pointsthinning', points, field, resolution, thinned)
```

**See also**

**Remove duplicate points**

**Description**

<put algorithm description here>

**Parameters**

**Points** [**vector: any**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Point to Keep** [**selection**] <put parameter description here>

Options:

- 0 — [0] first point
- 1 — [1] last point
- 2 — [2] point with minimum attribute value
- 3 — [3] point with maximum attribute value

Default: 0

**Numeric Attribute Values** [**selection**] <put parameter description here>

Options:

- 0 — [0] take value from the point to be kept

- 1 — [1] minimum value of all duplicates
- 2 — [2] maximum value of all duplicates
- 3 — [3] mean value of all duplicates

Default: 0

### Outputs

**Result** [vector] <put output description here>

### Console usage

```
processing.runalg('saga:removeduplicatepoints', points, field, method, numeric, result)
```

### See also

### Separate points by direction

#### Description

<put algorithm description here>

#### Parameters

**Points** [vector: point] <put parameter description here>

**Number of Directions** [number] <put parameter description here>

Default: 4

**Tolerance (Degree)** [number] <put parameter description here>

Default: 5

#### Outputs

**Output** [vector] <put output description here>

#### Console usage

```
processing.runalg('saga:separatepointsbydirection', points, directions, tolerance, output)
```

### See also

.

## 18.7.17 Shapes polygons

### Convert lines to polygons

#### Description

Converts lines to polygons.

#### Parameters

**Lines** [vector: line] Lines to convert.

#### Outputs

**Polygons** [vector] The resulting layer.

#### Console usage

```
processing.runalg('saga:convertlinestopolygons', lines, polygons)
```

#### See also

### Convert polygon/line vertices to points

#### Description

Converts the line or polygon vertices into points.

#### Parameters

**Shapes** [vector: any] Layer to process.

#### Outputs

**Points** [vector] The resulting layer.

#### Console usage

```
processing.runalg('saga:convertpolygonlineverticestopoints', shapes, points)
```

#### See also

### Polygon centroids

#### Description

Calculates the centroids of polygons.

## Parameters

**Polygons** [vector: polygon] Input layer.

**Centroids for each part** [boolean] Determines whether centroids should be calculated for each part of multipart polygon or not.

Default: *True*

## Outputs

**Centroids** [vector] The resulting layer.

## Console usage

```
processing.runalg('saga:polygoncentroids', polygons, method, centroids)
```

## See also

## Polygon dissolve

### Description

<put algorithm description here>

### Parameters

**Polygons** [vector: polygon] <put parameter description here>

1. **Attribute** [tablefield: any] Optional.

<put parameter description here>

2. **Attribute** [tablefield: any] Optional.

<put parameter description here>

3. **Attribute** [tablefield: any] Optional.

<put parameter description here>

**Dissolve...** [selection] <put parameter description here>

Options:

- 0 — [0] polygons with same attribute value
- 1 — [1] all polygons
- 2 — [2] polygons with same attribute value (keep inner boundaries)
- 3 — [3] all polygons (keep inner boundaries)

Default: *0*

### Outputs

**Dissolved Polygons** [vector] <put output description here>



### Console usage

```
processing.runalg('saga:polygondissolve', polygons, field_1, field_2, field_3, dissolve, dissolve)
```

### See also

## Polygon-line intersection

### Description

<put algorithm description here>

### Parameters

**Polygons** [vector: polygon] <put parameter description here>

**Lines** [vector: line] <put parameter description here>

### Outputs

**Intersection** [vector] <put output description here>

### Console usage

```
processing.runalg('saga:polygonlineintersection', polygons, lines, intersect)
```

### See also

## Polygon parts to separate polygons

### Description

<put algorithm description here>

### Parameters

**Polygons** [vector: polygon] <put parameter description here>

**Ignore Lakes** [boolean] <put parameter description here>

Default: *True*

### Outputs

**Polygon Parts** [vector] <put output description here>

### Console usage

```
processing.runalg('saga:polygonpartstoseparatepolygons', polygons, lakes, parts)
```

See also

## Polygon properties

### Description

<put algorithm description here>

### Parameters

**Polygons** [**vector: polygon**] <put parameter description here>

**Number of Parts** [**boolean**] <put parameter description here>

Default: *True*

**Number of Vertices** [**boolean**] <put parameter description here>

Default: *True*

**Perimeter** [**boolean**] <put parameter description here>

Default: *True*

**Area** [**boolean**] <put parameter description here>

Default: *True*

### Outputs

**Polygons with Property Attributes** [**vector**] <put output description here>

### Console usage

```
processing.runalg('saga:polygonproperties', polygons, bparts, bpoints, blength, barea, output)
```

See also

## Polygon shape indices

### Description

Calculates spatial statistics for polygons. This includes:

- area
- perimeter
- perimeter / area
- perimeter / square root of the area
- maximum distance
- maximum distance / area
- maximum distance / square root of the area
- shape index

### Parameters

**Shapes** [vector: polygon] Layer to analyze.

### Outputs

**Shape Index** [vector] The resulting layer.

### Console usage

```
processing.runalg('saga:polygonshapeindices', shapes, index)
```

### See also

## Polygons to edges and nodes

### Description

Extracts boundaries and nodes of polygons in separate files.

### Parameters

**Polygons** [vector: polygon] Input layer.

### Outputs

**Edges** [vector] Resulting line layer with polygons boundaries.

**Nodes** [vector] Resulting line layer with polygons nodes.

### Console usage

```
processing.runalg('saga:polygonstoedgesandnodes', polygons, edges, nodes)
```

### See also

.

## 18.7.18 Shapes tools

### Create graticule

#### Description

Creates a grid.

## Parameters

**Extent** [vector: any] Optional.

Grid will be created according to the selected layer.

**Output extent** [extent] Extent of the grid.

Default: *0,1,0,1*

**Division Width** [number] X-axes spacing between the lines.

Default: *1.0*

**Division Height** [number] Y-axes spacing between the lines.

Default: *1.0*

**Type** [selection] Geometry type of the resulting grid.

Options:

- 0 — [0] Lines
- 1 — [1] Rectangles

Default: *0*

## Outputs

**Graticule** [vector] The resulting layer.

## Console usage

```
processing.runalg('saga:creategraticule', extent, output_extent, distx, disty, type, graticule)
```

## See also

### Cut shapes layer

#### Description

<put algorithm description here>

#### Parameters

**Vector layer to cut** [vector: any] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] completely contained
- 1 — [1] intersects
- 2 — [2] center

Default: *0*

**Cutting polygons** [vector: any] <put parameter description here>

## Outputs

**Result** [vector] <put output description here>

**Extent** [vector] <put output description here>

## Console usage

```
processing.runalg('saga:cutshapeslayer', shapes, method, polygons_polygons, cut, extent)
```

## See also

### Get shapes extents

#### Description

Creates polygons according to the extent of the input layer features.

#### Parameters

**Shapes** [vector: any] Input layer.

**Parts** [boolean] Determines whether create polygon for each feature (`True`) or just create single polygon for whole layer (`False`).

Default: *True*

#### Outputs

**Extents** [vector] The resulting layer.

## Console usage

```
processing.runalg('saga:getshapesextents', shapes, parts, extents)
```

## See also

### Merge shapes layers

#### Description

Merges two or more input layer into a unique resulting layer. You can merge together only layer of the same type (polygons with polygons, lines with lines, points with points).

The attribute table of the resulting layer will include only the attributes of the first input layer. Two additional columns will be added: one corresponding to the ID of every merged layer and the other one corresponding to the original name of the merged layer.

### Parameters

**Main Layer** [vector: any] Initial layer.

**Additional Layers** [multipleinput: any vectors] Optional.

Layer(s) to merge with.

### Outputs

**Merged Layer** [vector] The resulting layer.

### Console usage

```
processing.runalg('saga:mergeshapelayers', main, layers, out)
```

### See also

### Polar to cartesian coordinates

#### Description

<put algorithm description here>

#### Parameters

**Polar Coordinates** [vector: any] <put parameter description here>

**Exaggeration** [tablefield: any] <put parameter description here>

**Exaggeration Factor** [number] <put parameter description here>

Default: *1*

**Radius** [number] <put parameter description here>

Default: *6371000.0*

**Degree** [boolean] <put parameter description here>

Default: *True*

#### Outputs

**Cartesian Coordinates** [vector] <put output description here>

#### Console usage

```
processing.runalg('saga:polartocartesiancoordinates', polar, f_exagg, d_exagg, radius, degree, ca
```

**See also**

**Quadtree structure to shapes**

**Description**

<put algorithm description here>

**Parameters**

**Shapes** [**vector: any**] <put parameter description here>

**Attribute** [**tablefield: any**] <put parameter description here>

**Outputs**

**Polygons** [**vector**] <put output description here>

**Lines** [**vector**] <put output description here>

**Duplicated Points** [**vector**] <put output description here>

**Console usage**

```
processing.runalg('saga:quadtreestructuretoshapes', shapes, attribute, polygons, lines, points)
```

**See also**

**Shapes buffer**

**Description**

Creates buffer around features based on fixed distance or distance field.

**Parameters**

**Shapes** [**vector: any**] Input layer.

**Buffer Distance** [**selection**] Buffering method.

Options:

- 0 — [0] fixed value
- 1 — [1] attribute field

Default: 0

**Buffer Distance (Fixed)** [**number**] Buffer distance for “fixed value” method.

Default: 100.0

**Buffer Distance (Attribute)** [**tablefield: any**] Name of the distance field for “attribute field” method.

**Scaling Factor for Attribute Value** [**number**] <put parameter description here>

Default: 1.0

**Number of Buffer Zones [number]** Number of buffer(s) to generate.

Default: *1.0*

**Circle Point Distance [Degree] [number]** Smoothness of the buffer borders: great numbers means rough borders.

Default: *5.0*

**Dissolve Buffers [boolean]** Determines whether to dissolve results or not.

Default: *True*

### Outputs

**Buffer [vector]** The resulting layer.

### Console usage

```
processing.runalg('saga:shapesbuffer', shapes, buf_type, buf_dist, buf_field, buf_scale, buf_zones)
```

### See also

### Split shapes layer randomly

#### Description

Splits the input layer randomly in two parts.

#### Parameters

**Shapes [vector: any]** Layer to split.

**Split ratio (%) [number]** Split ratio between the resulting layers.

Default: *50*

#### Outputs

**Group A [vector]** First resulting layer.

**Group B [vector]** Second resulting layer.

#### Console usage

```
processing.runalg('saga:splitshapeslayerrandomly', shapes, percent, a, b)
```

### See also

### Transform shapes

#### Description

<put algorithm description here>



## Parameters

**Shapes** [**vector: any**] <put parameter description here>

**dX** [**number**] <put parameter description here>

Default: *0.0*

**dY** [**number**] <put parameter description here>

Default: *0.0*

**Angle** [**number**] <put parameter description here>

Default: *0.0*

**Scale Factor X** [**number**] <put parameter description here>

Default: *1.0*

**Scale Factor Y** [**number**] <put parameter description here>

Default: *1.0*

**X** [**number**] <put parameter description here>

Default: *0.0*

**Y** [**number**] <put parameter description here>

Default: *0.0*

## Outputs

**Output** [**vector**] <put output description here>

## Console usage

```
processing.runalg('saga:transformshapes', in, dx, dy, angle, scalex, scaley, anchorx, anchory, out)
```

## See also

.

## 18.7.19 Shapes transect

### Transect through polygon shapefile

#### Description

<put algorithm description here>

#### Parameters

**Line Transect (s)** [**vector: line**] <put parameter description here>

**Theme** [**vector: any**] <put parameter description here>

**Theme Field** [**tablefield: any**] <put parameter description here>

## Outputs

**Result table** [table] <put output description here>

## Console usage

```
processing.runalg('saga:transectthroughpolygonshapefile', transect, theme, theme_field, transect_
```

## See also

.

## 18.7.20 Simulation fire

### Fire risk analysis

#### Description

<put algorithm description here>

#### Parameters

**DEM** [raster] <put parameter description here>

**Fuel Model** [raster] <put parameter description here>

**Wind Speed** [raster] <put parameter description here>

**Wind Direction** [raster] <put parameter description here>

**Dead Fuel Moisture 1H** [raster] <put parameter description here>

**Dead Fuel Moisture 10H** [raster] <put parameter description here>

**Dead Fuel Moisture 100H** [raster] <put parameter description here>

**Herbaceous Fuel Moisture** [raster] <put parameter description here>

**Wood Fuel Moisture** [raster] <put parameter description here>

**Value** [raster] Optional.

<put parameter description here>

**Base Probability** [raster] Optional.

<put parameter description here>

**Number of Events** [number] <put parameter description here>

Default: *1000*

**Fire Length** [number] <put parameter description here>

Default: *100*

## Outputs

**Danger** [raster] <put output description here>

**Compound Probability** [raster] <put output description here>

**Priority Index** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:fireriskanalysis', dem, fuel, windspd, winddir, m1h, m10h, m100h, mherb, m
```

## See also

## Simulation

### Description

<put algorithm description here>

### Parameters

**DEM** [raster] <put parameter description here>

**Fuel Model** [raster] <put parameter description here>

**Wind Speed** [raster] <put parameter description here>

**Wind Direction** [raster] <put parameter description here>

**Dead Fuel Moisture 1H** [raster] <put parameter description here>

**Dead Fuel Moisture 10H** [raster] <put parameter description here>

**Dead Fuel Moisture 100H** [raster] <put parameter description here>

**Herbaceous Fuel Moisture** [raster] <put parameter description here>

**Wood Fuel Moisture** [raster] <put parameter description here>

**Ignition Points** [raster] <put parameter description here>

**Update View** [boolean] <put parameter description here>

Default: *True*

### Outputs

**Time** [raster] <put output description here>

**Flame Length** [raster] <put output description here>

**Intensity** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:simulation', dem, fuel, windspd, winddir, m1h, m10h, m100h, mherb, mwood,
```

See also

.

## 18.7.21 Simulation hydrology

### Overland flow - kinematic wave d8

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

**Gauges** [vector: any] Optional.

<put parameter description here>

**Simulation Time** [h] [number] <put parameter description here>

Default: 24

**Simulation Time Step** [h] [number] <put parameter description here>

Default: 0.1

**Manning's Roughness** [number] <put parameter description here>

Default: 0.03

**Max. Iterations** [number] <put parameter description here>

Default: 100

**Epsilon** [number] <put parameter description here>

Default: 0.0001

**Precipitation** [selection] <put parameter description here>

Options:

- 0 — [0] Homogenous
- 1 — [1] Above Elevation
- 2 — [2] Left Half

Default: 0

**Threshold Elevation** [number] <put parameter description here>

Default: 0.0

#### Outputs

**Runoff** [raster] <put output description here>

**Flow at Gauges** [table] <put output description here>

### Console usage

```
processing.runalg('saga:overlandflowkinematicwaved8', dem, gauges, time_span, time_step, roughnes
```

### See also

### Water retention capacity

#### Description

<put algorithm description here>

#### Parameters

**Plot Holes** [vector: any] <put parameter description here>

**DEM** [raster] <put parameter description here>

#### Outputs

**Final Parameters** [vector] <put output description here>

**Water Retention Capacity** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:waterretentioncapacity', shapes, dem, output, retention)
```

### See also

.

## 18.7.22 Table calculus

### Fill gaps in records

#### Description

<put algorithm description here>

#### Parameters

**Table** [table] <put parameter description here>

**Order** [tablefield: any] <put parameter description here>

**Interpolation** [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbour
- 1 — [1] Linear

- 2 — [2] Spline

Default: 0

### Outputs

**Table without Gaps** [table] <put output description here>

### Console usage

```
processing.runalg('saga:fillgapsinrecords', table, order, method, nogaps)
```

### See also

## Principle components analysis

### Description

<put algorithm description here>

### Parameters

**Table** [table] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] correlation matrix
- 1 — [1] variance-covariance matrix
- 2 — [2] sums-of-squares-and-cross-products matrix

Default: 0

**Number of Components** [number] <put parameter description here>

Default: 3

### Outputs

**Principle Components** [table] <put output description here>

### Console usage

```
processing.runalg('saga:principlecomponentsanalysis', table, method, nfirst, pca)
```

### See also

## Running average

### Description

<put algorithm description here>

## Parameters

**Input** [table] <put parameter description here>

**Attribute** [tablefield: any] <put parameter description here>

**Number of Records** [number] <put parameter description here>

Default: 10

## Outputs

**Output** [table] <put output description here>

## Console usage

```
processing.runalg('saga:runningaverage', input, field, count, output)
```

## See also

.

## 18.7.23 Table tools

### Change date format

#### Description

Converts the date format of the input layer.

#### Parameters

**Table** [table] Input table.

**Date Field** [tablefield: any] Attribute the date.

**Input Format** [selection] Input date format.

Options:

- 0 — [0] dd.mm.yy
- 1 — [1] yy.mm.dd
- 2 — [2] dd:mm:yy
- 3 — [3] yy:mm:dd
- 4 — [4] ddmmyyyy, fix size
- 5 — [5] yyyymmdd, fix size
- 6 — [6] ddmmyy, fix size
- 7 — [7] yymmdd, fix size
- 8 — [8] Julian Day

Default: 0

**Output Format [selection]** Output date format.

Options:

- 0 — [0] dd.mm.yy
- 1 — [1] yy.mm.dd
- 2 — [2] dd:mm:yy
- 3 — [3] yy:mm:dd
- 4 — [4] ddmmyyyy, fix size
- 5 — [5] yyyyymmdd, fix size
- 6 — [6] ddmmyy, fix size
- 7 — [7] yymmdd, fix size
- 8 — [8] Julian Day

Default: 0

## Outputs

**Output [table]** The resulting table.

## Console usage

```
processing.runalg('saga:changedateformat', table, field, fmt_in, fmt_out, output)
```

## See also

## Change time format

## Description

Converts the time format of the input layer.

## Parameters

**Table [table]** Input table.

**Time Field [tablefield: any]** Attribute with time.

**Input Format [selection]** Input time format.

Options:

- 0 — [0] hh.mm.ss
- 1 — [1] hh:mm:ss
- 2 — [2] hhmmss, fix size
- 3 — [3] hours
- 4 — [4] minutes
- 5 — [5] seconds

Default: 0



**Output Format [selection]** Output time format.

Options:

- 0 — [0] hh.mm.ss
- 1 — [1] hh:mm:ss
- 2 — [2] hhmmss, fix size
- 3 — [3] hours
- 4 — [4] minutes
- 5 — [5] seconds

Default: 0

## Outputs

**Output [table]** The resulting table.

## Console usage

```
processing.runalg('saga:changetimeformat', table, field, fmt_in, fmt_out, output)
```

## See also

.

## 18.7.24 Terrain channels

### Channel network and drainage basins

#### Description

<put algorithm description here>

#### Parameters

**Elevation [raster]** <put parameter description here>

**Threshold [number]** <put parameter description here>

Default: 5.0

#### Outputs

**Flow Direction [raster]** <put output description here>

**Flow Connectivity [raster]** <put output description here>

**Strahler Order [raster]** <put output description here>

**Drainage Basins [raster]** <put output description here>

**Channels [vector]** <put output description here>

**Drainage Basins [vector]** <put output description here>

**Junctions** [vector] <put output description here>

### Console usage

```
processing.runalg('saga:channelnetworkanddrainagebasins', dem, threshold, direction, connection, c
```

### See also

### Channel network

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

**Flow Direction** [raster] Optional.

<put parameter description here>

**Initiation Grid** [raster] <put parameter description here>

**Initiation Type** [selection] <put parameter description here>

Options:

- 0 — [0] Less than
- 1 — [1] Equals
- 2 — [2] Greater than

Default: 0

**Initiation Threshold** [number] <put parameter description here>

Default: 0.0

**Divergence** [raster] Optional.

<put parameter description here>

**Tracing: Max. Divergence** [number] <put parameter description here>

Default: 10

**Tracing: Weight** [raster] Optional.

<put parameter description here>

**Min. Segment Length** [number] <put parameter description here>

Default: 10

#### Outputs

**Channel Network** [raster] <put output description here>

**Channel Direction** [raster] <put output description here>

**Channel Network** [vector] <put output description here>

### Console usage

```
processing.runalg('saga:channelnetwork', elevation, sinkroute, init_grid, init_method, init_value)
```

### See also

### Overland flow distance to channel network

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

**Channel Network** [raster] <put parameter description here>

**Flow Algorithm** [selection] <put parameter description here>

Options:

- 0 — [0] D8
- 1 — [1] MFD

Default: 0

#### Outputs

**Overland Flow Distance** [raster] <put output description here>

**Vertical Overland Flow Distance** [raster] <put output description here>

**Horizontal Overland Flow Distance** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:overlandflowdistancetochannelnetwork', elevation, channels, method, distanceto)
```

### See also

### Strahler order

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

## Outputs

**Strahler Order** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:strahlerorder', dem, strahler)
```

## See also

## Vertical distance to channel network

### Description

<put algorithm description here>

### Parameters

**Elevation** [raster] <put parameter description here>

**Channel Network** [raster] <put parameter description here>

**Tension Threshold [Percentage of Cell Size]** [number] <put parameter description here>

Default: *1*

**Keep Base Level below Surface** [boolean] <put parameter description here>

Default: *True*

### Outputs

**Vertical Distance to Channel Network** [raster] <put output description here>

**Channel Network Base Level** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:verticaldistancetochannelnetwork', elevation, channels, threshold, nounde
```

## See also

## Watershed basins

### Description

<put algorithm description here>

### Parameters

**Elevation [raster]** <put parameter description here>

**Channel Network [raster]** <put parameter description here>

**Sink Route [raster]** Optional.

<put parameter description here>

**Min. Size [number]** <put parameter description here>

Default: 0

### Outputs

**Watershed Basins [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:watershedbasins', elevation, channels, sinkroute, minsize, basins)
```

### See also

.

## 18.7.25 Terrain hydrology

### Burn stream network into dem

#### Description

<put algorithm description here>

#### Parameters

**DEM [raster]** <put parameter description here>

**Streams [raster]** <put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] simply decrease cell's value by epsilon
- 1 — [1] lower cell's value to neighbours minimum value minus epsilon

Default: 0

**Epsilon [number]** <put parameter description here>

Default: 1.0

#### Outputs

**Processed DEM [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:burnstreamnetworkintodem', dem, stream, method, epsilon, burn)
```

## See also

## Catchment area (flow tracing)

### Description

<put algorithm description here>

### Parameters

**Elevation [raster]** <put parameter description here>

**Sink Routes [raster]** Optional.

<put parameter description here>

**Weight [raster]** Optional.

<put parameter description here>

**Material [raster]** Optional.

<put parameter description here>

**Target [raster]** Optional.

<put parameter description here>

**Step [number]** <put parameter description here>

Default: *1*

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Rho 8
- 1 — [1] Kinematic Routing Algorithm
- 2 — [2] DEMON

Default: *0*

**DEMON - Min. DQV [number]** <put parameter description here>

Default: *0.0*

**Flow Correction [boolean]** <put parameter description here>

Default: *True*

### Outputs

**Catchment Area [raster]** <put output description here>

**Catchment Height [raster]** <put output description here>

**Catchment Slope [raster]** <put output description here>

**Total accumulated Material [raster]** <put output description here>

**Accumulated Material from \_left\_ side** [raster] <put output description here>

**Accumulated Material from \_right\_ side** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:catchmentareafLOWtracing', elevation, sinkroute, weight, material, target,
```

### See also

## Catchment area (recursive)

### Description

<put algorithm description here>

### Parameters

**Elevation** [raster] <put parameter description here>

**Sink Routes** [raster] Optional.

<put parameter description here>

**Weight** [raster] Optional.

<put parameter description here>

**Material** [raster] Optional.

<put parameter description here>

**Target** [raster] Optional.

<put parameter description here>

**Step** [number] <put parameter description here>

Default: 1

**Target Areas** [raster] Optional.

<put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Rho 8
- 2 — [2] Deterministic Infinity
- 3 — [3] Multiple Flow Direction

Default: 0

**Convergence** [number] <put parameter description here>

Default: 1.1

## Outputs

**Catchment Area** [raster] <put output description here>

**Catchment Height** [raster] <put output description here>

**Catchment Slope** [raster] <put output description here>

**Total accumulated Material** [raster] <put output description here>

**Accumulated Material from \_left\_ side** [raster] <put output description here>

**Accumulated Material from \_right\_ side** [raster] <put output description here>

**Flow Path Length** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:catchmentarearecursive', elevation, sinkroute, weight, material, target, ...)
```

## See also

### Catchment Area

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Rho 8
- 2 — [2] Braunschweiger Reliefmodell
- 3 — [3] Deterministic Infinity
- 4 — [4] Multiple Flow Direction
- 5 — [5] Multiple Triangular Flow Directon

Default: 0

## Outputs

**Catchment Area** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:catchmentarea', elevation, method, carea)
```



**See also**

**Cell balance**

**Description**

<put algorithm description here>

**Parameters**

**Elevation [raster]** <put parameter description here>

**Parameter [raster]** Optional.

<put parameter description here>

**Default Weight [number]** <put parameter description here>

Default: *1.0*

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Multiple Flow Direction

Default: *0*

**Outputs**

**Cell Balance [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:cellbalance', dem, weights, weight, method, balance)
```

**See also**

**Edge contamination**

**Description**

<put algorithm description here>

**Parameters**

**Elevation [raster]** <put parameter description here>

**Outputs**

**Edge Contamination [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:edgecontamination', dem, contamination)
```

### See also

### Fill Sinks

### Description

<put algorithm description here>

### Parameters

**DEM [raster]** <put parameter description here>

**Minimum Slope [Degree] [number]** <put parameter description here>

Default: *0.01*

### Outputs

**Filled DEM [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:fillsinks', dem, minslope, result)
```

### See also

### Fill sinks (wang & liu)

### Description

<put algorithm description here>

### Parameters

**DEM [raster]** <put parameter description here>

**Minimum Slope [Degree] [number]** <put parameter description here>

Default: *0.01*

### Outputs

**Filled DEM [raster]** <put output description here>

**Flow Directions [raster]** <put output description here>

**Watershed Basins [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:fillsinkswangliu', elev, minslope, filled, fdir, wshed)
```

### See also

#### Fill sinks xxi (wang & liu)

#### Description

<put algorithm description here>

#### Parameters

**DEM [raster]** <put parameter description here>

**Minimum Slope [Degree] [number]** <put parameter description here>

Default: *0.01*

#### Outputs

**Filled DEM [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:fillsinksxxlwangliu', elev, minslope, filled)
```

### See also

#### Flat detection

#### Description

<put algorithm description here>

#### Parameters

**DEM [raster]** <put parameter description here>

**Flat Area Values [selection]** <put parameter description here>

Options:

- 0 — [0] elevation
- 1 — [1] enumeration

Default: *0*

### Outputs

**No Flats** [raster] <put output description here>

**Flat Areas** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:flatdetection', dem, flat_output, noflats, flats)
```

### See also

### Flow path length

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

**Seeds** [raster] Optional.

<put parameter description here>

**Seeds Only** [boolean] <put parameter description here>

Default: *True*

**Flow Routing Algorithm** [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8 (D8)
- 1 — [1] Multiple Flow Direction (FD8)

Default: *0*

**Convergence (FD8)** [number] <put parameter description here>

Default: *1.1*

### Outputs

**Flow Path Length** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:flowpathlength', elevation, seed, seeds_only, method, convergence, length)
```

**See also**

**Flow width and specific catchment area**

**Description**

<put algorithm description here>

**Parameters**

**Elevation [raster]** <put parameter description here>

**Total Catchment Area (TCA) [raster]** Optional.

<put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Multiple Flow Direction (Quinn et al. 1991)
- 2 — [2] Aspect

Default: 0

**Outputs**

**Flow Width [raster]** <put output description here>

**Specific Catchment Area (SCA) [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:flowwidthandspecificcatchmentarea', dem, tca, method, width, sca)
```

**See also**

**Lake flood**

**Description**

<put algorithm description here>

**Parameters**

**DEM [raster]** <put parameter description here>

**Seeds [raster]** <put parameter description here>

**Absolute Water Levels [boolean]** <put parameter description here>

Default: *True*

## Outputs

**Lake [raster]** <put output description here>

**Surface [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:lakeflood', elev, seeds, level, outdepth, outlevel)
```

## See also

## Ls factor

### Description

<put algorithm description here>

### Parameters

**Slope [raster]** <put parameter description here>

**Catchment Area [raster]** <put parameter description here>

**Area to Length Conversion [selection]** <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (specific catchment area)
- 2 — [2] square root (catchment length)

Default: 0

**Method (LS) [selection]** <put parameter description here>

Options:

- 0 — [0] Moore et al. 1991
- 1 — [1] Desmet & Govers 1996
- 2 — [2] Bochner & Selige 2006

Default: 0

**Rill/Interrill Erosivity [number]** <put parameter description here>

Default: 0.0

**Stability [selection]** <put parameter description here>

Options:

- 0 — [0] stable
- 1 — [1] instable (thawing)

Default: 0

## Outputs

**LS Factor** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:lsfactor', slope, area, conv, method, erosivity, stability, ls)
```

## See also

## Saga wetness index

### Description

<put algorithm description here>

### Parameters

**Elevation** [raster] <put parameter description here>

**t** [number] <put parameter description here>

Default: *10*

### Outputs

**Catchment area** [raster] <put output description here>

**Catchment slope** [raster] <put output description here>

**Modified catchment area** [raster] <put output description here>

**Wetness index** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:sagawetnessindex', dem, t, c, gn, cs, sb)
```

## See also

## Sink drainage route detection

### Description

<put algorithm description here>

### Parameters

**Elevation** [raster] <put parameter description here>

**Threshold** [boolean] <put parameter description here>

Default: *True*

**Threshold Height [number]** <put parameter description here>

Default: *100.0*

### Outputs

**Sink Route [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:sinkdrainageroutedetection', elevation, threshold, thrsheight, sinkroute)
```

### See also

### Sink removal

#### Description

<put algorithm description here>

#### Parameters

**DEM [raster]** <put parameter description here>

**Sink Route [raster]** Optional.

<put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Deepen Drainage Routes
- 1 — [1] Fill Sinks

Default: *0*

**Threshold [boolean]** <put parameter description here>

Default: *True*

**Threshold Height [number]** <put parameter description here>

Default: *100.0*

### Outputs

**Preprocessed DEM [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:sinkremoval', dem, sinkroute, method, threshold, thrsheight, dem_preproc)
```



**See also**

**Slope length**

**Description**

<put algorithm description here>

**Parameters**

**Elevation [raster]** <put parameter description here>

**Outputs**

**Slope Length [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:slopelength', dem, length)
```

**See also**

**Stream power index**

**Description**

<put algorithm description here>

**Parameters**

**Slope [raster]** <put parameter description here>

**Catchment Area [raster]** <put parameter description here>

**Area Conversion [selection]** <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (pseudo specific catchment area)

Default: 0

**Outputs**

**Stream Power Index [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:streampowerindex', slope, area, conv, spi)
```

## See also

### Topographic wetness index (twi)

#### Description

<put algorithm description here>

#### Parameters

**Slope** [raster] <put parameter description here>

**Catchment Area** [raster] <put parameter description here>

**Transmissivity** [raster] Optional.

<put parameter description here>

**Area Conversion** [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (pseudo specific catchment area)

Default: 0

**Method (TWI)** [selection] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] TOPMODEL

Default: 0

#### Outputs

**Topographic Wetness Index** [raster] <put output description here>

#### Console usage

```
processing.runalg('saga:topographicwetnessindextwi', slope, area, trans, conv, method, twi)
```

## See also

### Upslope Area

#### Description

<put algorithm description here>

## Parameters

**Target Area [raster]** Optional.

<put parameter description here>

**Target X coordinate [number]** <put parameter description here>

Default: *0.0*

**Target Y coordinate [number]** <put parameter description here>

Default: *0.0*

**Elevation [raster]** <put parameter description here>

**Sink Routes [raster]** Optional.

<put parameter description here>

**Method [selection]** <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Deterministic Infinity
- 2 — [2] Multiple Flow Direction

Default: *0*

**Convergence [number]** <put parameter description here>

Default: *1.1*

## Outputs

**Upslope Area [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:upslopearea', target, target_pt_x, target_pt_y, elevation, sinkroute, met)
```

## See also

.

## 18.7.26 Terrain lighting

### Analytical hillshading

#### Description

<put algorithm description here>

## Parameters

**Elevation** [**raster**] <put parameter description here>

**Shading Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] Standard (max. 90Degree)
- 2 — [2] Combined Shading
- 3 — [3] Ray Tracing

Default: 0

**Azimuth** [**Degree**] [**number**] <put parameter description here>

Default: 315.0

**Declination** [**Degree**] [**number**] <put parameter description here>

Default: 45.0

**Exaggeration** [**number**] <put parameter description here>

Default: 4.0

## Outputs

**Analytical Hillshading** [**raster**] <put output description here>

## Console usage

```
processing.runalg('saga:analyticalhillshading', elevation, method, azimuth, declination, exaggeration)
```

## See also

### Sky view factor

#### Description

<put algorithm description here>

## Parameters

**Elevation** [**raster**] <put parameter description here>

**Maximum Search Radius** [**number**] <put parameter description here>

Default: 10000

**Method** [**selection**] <put parameter description here>

Options:

- 0 — [0] multi scale
- 1 — [1] sectors

Default: 0

**Multi Scale Factor** [number] <put parameter description here>

Default: 3

**Number of Sectors** [number] <put parameter description here>

Default: 8

### Outputs

**Visible Sky** [raster] <put output description here>

**Sky View Factor** [raster] <put output description here>

**Sky View Factor (Simplified)** [raster] <put output description here>

**Terrain View Factor** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:skyviewfactor', dem, maxradius, method, level_inc, ndirs, visible, svf, s
```

### See also

## Topographic correction

### Description

<put algorithm description here>

### Parameters

**Elevation** [raster] <put parameter description here>

**Original Image** [raster] <put parameter description here>

**Azimuth** [number] <put parameter description here>

Default: *180.0*

**Height** [number] <put parameter description here>

Default: *45.0*

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Cosine Correction (Teillet et al. 1982)
- 1 — [1] Cosine Correction (Civco 1989)
- 2 — [2] Minnaert Correction
- 3 — [3] Minnaert Correction with Slope (Riano et al. 2003)
- 4 — [4] Minnaert Correction with Slope (Law & Nichol 2004)
- 5 — [5] C Correction
- 6 — [6] Normalization (after Civco, modified by Law & Nichol)

Default: *0*

**Minnaert Correction** [number] <put parameter description here>

Default: *0.5*

**Maximum Cells (C Correction Analysis)** [number] <put parameter description here>

Default: *1000*

**Value Range** [selection] <put parameter description here>

Options:

- 0 — [0] 1 byte (0-255)
- 1 — [1] 2 byte (0-65535)

Default: *0*

### Outputs

**Corrected Image** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:topographiccorrection', dem, original, azi, hgt, method, minnaert, maxcel.
```

### See also

.

## 18.7.27 Terrain morphometry

### Convergence index

#### Description

Calculates an index of convergence/divergence regarding to overland flow. By its meaning it is similar to plan or horizontal curvature, but gives much smoother results. The calculation uses the aspects of surrounding cells, i.e. it looks to which degree surrounding cells point to the center cell. The result is given as percentages, negative values correspond to convergent, positive to divergent flow conditions. Minus 100 would be like a peak of a cone, plus 100 a pit, and 0 an even slope.

#### Parameters

**Elevation** [raster] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Aspect
- 1 — [1] Gradient

Default: *0*

**Gradient Calculation** [selection] <put parameter description here>

Options:

- 0 — [0] 2 x 2

- 1 — [1] 3 x 3

Default: 0

## Outputs

**Convergence Index [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:convergenceindex', elevation, method, neighbours, result)
```

## See also

- Koethe, R. / Lehmeier, F. (1996): 'SARA, System zur Automatischen Relief-Analyse', Benutzerhandbuch, 2. Auflage [Geogr. Inst. Univ. Goettingen, unpublished]

## Convergence index (search radius)

### Description

<put algorithm description here>

### Parameters

**Elevation [raster]** <put parameter description here>

**Radius [Cells] [number]** <put parameter description here>

Default: 10

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: 1

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: True

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: 1

**Gradient [boolean]** <put parameter description here>

Default: True

**Difference [selection]** <put parameter description here>

Options:

- 0 — [0] direction to the center cell
- 1 — [1] center cell's aspect direction

Default: 0

## Outputs

**Convergence Index [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:convergenceindexsearchradius', elevation, radius, distance_weighting_weight)
```

## See also

## Curvature classification

### Description

<put algorithm description here>

### Parameters

**Plan Curvature [raster]** <put parameter description here>

**Profile Curvature [raster]** <put parameter description here>

**Threshold for plane [number]** <put parameter description here>

Default: 0.001

## Outputs

**Curvature Classification [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:curvatureclassification', cplan, cprof, threshold, class)
```

## See also

## Diurnal anisotropic heating

### Description

<put algorithm description here>



### Parameters

**Elevation [raster]** <put parameter description here>

**Alpha Max (Degree) [number]** <put parameter description here>

Default: *202.5*

### Outputs

**Diurnal Anisotropic Heating [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:diurnalanisotropicheating', dem, alpha_max, dah)
```

### See also

### Downslope distance gradient

#### Description

<put algorithm description here>

### Parameters

**Elevation [raster]** <put parameter description here>

**Vertical Distance [number]** <put parameter description here>

Default: *10*

**Output [selection]** <put parameter description here>

Options:

- 0 — [0] distance
- 1 — [1] gradient (tangens)
- 2 — [2] gradient (degree)

Default: *0*

### Outputs

**Gradient [raster]** <put output description here>

**Gradient Difference [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:downslopedistancegradient', dem, distance, output, gradient, difference)
```

See also

## Effective air flow heights

### Description

<put algorithm description here>

### Parameters

**Elevation** [raster] <put parameter description here>

**Wind Direction** [raster] Optional.

<put parameter description here>

**Wind Speed** [raster] Optional.

<put parameter description here>

**Constant Wind Direction** [Degree] [number] <put parameter description here>

Default: *135*

**Old Version** [boolean] <put parameter description here>

Default: *True*

**Search Distance** [km] [number] <put parameter description here>

Default: *300*

**Acceleration** [number] <put parameter description here>

Default: *1.5*

**Use Pyramids with New Version** [boolean] <put parameter description here>

Default: *True*

**Lee Factor** [number] <put parameter description here>

Default: *0.5*

**Luv Factor** [number] <put parameter description here>

Default: *1.0*

**Wind Direction Units** [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: *0*

**Wind Speed Scale Factor** [number] <put parameter description here>

Default: *1.0*

### Outputs

**Effective Air Flow Heights** [raster] <put output description here>

### Console usage

```
processing.runalg('saga:effectiveairflowheights', dem, dir, len, dir_const, oldver, maxdist, acce
```

### See also

### Hypsometry

#### Description

<put algorithm description here>

#### Parameters

**Elevation [raster]** <put parameter description here>

**Number of Classes [number]** <put parameter description here>

Default: *100.0*

**Sort [selection]** <put parameter description here>

Options:

- 0 — [0] up
- 1 — [1] down

Default: *0*

**Classification Constant [selection]** <put parameter description here>

Options:

- 0 — [0] height
- 1 — [1] area

Default: *0*

**Use Z-Range [boolean]** <put parameter description here>

Default: *True*

**Z-Range Min [number]** <put parameter description here>

Default: *0.0*

**Z-Range Max [number]** <put parameter description here>

Default: *1000.0*

#### Outputs

**Hypsometry [table]** <put output description here>

### Console usage

```
processing.runalg('saga:hypsometry', elevation, count, sorting, method, bzrange, zrange_min, zran
```

**See also**

**Land surface temperature**

**Description**

<put algorithm description here>

**Parameters**

**Elevation** [m] [raster] <put parameter description here>

**Short Wave Radiation** [kW/m2] [raster] <put parameter description here>

**Leaf Area Index** [raster] <put parameter description here>

**Elevation at Reference Station** [m] [number] <put parameter description here>

Default: 0.0

**Temperature at Reference Station** [Deg.Celsius] [number] <put parameter description here>

Default: 0.0

**Temperature Gradient** [Deg.Celsius/km] [number] <put parameter description here>

Default: 6.5

**C Factor** [number] <put parameter description here>

Default: 1.0

**Outputs**

**Land Surface Temperature** [Deg.Celsius] [raster] <put output description here>

**Console usage**

```
processing.runalg('saga:landsurfacetemperature', dem, swr, lai, z_reference, t_reference, t_gradient)
```

**See also**

**Mass balance index**

**Description**

<put algorithm description here>

**Parameters**

**Elevation** [raster] <put parameter description here>

**Vertical Distance to Channel Network** [raster] Optional.

<put parameter description here>

**T Slope [number]** <put parameter description here>

Default: *15.0*

**T Curvature [number]** <put parameter description here>

Default: *0.01*

**T Vertical Distance to Channel Network [number]** <put parameter description here>

Default: *15.0*

### Outputs

**Mass Balance Index [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:massbalanceindex', dem, hrel, tslope, tcurve, threl, mbi)
```

### See also

## Morphometric protection index

### Description

<put algorithm description here>

### Parameters

**Elevation [raster]** <put parameter description here>

**Radius [number]** <put parameter description here>

Default: *2000.0*

### Outputs

**Protection Index [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:morphometricprotectionindex', dem, radius, protection)
```

### See also

## Multiresolution index of valley bottom flatness (mrvbf)

### Description

<put algorithm description here>

## Parameters

**Elevation [raster]** <put parameter description here>

**Initial Threshold for Slope [number]** <put parameter description here>

Default: *16*

**Threshold for Elevation Percentile (Lowness) [number]** <put parameter description here>

Default: *0.4*

**Threshold for Elevation Percentile (Upness) [number]** <put parameter description here>

Default: *0.35*

**Shape Parameter for Slope [number]** <put parameter description here>

Default: *4.0*

**Shape Parameter for Elevation Percentile [number]** <put parameter description here>

Default: *3.0*

**Update Views [boolean]** <put parameter description here>

Default: *True*

**Classify [boolean]** <put parameter description here>

Default: *True*

**Maximum Resolution (Percentage) [number]** <put parameter description here>

Default: *100*

## Outputs

**MRVBF [raster]** <put output description here>

**MRRTF [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:multiresolutionindexofvalleybottomflatnessmrvbf', dem, t_slope, t_pctl_v,
```

## See also

### Real area calculation

#### Description

<put algorithm description here>

#### Parameters

**Elevation [raster]** <put parameter description here>

#### Outputs

**Real Area Grid [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:realareacalculation', dem, area)
```

### See also

### Relative heights and slope positions

#### Description

<put algorithm description here>

#### Parameters

**Elevation [raster]** <put parameter description here>

**w [number]** <put parameter description here>

Default: *0.5*

**t [number]** <put parameter description here>

Default: *10.0*

**e [number]** <put parameter description here>

Default: *2.0*

#### Outputs

**Slope Height [raster]** <put output description here>

**Valley Depth [raster]** <put output description here>

**Normalized Height [raster]** <put output description here>

**Standardized Height [raster]** <put output description here>

**Mid-Slope Positon [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:relativeheightsandslopepositions', dem, w, t, e, ho, hu, nh, sh, ms)
```

### See also

### Slope, aspect, curvature

#### Description

<put algorithm description here>

## Parameters

**Elevation** [raster] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Maximum Slope (Travis et al. 1975)
- 1 — [1] Maximum Triangle Slope (Tarboton 1997)
- 2 — [2] Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996)
- 3 — [3] Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985)
- 4 — [4] Fit 2.Degree Polynom (Heerdegen & Beran 1982)
- 5 — [5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
- 6 — [6] Fit 3.Degree Polynom (Haralick 1983)

Default: 5

## Outputs

**Slope** [raster] <put output description here>

**Aspect** [raster] <put output description here>

**Curvature** [raster] <put output description here>

**Plan Curvature** [raster] <put output description here>

**Profile Curvature** [raster] <put output description here>

## Console usage

```
processing.runalg('saga:slopeaspectcurvature', elevation, method, slope, aspect, curv, hcurv, vcurv)
```

## See also

### Surface specific points

#### Description

<put algorithm description here>

#### Parameters

**Elevation** [raster] <put parameter description here>

**Method** [selection] <put parameter description here>

Options:

- 0 — [0] Mark Highest Neighbour
- 1 — [1] Opposite Neighbours
- 2 — [2] Flow Direction
- 3 — [3] Flow Direction (up and down)



- 4 — [4] Peucker & Douglas

Default: 0

**Threshold [number]** <put parameter description here>

Default: 2.0

### Outputs

**Result [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:surfacespecificpoints', elevation, method, threshold, result)
```

### See also

### Terrain ruggedness index (tri)

#### Description

<put algorithm description here>

#### Parameters

**Elevation [raster]** <put parameter description here>

**Radius (Cells) [number]** <put parameter description here>

Default: 1

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: 1

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: 1.0

### Outputs

**Terrain Ruggedness Index (TRI) [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:terrainruggednessindextri', dem, radius, distance_weighting_weighting, di
```

### See also

### Topographic position index (tpi)

#### Description

<put algorithm description here>

#### Parameters

**Elevation [raster]** <put parameter description here>

**Standardize [boolean]** <put parameter description here>

Default: *True*

**Min Radius [number]** <put parameter description here>

Default: *0.0*

**Max Radius [number]** <put parameter description here>

Default: *100.0*

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: *1*

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: *75.0*

#### Outputs

**Topographic Position Index [raster]** <put output description here>

### Console usage

```
processing.runalg('saga:topographicpositionindextpi', dem, standard, radius_min, radius_max, dist
```

**See also**

**Tpi based landform classification**

**Description**

<put algorithm description here>

**Parameters**

**Elevation [raster]** <put parameter description here>

**Min Radius A [number]** <put parameter description here>

Default: *0*

**Max Radius A [number]** <put parameter description here>

Default: *100*

**Min Radius B [number]** <put parameter description here>

Default: *0*

**Max Radius B [number]** <put parameter description here>

Default: *1000*

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: *1*

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: *75.0*

**Outputs**

**Landforms [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:tpibasedlandformclassification', dem, radius_a_min, radius_a_max, radius_b_min, radius_b_max, distance_weighting, inverse_distance_weighting_power, inverse_distance_offset, gaussian_and_exponential_weighting_bandwidth)
```

**See also**

**Vector ruggedness measure (vrm)**

**Description**

<put algorithm description here>

**Parameters**

**Elevation [raster]** <put parameter description here>

**Radius (Cells) [number]** <put parameter description here>

Default: *1*

**Distance Weighting [selection]** <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

**Inverse Distance Weighting Power [number]** <put parameter description here>

Default: *1*

**Inverse Distance Offset [boolean]** <put parameter description here>

Default: *True*

**Gaussian and Exponential Weighting Bandwidth [number]** <put parameter description here>

Default: *1*

**Outputs**

**Vector Terrain Ruggedness (VRM) [raster]** <put output description here>

**Console usage**

```
processing.runalg('saga:vectorruggednessmeasurevrm', dem, radius, distance_weighting_weighting, d
```

**See also**

**Wind effect**

**Description**

<put algorithm description here>

## Parameters

**Elevation [raster]** <put parameter description here>

**Wind Direction [raster]** Optional.

<put parameter description here>

**Wind Speed [raster]** Optional.

<put parameter description here>

**Constant Wind Direction [Degree] [number]** <put parameter description here>

Default: *135*

**Old Version [boolean]** <put parameter description here>

Default: *True*

**Search Distance [km] [number]** <put parameter description here>

Default: *300.0*

**Acceleration [number]** <put parameter description here>

Default: *1.5*

**Use Pyramids [boolean]** <put parameter description here>

Default: *True*

**Wind Direction Units [selection]** <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: *0*

**Wind Speed Scale Factor [number]** <put parameter description here>

Default: *1.0*

## Outputs

**Wind Effect [raster]** <put output description here>

**Windward Effect [raster]** <put output description here>

**Leeward Effect [raster]** <put output description here>

## Console usage

```
processing.runalg('saga:windeffect', dem, dir, len, dir_const, oldver, maxdist, accel, pyramids, o
```

## See also

.

## 18.7.28 Terrain profiles

### Cross profiles

#### Description

<put algorithm description here>

#### Parameters

**DEM [raster]** <put parameter description here>

**Lines [vector: line]** <put parameter description here>

**Profile Distance [number]** <put parameter description here>

Default: *10.0*

**Profile Length [number]** <put parameter description here>

Default: *10.0*

**Profile Samples [number]** <put parameter description here>

Default: *10.0*

#### Outputs

**Cross Profiles [vector]** <put output description here>

#### Console usage

```
processing.runalg('saga:crossprofiles', dem, lines, dist_line, dist_profile, num_profile, profile)
```

#### See also

### Profile from points table

#### Description

<put algorithm description here>

#### Parameters

**Grid [raster]** <put parameter description here>

**Input [table]** <put parameter description here>

**X [tablefield: any]** <put parameter description here>

**Y [tablefield: any]** <put parameter description here>

#### Outputs

**Result [table]** <put output description here>

### Console usage

```
processing.runalg('saga:profilefrompointstable', grid, table, x, y, result)
```

### See also

### Profiles from lines

#### Description

<put algorithm description here>

#### Parameters

**DEM [raster]** <put parameter description here>

**Values [multipleinput: rasters]** Optional.

<put parameter description here>

**Lines [vector: line]** <put parameter description here>

**Name [tablefield: any]** <put parameter description here>

**Each Line as new Profile [boolean]** <put parameter description here>

Default: *True*

#### Outputs

**Profiles [vector]** <put output description here>

**Profiles [vector]** <put output description here>

### Console usage

```
processing.runalg('saga:profilesfromlines', dem, values, lines, name, split, profile, profiles)
```

### See also

.

## 18.8 TauDEM algorithm provider

**TauDEM** (Terrain Analysis Using Digital Elevation Models) is a set of Digital Elevation Model (DEM) tools for the extraction and analysis of hydrologic information from topography as represented by a DEM. This is software developed at Utah State University (USU) for hydrologic digital elevation model analysis and watershed delineation.

TauDEM is distributed as a set of standalone command line executable programs for a Windows and source code for compiling and use on other systems.

---

**Bemerkung:** Please remember that Processing contains only the interface description, so you need to install TauDEM 5.0.6 by yourself and configure Processing properly.

---

Documentation for TauDEM algorithms derived from official [TauDEM documentation](#)

## 18.8.1 Basic Grid Analysis

### D8 Contributing Area

#### Description

Calculates a grid of contributing areas using the single direction D8 flow model. The contribution of each grid cell is taken as one (or when the optional weight grid is used, the value from the weight grid). The contributing area for each grid cell is taken as its own contribution plus the contribution from upslope neighbors that drain in to it according to the D8 flow model.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a contributing area value may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the contributing area. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be turned off in cases where you know this is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

#### Parameters

**D8 Flow Direction Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Outlets Shapefile [vector: point]** Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

**Weight Grid [raster]** Optional.

A grid giving contribution to flow for each cell. These contributions (also sometimes referred to as weights or loadings) are used in the contributing area accumulation. If this input file is not used, the contribution to flow will assumed to be one for each grid cell.

**Check for edge contamination [boolean]** A flag that indicates whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a contributing area value may be underestimated due to the fact that grid cells outside of the domain have not been evaluated. This occurs when drainage is inwards from the boundaries or areas with NODATA values for elevation. The algorithm recognizes this and reports NODATA for the impated cells. It is common to see streaks of NODATA values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of available data. Edge contamination checking may be turned off in cases where you know this is not an issue, or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Default: *True*



## Outputs

**D8 Contributing Area Grid [raster]** A grid of contributing area values calculated as the cells own contribution plus the contribution from upslope neighbors that drain in to it according to the D8 flow model.

## Console usage

```
processing.runalg('taudem:d8contributingarea', -p, -o, -wg, -nc, -ad8)
```

## See also

## D8 Flow Directions

### Description

Creates 2 grids. The first contains the flow direction from each grid cell to one of its adjacent or diagonal neighbors, calculated using the direction of steepest descent. The second contain the slope, as evaluated in the direction of steepest descent, and is reported as drop/distance, i.e. tan of the angle. Flow direction is reported as NODATA for any grid cell adjacent to the edge of the DEM domain, or adjacent to a NODATA value in the DEM. In flat areas, flow directions are assigned away from higher ground and towards lower ground using the method of Garbrecht and Martz (1997). The D8 flow direction algorithm may be applied to a DEM that has not had its pits filled, but it will then result in NODATA values for flow direction and slope at the lowest point of each pit.

D8 Flow Direction Coding:

- 1 — East
- 2 — Northeast
- 3 — North
- 4 — Northwest
- 5 — West
- 6 — Southwest
- 7 — South
- 8 — Southeast

The flow direction routing across flat areas is performed according to the method described by Garbrecht, J. and L. W. Martz, (1997), “The Assignment of Drainage Direction Over Flat Surfaces in Raster Digital Elevation Models”, *Journal of Hydrology*, 193: 204-213.

## Parameters

**Pit Filled Elevation Grid [raster]** A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed. Pits are low elevation areas in digital elevation models (DEMs) that are completely surrounded by higher terrain. They are generally taken to be artifacts of the digitation process that interfere with the processing of flow across DEMs. So they are removed by raising their elevation to the point where they just drain off the domain. This step is not essential if you have reason to believe that the pits in your DEM are real. If a few pits actually exist and so should not be removed, while at the same time others are believed to be artifacts that need to be removed, the actual pits should have NODATA elevation values inserted at their lowest point. NODATA values serve to define edges of the domain in the flow field, and elevations are only raised to where flow is off an edge, so an internal NODATA value will stop a pit from being removed, if necessary.

## Outputs

**D8 Flow Direction Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope.

**D8 Slope Grid [raster]** A grid giving slope in the D8 flow direction. This is measured as drop/distance.

## Console usage

```
processing.runalg('taudem:d8flowdirections', -fel, -p, -sd8)
```

## See also

## D-Infinity Contributing Area

### Description

Calculates a grid of specific catchment area which is the contributing area per unit contour length using the multiple flow direction D-infinity approach. D-infinity flow direction is defined as steepest downward slope on planar triangular facets on a block centered grid. The contribution at each grid cell is taken as the grid cell length (or when the optional weight grid input is used, from the weight grid). The contributing area of each grid cell is then taken as its own contribution plus the contribution from upslope neighbors that have some fraction draining to it according to the D-infinity flow model. The flow from each cell either all drains to one neighbor, if the angle falls along a cardinal ( $0, \pi/2, \pi, 3\pi/2$ ) or ordinal ( $\pi/4, 3\pi/4, 5\pi/4, 7\pi/4$ ) direction, or is on an angle falling between the direct angle to two adjacent neighbors. In the latter case the flow is proportioned between these two neighbor cells according to how close the flow direction angle is to the direct angle to those cells. The contour length used here is the grid cell size. The resulting units of the specific catchment area are length units the same as those of the grid cell size.

When the optional weight grid is not used, the result is reported in terms of specific catchment area, the upslope area per unit contour length, taken here as the number of cells times grid cell length (cell area divided by cell length). This assumes that grid cell length is the effective contour length, in the definition of specific catchment area and does not distinguish any difference in contour length dependent upon the flow direction. When the optional weight grid is used, the result is reported directly as a summation of weights, without any scaling.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D-infinity flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a contributing area value may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the contributing area. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be turned off in cases where you know it is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

## Parameters

**D-Infinity Flow Direction Grid [raster]** A grid of flow directions based on the D-infinity flow method using the steepest slope of a triangular facet. Flow direction is determined as the direction of the steepest downward slope on the 8 triangular facets of a 3x3 block centered grid. Flow direction is encoded as an angle in radians, counter-clockwise from east as a continuous (floating point) quantity between 0 and  $2\pi$ . The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

**Outlets Shapefile** [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

**Weight Grid** [raster] Optional.

A grid giving contribution to flow for each cell. These contributions (also sometimes referred to as weights or loadings) are used in the contributing area accumulation. If this input file is not used, the result is reported in terms of specific catchment area (the upslope area per unit contour length) taken as the number of cells times grid cell length (cell area divided by cell length).

**Check for edge contamination** [boolean] A flag that indicates whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a contributing area value may be underestimated due to the fact that grid cells outside of the domain have not been evaluated. This occurs when drainage is inwards from the boundaries or areas with NODATA values for elevation. The algorithm recognizes this and reports NODATA for the imputed cells. It is common to see streaks of NODATA values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of available data. Edge contamination checking may be turned off in cases where you know this is not an issue, or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Default: *True*

## Outputs

**D-Infinity Specific Catchment Area Grid** [raster] A grid of specific catchment area which is the contributing area per unit contour length using the multiple flow direction D-infinity approach. The contributing area of each grid cell is then taken as its own contribution plus the contribution from upslope neighbors that have some fraction draining to it according to the D-infinity flow model.

## Console usage

```
processing.runalg('taudem:dinfinitecontributingarea', -ang, -o, -wg, -nc, -sca)
```

## See also

## D-Infinity Flow Directions

### Description

Assigns a flow direction based on the D-infinity flow method using the steepest slope of a triangular facet (Tarboton, 1997, "A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models", Water Resources Research, 33(2): 309-319). Flow direction is defined as steepest downward slope on planar triangular facets on a block centered grid. Flow direction is encoded as an angle in radians counter-clockwise from east as a continuous (floating point) quantity between 0 and  $2\pi$ . The flow direction angle is determined as the direction of the steepest downward slope on the eight triangular facets formed in a 3 x 3 grid cell window centered on the grid cell of interest. The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

A block-centered representation is used with each elevation value taken to represent the elevation of the center of the corresponding grid cell. Eight planar triangular facets are formed between each grid cell and its eight neighbors. Each of these has a downslope vector which when drawn outwards from the center may be at an angle that lies within or outside the 45 degree ( $\pi/4$  radian) angle range of the facet at the center point. If the slope vector angle is within the facet angle, it represents the steepest flow direction on that facet. If the slope vector angle is

outside a facet, the steepest flow direction associated with that facet is taken along the steepest edge. The slope and flow direction associated with the grid cell is taken as the magnitude and direction of the steepest downslope vector from all eight facets. Slope is measured as drop/distance, i.e. tan of the slope angle.

In the case where no slope vectors are positive (downslope), the flow direction is set using the method of Garbrecht and Martz (1997) for the determination of flow across flat areas. This makes flat areas drain away from high ground and towards low ground. The flow path grid to enforce drainage along existing streams is an optional input, and if used, takes precedence over elevations for the setting of flow directions.

The D-infinity flow direction algorithm may be applied to a DEM that has not had its pits filled, but it will then result in “no data” values for the D-infinity flow direction and slope associated with the lowest point of the pit.

### Parameters

**Pit Filled Elevation Grid [raster]** A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

### Outputs

**D-Infinity Flow Directions Grid [raster]** A grid of flow directions based on the D-infinity flow method using the steepest slope of a triangular facet. Flow direction is determined as the direction of the steepest downward slope on the 8 triangular facets of a 3x3 block centered grid. Flow direction is encoded as an angle in radians, counter-clockwise from east as a continuous (floating point) quantity between 0 and  $2\pi$ . The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

**D-Infinity Slope Grid [raster]** A grid of slope evaluated using the D-infinity method described in Tarboton, D. G., (1997), “A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models”, *Water Resources Research*, 33(2): 309-319. This is the steepest outwards slope on one of eight triangular facets centered at each grid cell, measured as drop/distance, i.e. tan of the slope angle.

### Console usage

```
processing.runalg('taudem:dinfinityflowdirections', -fel, -ang, -slp)
```

### See also

### Grid Network

#### Description

Creates 3 grids that contain for each grid cell: 1) the longest path, 2) the total path, and 3) the Strahler order number. These values are derived from the network defined by the D8 flow model.

The longest upslope length is the length of the flow path from the furthest cell that drains to each cell. The total upslope path length is the length of the entire grid network upslope of each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Strahler order is defined as follows: A network of flow paths is defined by the D8 Flow Direction grid. Source flow paths have a Strahler order number of one. When two flow paths of different order join the order of the downstream flow path is the order of the highest incoming flow path. When two flow paths of equal order join the downstream flow path order is increased by 1. When more than two flow paths join the downstream flow path order is calculated as the maximum of the highest incoming flow path order or the second highest incoming flow path order + 1. This generalizes the common definition to cases where more than two flow paths join at a point.

Where the optional mask grid and threshold value are input, the function is evaluated only considering grid cells that lie in the domain with mask grid value greater than or equal to the threshold value. Source (first order) grid cells are taken as those that do not have any other grid cells from inside the domain draining in to them, and only when two of these flow paths join is order propagated according to the ordering rules. Lengths are also only evaluated counting paths within the domain greater than or equal to the threshold.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

### Parameters

**D8 Flow Direction Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Outlets Shapefile [vector: point]** Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

**Mask Grid [raster]** Optional.

A grid that is used to determine the domain to be analyzed. If the mask grid value  $\geq$  mask threshold (see below), then the cell will be included in the domain. While this tool does not have an edge contamination flag, if edge contamination analysis is needed, then a mask grid from a function like “**D8 Contributing Area**” that does support edge contamination can be used to achieve the same result.

**Mask Threshold [number]** This input parameter is used in the calculation mask grid value  $\geq$  mask threshold to determine if the grid cell is in the domain to be analyzed.

Default: *100*

### Outputs

**Longest Upslope Length Grid [raster]** A grid that gives the length of the longest upslope D8 flow path terminating at each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

**Total Upslope Length Grid [raster]** The total upslope path length is the length of the entire D8 flow grid network upslope of each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

**Strahler Network Order Grid [raster]** A grid giving the Strahler order number for each cell. A network of flow paths is defined by the D8 Flow Direction grid. Source flow paths have a Strahler order number of one. When two flow paths of different order join the order of the downstream flow path is the order of the highest incoming flow path. When two flow paths of equal order join the downstream flow path order is increased by 1. When more than two flow paths join the downstream flow path order is calculated as the maximum of the highest incoming flow path order or the second highest incoming flow path order + 1. This generalizes the common definition to cases where more than two flow paths join at a point.

### Console usage

```
processing.runalg('taudem:gridnetwork', d8_flow_dir_grid, outlets_shape, mask_grid, threshold, lo
```

See also

## Pit Remove

### Description

Identifies all pits in the DEM and raises their elevation to the level of the lowest pour point around their edge. Pits are low elevation areas in digital elevation models (DEMs) that are completely surrounded by higher terrain. They are generally taken to be artifacts that interfere with the routing of flow across DEMs, so are removed by raising their elevation to the point where they drain off the edge of the domain. The pour point is the lowest point on the boundary of the “watershed” draining to the pit. This step is not essential if you have reason to believe that the pits in your DEM are real. If a few pits actually exist and so should not be removed, while at the same time others are believed to be artifacts that need to be removed, the actual pits should have NODATA elevation values inserted at their lowest point. NODATA values serve to define edges in the domain, and elevations are only raised to where flow is off an edge, so an internal NODATA value will stop a pit from being removed, if necessary.

### Parameters

**Elevation Grid [raster]** A digital elevation model (DEM) grid to serve as the base input for the terrain analysis and stream delineation.

### Outputs

**Pit Removed Elevation Grid [raster]** A grid of elevation values with pits removed so that flow is routed off of the domain.

### Console usage

```
processing.runalg('taudem:pitremove', -z, -fel)
```

See also

.

## 18.8.2 Specialized Grid Analysis

### D8 Distance To Streams

#### Description

Computes the horizontal distance to stream for each grid cell, moving downslope according to the D8 flow model, until a stream grid cell is encountered.

#### Parameters

**D8 Flow Direction Grid [raster]** This input is a grid of flow directions that are encoded using the D8 method where all flow from a cells goes to a single neighboring cell in the direction of steepest descent. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Stream Raster Grid [raster]** A grid indicating streams. Such a grid can be created by several of the tools in the “**Stream Network Analysis**” toolset. However, the tools in the “**Stream Network Analysis**” toolset only create grids with a value of 0 for no stream, or 1 for stream cells. This tool can also accept grids with values greater than 1, which can be used in conjunction with the `Threshold` parameter to determine the location of streams. This allows Contributing Area grids to be used to define streams as well as the normal Stream Raster grids. This grid expects integer (long integer) values and any non-integer values will be truncated to an integer before being evaluated.

**Threshold [number]** This value acts as threshold on the `Stream Raster Grid` to determine the location of streams. Cells with a `Stream Raster Grid` value greater than or equal to the `Threshold` value are interpreted as streams.

Default: 50

## Outputs

**Output Distance to Streams [raster]** A grid giving the horizontal distance along the flow path as defined by the D8 Flow Directions Grid to the streams in the Stream Raster Grid.

## Console usage

```
processing.runalg('taudem:d8distancetostreams', -p, -src, -thresh, -dist)
```

## See also

## D-Infinity Avalanche Runout

### Description

Identifies an avalanche’s affected area and the flow path length to each cell in that affected area. All cells downslope from each source area cell, up to the point where the slope from the source to the affected area is less than a threshold angle called the Alpha Angle can be in the affected area. This tool uses the D-infinity multiple flow direction method for determining flow direction. This will likely cause very small amounts of flow to be dispersed to some downslope cells that might overstate the affected area, so a threshold proportion can be set to avoid this excess dispersion. The flow path length is the distance from the cell in question to the source cell that has the highest angle.

All points downslope from the source area are potentially in the affected area, but not beyond a point where the slope from the source to the affected area is less than a threshold angle called the Alpha Angle.

Slope is to be measured using the straight line distance from source point to evaluation point.

It makes more physical sense to me for the angle to be measured along the flow path. Nevertheless it is equally easy to code straight line angles as angles along the flow path, so an option that allows switching will be provided. The most practical way to evaluate avalanche runout is to keep track of the source point with the greatest angle to each point. Then the recursive upslope flow algebra approach will look at a grid cell and all its upslope neighbors that flow to it. Information from the upslope neighbors will be used to calculate the angle to the grid cell in question and retain it in the runout zone if the angle exceeds the alpha angle. This procedure makes the assumption that the maximum angle at a grid cell will be from the set of cells that have maximum angles to the inflowing neighbors. This will always be true of angle is calculated along a flow path, but I can conceive of cases where flow paths bend back on themselves where this would not be the case for straight line angles.

The D-infinity multiple flow direction field assigns flow from each grid cell to multiple downslope neighbors using proportions ( $P_{ik}$ ) that vary between 0 and 1 and sum to 1 for all flows out of a grid cell. It may be desirable to specify a threshold  $T$  that this proportion has to exceed before a grid cell is counted as flowing to a downslope



grid cell, e.g.  $P_{ik} > T$  ( $=0.2$  say) to avoid dispersion to grid cells that get very little flow.  $T$  will be specified as a user input. If all upslope grid cells are to be used  $T$  may be input as 0.

Avalanche source sites are to be input as a short integer grid (name suffix *\*ass*, e.g. *demass*) comprised of positive values where avalanches may be triggered and 0 values elsewhere.

The following grids are output:

- *rz* — A runout zone indicator with value 0 to indicate that this grid cell is not in the runout zone and value  $> 0$  to indicate that this grid cell is in the runout zone. Since there may be information in the angle to the associated source site, this variable will be assigned the angle to the source site (in degrees)
- *dm* — Along flow distance from the source site that has the highest angle to the point in question

## Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

**Pit Filled Elevation Grid [raster]** This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

**Avalanche Source Site Grid [raster]** This is a grid of source areas for snow avalanches that are commonly identified manually using a mix of experience and visual interpretation of maps. Avalanche source sites are to be input as a short integer grid (name suffix *\*ass*, e.g. *demass*) comprised of positive values where avalanches may be triggered and 0 values elsewhere.

**Proportion Threshold [number]** This value is a threshold proportion that is used to limit the dispersion of flow caused by using the D-infinity multiple flow direction method for determining flow direction. The D-infinity multiple flow direction method often causes very small amounts of flow to be dispersed to some downslope cells that might overstate the affected area, so a threshold proportion can be set to avoid this excess dispersion.

Default: *0.2*

**Alpha Angle Threshold [number]** This value is the threshold angle, called the Alpha Angle, that is used to determine which of the cells downslope from the source cells are in the affected area. Only the cells downslope from each source area cell, up to the point where the slope from the source to the affected area is less than a threshold angle are in the affected area.

Default: *18*

**Measure distance along flow path [boolean]** This option selects the method used to measure the distance used to calculate the slope angle. If option is *True* then measure it along the flow path, where the *False* option causes the slope to be measure along the straight line distance from the source cell to the evaluation cell.

Default: *True*

## Outputs

**Runout Zone Grid [raster]** This grid Identifies the avalanche’s runout zone (affected area) using a runout zone indicator with value 0 to indicate that this grid cell is not in the runout zone and value  $> 0$  to indicate that this grid cell is in the runout zone. Since there may be information in the angle to the associated source site, this variable will be assigned the angle to the source site (in degrees).

**Path Distance Grid [raster]** This is a grid of the flow distance from the source site that has the highest angle to each cell.



## Console usage

```
processing.runalg('taudem:dinfiniteavalancherunout', -ang, -fel, -ass, -thresh, -alpha, -direct,
```

## See also

## D-Infinity Concentration Limited Accumulation

### Description

This function applies to the situation where an unlimited supply of a substance is loaded into flow at a concentration or solubility threshold  $C_{sol}$  over a region indicated by an indicator grid ( $dg$ ). It a grid of the concentration of a substance at each location in the domain, where the supply of substance from a supply area is loaded into the flow at a concentration or solubility threshold. The flow is first calculated as a D-infinity weighted contributing area of an input Effective Runoff Weight Grid (notionally excess precipitation). The concentration of substance over the supply area (indicator grid) is at the concentration threshold. As the substance moves downslope with the D-infinity flow field, it is subject to first order decay in moving from cell to cell as well as dilution due to changes in flow. The decay multiplier grid gives the fractional (first order) reduction in quantity in moving from grid cell  $x$  to the next downslope cell. If the outlets shapefile is used, the tool only evaluates the part of the domain that contributes flow to the locations given by the shapefile. This is useful for a tracking a contaminant or compound from an area with unlimited supply of that compound that is loaded into a flow at a concentration or solubility threshold over a zone and flow from the zone may be subject to decay or attenuation.

The indicator grid ( $dg$ ) is used to delineate the area of the substance supply using the (0, 1) indicator function  $i(x)$ .  $A[]$  denotes the weighted accumulation operator evaluated using the D-Infinity Contributing Area function. The Effective Runoff Weight Grid gives the supply to the flow (e.g. the excess rainfall if this is overland flow) denoted as  $w(x)$ . The specific discharge is then given by:

$$Q(x) = A[w(x)]$$

This weighted accumulation  $Q(x)$  is output as the Overland Flow Specific Discharge Grid. Over the substance supply area concentration is at the threshold (the threshold is a saturation or solubility limit). If  $i(x) = 1$ , then

$$C(x) = C_{sol}, \text{ and } L(x) = C_{sol} Q(x),$$

where  $L(x)$  denotes the load being carried by the flow. At remaining locations, the load is determined by load accumulation and the concentration by dilution:

Here  $d(x) = d(i, j)$  is a decay multiplier giving the fractional (first order) reduction in mass in moving from grid cell  $x$  to the next downslope cell. If travel (or residence) times  $t(x)$  associated with flow between cells are available  $d(x)$  may be evaluated as  $\exp(-k t(x))$  where  $k$  is a first order decay parameter. The Concentration grid output is  $C(x)$ . If the outlets shapefile is used, the tool only evaluates the part of the domain that contributes flow to the locations given by the shapefile.

Useful for a tracking a contaminant released or partitioned to flow at a fixed threshold concentration.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This grid can be created by the function “D-Infinity Flow Directions”.

**Disturbance Indicator Grid [raster]** A grid that indicates the source zone of the area of substance supply and must be 1 inside the zone and 0 or NODATA over the rest of the domain.

**Decay Multiplier Grid [raster]** A grid giving the factor by which flow leaving each grid cell is multiplied before accumulation on downslope grid cells. This may be used to simulate the movement of an attenuating or decaying substance. If travel (or residence) times  $t(x)$  associated with flow between cells are available  $d(x)$  may be evaluated as  $\exp(-k t(x))$  where  $k$  is a first order decay parameter.

**Effective Runoff Weight Grid [raster]** A grid giving the input quantity (notionally effective runoff or excess precipitation) to be used in the D-infinity weighted contributing area evaluation of Overland Flow Specific Discharge.

**Outlets shapefile [vector: point]** Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

**Concentration Threshold [number]** The concentration or solubility threshold. Over the substance supply area, concentration is at this threshold.

Default: *1.0*

**Check for edge contamination [boolean]** This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining contributing area.

Default: *True*

## Outputs

**Concentration Grid [raster]** A grid giving the resulting concentration of the compound of interest in the flow.

## Console usage

```
processing.runalg('taudem:dinfiniteconcentrationlimitedaccumulation', -ang, -dg, -dm, -q, -o, -cs
```

## See also

## D-Infinity Decaying Accumulation

### Description

The D-Infinity Decaying Accumulation tool creates a grid of the accumulated quantity at each location in the domain where the quantity accumulates with the D-infinity flow field, but is subject to first order decay in moving from cell to cell. By default, the quantity contribution of each grid cell is the cell length to give a per unit width accumulation, but can optionally be expressed with a weight grid. The decay multiplier grid gives the fractional (first order) reduction in quantity in accumulating from grid cell  $x$  to the next downslope cell.

A decayed accumulation operator  $DA[.]$  takes as input a mass loading field  $m(x)$  expressed at each grid location as  $m(i, j)$  that is assumed to move with the flow field but is subject to first order decay in moving from cell to cell. The output is the accumulated mass at each location  $DA(x)$ . The accumulation of  $m$  at each grid cell can be numerically evaluated.

Here  $d(x) = d(i, j)$  is a decay multiplier giving the fractional (first order) reduction in mass in moving from grid cell  $x$  to the next downslope cell. If travel (or residence) times  $t(x)$  associated with flow between cells are available  $d(x)$  may be evaluated as  $\exp(-k t(x))$  where  $k$  is a first order decay parameter. The weight grid is used to represent the mass loading  $m(x)$ . If not specified this is taken as 1. If the outlets shapefile is used the function is only evaluated on that part of the domain that contributes flow to the locations given by the shapefile.

Useful for a tracking contaminant or compound subject to decay or attenuation.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This grid can be created by the function “**D-Infinity Flow Directions**”.

**Decay Multiplier Grid [raster]** A grid giving the factor by which flow leaving each grid cell is multiplied before accumulation on downslope grid cells. This may be used to simulate the movement of an attenuating substance.

**Weight Grid [raster]** Optional.

A grid giving weights (loadings) to be used in the accumulation. If this optional grid is not specified, weights are taken as the linear grid cell size to give a per unit width accumulation.

**Outlets Shapefile [vector: point]** Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

**Check for edge contamination [boolean]** This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining contributing area.

Default: *True*

### Outputs

**Decayed Specific Catchment Area Grid [raster]** The D-Infinity Decaying Accumulation tool creates a grid of the accumulated mass at each location in the domain where mass moves with the D-infinity flow field, but is subject to first order decay in moving from cell to cell.

### Console usage

```
processing.runalg('taudem:dinfiniteydecayingaccumulation', -ang, -dm, -wg, -o, -nc, -dsca)
```

### See also

## D-Infinity Distance Down

### Description

Calculates the distance downslope to a stream using the D-infinity flow model. The D-infinity flow model is a multiple flow direction model, because the outflow from each grid cell is proportioned between up to 2 downslope grid cells. As such, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of different cells. The statistical method may be selected as the longest, shortest or weighted average of the flow path distance to the stream. Also one of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path, the vertical component of the straight line path, or the total surface flow path.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

**Pit Filled Elevation Grid [raster]** This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

**Stream Raster Grid [raster]** A grid indicating streams, by using a grid cell value of 1 on streams and 0 off streams. This is usually the output of one of the tools in the “**Stream Network Analysis**” toolset.

**Weight Path Grid [raster]** Optional.

A grid giving weights (loadings) to be used in the distance calculation. This might be used for example where only flow distance through a buffer is to be calculated. The weight is then 1 in the buffer and 0 outside it. Alternatively the weight may reflect some sort of cost function for travel over the surface, perhaps representing travel time or attenuation of a process. If this input file is not used, the loadings will assumed to be one for each grid cell.

**Statistical Method [selection]** Statistical method used to calculate the distance down to the stream. In the D-Infinity flow model, the outflow from each grid cell is proportioned between two downslope grid cells. Therefore, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of cells. The distance to the stream may be defined as the longest (maximum), shortest (minimum) or weighted average of the distance down to the stream.

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Average

Default: 2

**Distance Method [selection]** Distance method used to calculate the distance down to the stream. One of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path (horizontal), the vertical component of the straight line path (vertical), or the total surface flow path (surface).

Options:

- 0 — Pythagoras
- 1 — Horizontal
- 2 — Vertical
- 3 — Surface

Default: 1

**Check for edge contamination [boolean]** A flag that determines whether the tool should check for edge contamination. This is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being counted. In the context of Distance Down this occurs when part of a flow path traced downslope from a grid cell leaves the domain without reaching a stream grid cell. With edge contamination checking selected, the algorithm recognizes this and reports no data for the result. This is the desired effect and indicates that values for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be overridden in cases where you know this is not an issue or want to evaluate the distance using only the fraction of flow paths that terminate at a stream.

Default: *True*

## Outputs

**D-Infinity Drop to Stream Grid [raster]** Grid containing the distance to stream calculated using the D-infinity flow model and the statistical and path methods chosen.

## Console usage

```
processing.runalg('taudem:dinfinitydistancedown', dinf_flow_dir_grid, pit_filled_grid, stream_grid)
```

## See also

## D-Infinity Distance Up

### Description

This tool calculates the distance from each grid cell up to the ridge cells along the reverse D-infinity flow directions. Ridge cells are defined to be grid cells that have no contribution from grid cells further upslope. Given the convergence of multiple flow paths at any grid cell, any given grid cell can have multiple upslope ridge cells. There are three statistical methods that this tool can use: maximum distance, minimum distance and waited flow average over these flow paths. A variant on the above is to consider only grid cells that contribute flow with a proportion greater than a user specified threshold ( $t$ ) to be considered as upslope of any given grid cell. Setting  $t=0.5$  would result in only one flow path from any grid cell and would give the result equivalent to a D8 flow model, rather than D-infinity flow model, where flow is proportioned between two downslope grid cells. Finally there are several different optional paths that can be measured: the total straight line path (Pythagoras), the horizontal component of the straight line path, the vertical component of the straight line path, or the total surface flow path.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

**Pit Filled Elevation Grid [raster]** This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

**Slope Grid [raster]** This input is a grid of slope values. This is measured as drop/distance and it is most often obtained as the output of the “**D-Infinity Flow Directions**” tool.

**Statistical Method [selection]** Statistical method used to calculate the distance down to the stream. In the D-Infinity flow model, the outflow from each grid cell is proportioned between two downslope grid cells. Therefore, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of cells. The distance to the stream may be defined as the longest (maximum), shortest (minimum) or weighted average of the distance down to the stream.

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Average

Default: 2

**Distance Method [selection]** Distance method used to calculate the distance down to the stream. One of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path (horizontal), the vertical component of the straight line path (vertical), or the total surface flow path (surface).

Options:

- 0 — Pythagoras

- 1 — Horizontal
- 2 — Vertical
- 3 — Surface

Default: *1*

**Proportion Threshold [number]** The proportion threshold parameter where only grid cells that contribute flow with a proportion greater than this user specified threshold ( $\tau$ ) is considered to be upslope of any given grid cell. Setting  $\tau=0.5$  would result in only one flow path from any grid cell and would give the result equivalent to a D8 flow model, rather than D-Infinity flow model, where flow is proportioned between two downslope grid cells.

Default: *0.5*

**Check for edge contamination [boolean]** A flag that determines whether the tool should check for edge contamination. This is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being counted.

Default: *True*

### Outputs

**D-Infinity Distance Up [raster]** Grid containing the distances up to the ridge calculated using the D-Infinity flow model and the statistical and path methods chosen.

### Console usage

```
processing.runalg('taudem:dinfinitydistanceup', dinf_flow_dir_grid, pit_filled_grid, slope_grid, ...)
```

### See also

## D-Infinity Reverse Accumulation

### Description

This works in a similar way to evaluation of weighted Contributing area, except that the accumulation is by propagating the weight loadings upslope along the reverse of the flow directions to accumulate the quantity of weight loading downslope from each grid cell. The function also reports the maximum value of the weight loading downslope from each grid cell in the Maximum Downslope grid.

This function is designed to evaluate and map the hazard due to activities that may have an effect downslope. The example is land management activities that increase runoff. Runoff is sometimes a trigger for landslides or debris flows, so the weight grid here could be taken as a terrain stability map. Then the reverse accumulation provides a measure of the amount of unstable terrain downslope from each grid cell, as an indicator of the danger of activities that may increase runoff, even though there may be no potential for any local impact.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

**Weight Grid [raster]** A grid giving weights (loadings) to be used in the accumulation.

## Outputs

**Reverse Accumulation Grid [raster]** The grid giving the result of the “Reverse Accumulation” function. This works in a similar way to evaluation of weighted Contributing area, except that the accumulation is by propagating the weight loadings upslope along the reverse of the flow directions to accumulate the quantity of loading downslope from each grid cell.

**Maximum Downslope Grid [raster]** The grid giving the maximum of the weight loading grid downslope from each grid cell.

## Console usage

```
processing.runalg('taudem:dinfiniteinverseaccumulation', -ang, -wg, -racc, -dmax)
```

## See also

### D-Infinity Transport Limited Accumulation - 2

#### Description

This function is designed to calculate the transport and deposition of a substance (e.g. sediment) that may be limited by both supply and the capacity of the flow field to transport it. This function accumulates substance flux (e.g. sediment transport) subject to the rule that transport out of any grid cell is the minimum between supply and transport capacity,  $T_{cap}$ . The total supply at a grid cell is calculated as the sum of the transport in from upslope grid cells,  $T_{in}$ , plus the local supply contribution,  $E$  (e.g. erosion). This function also outputs deposition,  $D$ , calculated as total supply minus actual transport.

Here  $E$  is the supply.  $T_{out}$  at each grid cell becomes  $T_{in}$  for downslope grid cells and is reported as Transport limited accumulation ( $t_{la}$ ).  $D$  is deposition ( $t_{dep}$ ). The function provides the option to evaluate concentration of a compound (contaminant) adhered to the transported substance. This is evaluated as follows:

Where  $L_{in}$  is the total incoming compound loading and  $C_{in}$  and  $T_{in}$  refer to the Concentration and Transport entering from each upslope grid cell.

If

else

where  $C_s$  is the concentration supplied locally and the difference in the second term on the right represents the additional supply from the local grid cell. Then,

$C_{out}$  at each grid cell comprises is the concentration grid output from this function.

If the outlets shapefile is used the tool only evaluates that part of the domain that contributes flow to the locations given by the shapefile.

Transport limited accumulation is useful for modeling erosion and sediment delivery, including the spatial dependence of sediment delivery ratio and contaminant that adheres to sediment.

## Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

**Supply Grid [raster]** A grid giving the supply (loading) of material to a transport limited accumulation function. In the application to erosion, this grid would give the erosion detachment, or sediment supplied at each grid cell.

**Transport Capacity Grid [raster]** A grid giving the transport capacity at each grid cell for the transport limited accumulation function. In the application to erosion this grid would give the transport capacity of the carrying flow.

**Input Concentration Grid [raster]** A grid giving the concentration of a compound of interest in the supply to the transport limited accumulation function. In the application to erosion, this grid would give the concentration of say phosphorous adhered to the eroded sediment.

**Outlets Shapefile [vector: point]** Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

**Check for edge contamination [boolean]** This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining the result.

Default: *True*

## Outputs

**Transport Limited Accumulation Grid [raster]** This grid is the weighted accumulation of supply accumulated respecting the limitations in transport capacity and reports the transport rate calculated by accumulating the substance flux subject to the rule that the transport out of any grid cell is the minimum of the total supply (local supply plus transport in) to that grid cell and the transport capacity.

**Deposition Grid [raster]** A grid giving the deposition resulting from the transport limited accumulation. This is the residual from the transport in to each grid cell minus the transport capacity out of the grid cell. The deposition grid is calculated as the transport in + the local supply - the transport out.

**Output Concentration Grid [raster]** If an input concentration in supply grid is given, then this grid is also output and gives the concentration of a compound (contaminant) adhered or bound to the transported substance (e.g. sediment) is calculated.

## Console usage

```
processing.runalg('taudem:dinfiniteytransportlimitedaccumulation2', dinf_flow_dir_grid, supply_grid)
```



See also

## D-Infinity Transport Limited Accumulation

### Description

This function is designed to calculate the transport and deposition of a substance (e.g. sediment) that may be limited by both supply and the capacity of the flow field to transport it. This function accumulates substance flux (e.g. sediment transport) subject to the rule that transport out of any grid cell is the minimum between supply and transport capacity,  $T_{cap}$ . The total supply at a grid cell is calculated as the sum of the transport in from upslope grid cells,  $T_{in}$ , plus the local supply contribution,  $E$  (e.g. erosion). This function also outputs deposition,  $D$ , calculated as total supply minus actual transport.

Here  $E$  is the supply.  $T_{out}$  at each grid cell becomes  $T_{in}$  for downslope grid cells and is reported as Transport limited accumulation ( $t_{la}$ ).  $D$  is deposition ( $t_{dep}$ ). The function provides the option to evaluate concentration of a compound (contaminant) adhered to the transported substance. This is evaluated as follows:

Where  $L_{in}$  is the total incoming compound loading and  $C_{in}$  and  $T_{in}$  refer to the Concentration and Transport entering from each upslope grid cell.

If

else

where  $C_s$  is the concentration supplied locally and the difference in the second term on the right represents the additional supply from the local grid cell. Then,

$C_{out}$  at each grid cell comprises is the concentration grid output from this function.

If the outlets shapefile is used the tool only evaluates that part of the domain that contributes flow to the locations given by the shapefile.

Transport limited accumulation is useful for modeling erosion and sediment delivery, including the spatial dependence of sediment delivery ratio and contaminant that adheres to sediment.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

**Supply Grid [raster]** A grid giving the supply (loading) of material to a transport limited accumulation function. In the application to erosion, this grid would give the erosion detachment, or sediment supplied at each grid cell.

**Transport Capacity Grid [raster]** A grid giving the transport capacity at each grid cell for the transport limited accumulation function. In the application to erosion this grid would give the transport capacity of the carrying flow.

**Outlets Shapefile [vector: point]** Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

**Check for edge contamination [boolean]** This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining the result.

Default: *True*

## Outputs

**Transport Limited Accumulation Grid [raster]** This grid is the weighted accumulation of supply accumulated respecting the limitations in transport capacity and reports the transport rate calculated by accumulating the substance flux subject to the rule that the transport out of any grid cell is the minimum of the total supply (local supply plus transport in) to that grid cell and the transport capacity.

**Deposition Grid [raster]** A grid giving the deposition resulting from the transport limited accumulation. This is the residual from the transport in to each grid cell minus the transport capacity out of the grid cell. The deposition grid is calculated as the transport in + the local supply - the transport out.

## Console usage

```
processing.runalg('taudem:dinfiniteyntransportlimitedaccumulation', dinf_flow_dir_grid, supply_grid,
```

## See also

## D-Infinity Upslope Dependence

### Description

The D-Infinity Upslope Dependence tool quantifies the amount each grid cell in the domain contributes to a destination set of grid cells. D-Infinity flow directions proportion flow from each grid cell between multiple downslope grid cells. Following this flow field downslope the amount of flow originating at each grid cell that reaches the destination zone is defined. Upslope influence is evaluated using a downslope recursion, examining grid cells downslope from each grid cell, so that the map produced identifies the area upslope where flow through the destination zone originates, or the area it depends on, for its flow.

The figures below illustrate the amount each source point in the domain  $x$  (blue) contributes to the destination point or zone  $y$  (red). If the indicator weighted contributing area function is denoted  $I(y; x)$  giving the weighted contribution using a unit value (1) from specific grid cells  $y$  to grid cells  $x$ , then the upslope dependence is:  $D(x; y) = I(y; x)$ .

This is useful for example to track where flow or a flow related substance or contaminant that enters a destination area may come from.

### Parameters

**D-Infinity Flow Direction Grid [raster]** A grid giving flow direction by the D-Infinity method where the flow direction angle is determined as the direction of the steepest downward slope on the eight triangular facets formed in a 3x3 grid cell window centered on the grid cell of interest. This grid can be produced using the “**D-Infinity Flow Direction**” tool.

**Destination Grid [raster]** A grid that encodes the destination zone that may receive flow from upslope. This grid must be 1 inside the zone  $y$  and 0 over the rest of the domain.

## Outputs

**Output Upslope Dependence Grid [raster]** A grid quantifying the amount each source point in the domain contributes to the zone defined by the destination grid.

## Console usage

```
processing.runalg('taudem:dinfinityupslopedependence', -ang, -dg, -dep)
```

## See also

## Slope Average Down

### Description

This tool computes slope in a D8 downslope direction averaged over a user selected distance. Distance should be specified in horizontal map units.

### Parameters

**D8 Flow Direction Grid [raster]** This input is a grid of flow directions that are encoded using the D8 method where all flow from a cells goes to a single neighboring cell in the direction of steepest descent. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Pit Filled Elevation Grid [raster]** This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

**Downslope Distance [number]** Input parameter of downslope distance over which to calculate the slope (in horizontal map units).

Default: 50

## Outputs

**Slope Average Down Grid [raster]** This output is a grid of slopes calculated in the D8 downslope direction, averaged over the selected distance.

## Console usage

```
processing.runalg('taudem:slopeaveragedown', -p, -fel, -dn, -slpd)
```

See also

## Slope Over Area Ratio

### Description

Calculates the ratio of the slope to the specific catchment area (contributing area). This is algebraically related to the more common  $\ln(a/\tan \beta)$  wetness index, but contributing area is in the denominator to avoid divide by 0 errors when slope is 0.

### Parameters

**Slope Grid [raster]** A grid of slope. This grid can be generated using either the “**D8 Flow Directions**” tool or the “**D-Infinity Flow Directions**” tool.

**Specific Catchment Area Grid [raster]** A grid giving the contributing area value for each cell taken as its own contribution plus the contribution from upslope neighbors that drain in to it. Contributing area is counted in terms of the number of grid cells (or summation of weights). This grid can be generated using either the “**D8 Contributing Area**” tool or the “**D-Infinity Contributing Area**” tool.

### Outputs

**Slope Divided By Area Ratio Grid [raster]** A grid of the ratio of slope to specific catchment area (contributing area). This is algebraically related to the more common  $\ln(a/\tan \beta)$  wetness index, but contributing area is in the denominator to avoid divide by 0 errors when slope is 0.

### Console usage

```
processing.runalg('taudem:slopeoverarearatio', -slp, -sca, -sar)
```

See also

## 18.8.3 Stream Network Analysis

### D8 Extreme Upslope Value

#### Description

Evaluates the extreme (either maximum or minimum) upslope value from an input grid based on the D8 flow model. This is intended initially for use in stream raster generation to identify a threshold of the slope times area product that results in an optimum (according to drop analysis) stream network.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a result may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the result for these grid cells. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that the result for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge

contamination checking may be turned off in cases where you know this is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

### Parameters

**D8 Flow Directions Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Upslope Values Grid [raster]** This is the grid of values of which the maximum or minimum upslope value is selected. The values most commonly used are the slope times area product needed when generating stream rasters according to drop analysis.

**Outlets Shapefile [vector: point]** Optional.

A point shape file defining outlets of interest. If this input file is used, only the area upslope of these outlets will be evaluated by the tool.

**Check for edge contamination [boolean]** A flag that indicates whether the tool should check for edge contamination.

Default: *True*

**Use max upslope value [boolean]** A flag to indicate whether the maximum or minimum upslope value is to be calculated.

Default: *True*

### Outputs

**Extreme Upslope Values Grid [raster]** A grid of the maximum/minimum upslope values.

### Console usage

```
processing.runalg('taudem:d8extremeupslopevalue', -p, -sa, -o, -nc, -min, -ssa)
```

### See also

### Length Area Stream Source

#### Description

Creates an indicator grid (1, 0) that evaluates  $A \geq (M) (Ly)$  based on upslope path length, D8 contributing area grid inputs, and parameters  $M$  and  $y$ . This grid indicates likely stream source grid cells. This is an experimental method with theoretical basis in Hack’s law which states that for streams  $L \sim A^{0.6}$ . However for hillslopes with parallel flow  $L \sim A$ . So a transition from hillslopes to streams may be represented by  $L \sim A^{0.8}$  suggesting identifying grid cells as stream cells if  $A > M (L^{1/0.8})$ .

### Parameters

**Length Grid [raster]** A grid of the maximum upslope length for each cell. This is calculated as the length of the flow path from the furthest cell that drains to each cell. Length is measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal. It is this length ( $L$ ) that is used in the formula,  $A > (M) (Ly)$ , to determine which cells are considered stream cells. This grid can be obtained as an output from the “**Grid Network**” tool.

**Contributing Area Grid [raster]** A grid of contributing area values for each cell that were calculated using the D8 algorithm. The contributing area for a cell is the sum of its own contribution plus the contribution from all upslope neighbors that drain to it, measured as a number of cells. This grid is typically obtained as the output of the “**D8 Contributing Area**” tool. In this tool, it is the contributing area ( $A$ ) that is compared in the formula  $A > (M) (L^y)$  to determine the transition to a stream.

**Threshold [number]** The multiplier threshold ( $M$ ) parameter which is used in the formula:  $A > (M) (L^y)$ , to identify the beginning of streams.

Default: *0.03*

**Exponent [number]** The exponent ( $y$ ) parameter which is used in the formula:  $A > (M) (L^y)$ , to identify the beginning of streams. In branching systems, Hack’s law suggests that  $L = 1/M A^{(1/y)}$  with  $1/y = 0.6$  (or 0.56) ( $y$  about 1.7). In parallel flow systems  $L$  is proportional to  $A$  ( $y$  about 1). This method tries to identify the transition between these two paradigms by using an exponent  $y$  somewhere in between ( $y$  about 1.3).

Default: *1.3*

### Outputs

**Stream Source Grid [raster]** An indicator grid (1,0) that evaluates  $A \geq (M)(L^y)$ , based on the maximum upslope path length, the D8 contributing area grid inputs, and parameters  $M$  and  $y$ . This grid indicates likely stream source grid cells.

### Console usage

```
processing.runalg('taudem:lengthareastreamsource', length_grid, contrib_area_grid, threshold, exp
```

### See also

## Move Outlets To Streams

### Description

Moves outlet points that are not aligned with a stream cell from a stream raster grid, downslope along the D8 flow direction until a stream raster cell is encountered, the “max\_dist” number of grid cells are examined, or the flow path exits the domain (i.e. a “no data” value is encountered for the D8 flow direction). The output file is a new outlets shapefile where each point has been moved to coincide with the stream raster grid, if possible. A field “dist\_moved” is added to the new outlets shapefile to indicate the changes made to each point. Points that are already on a stream cell are not moved and their “dist\_moved” field is assigned a value 0. Points that are initially not on a stream cell are moved by sliding them downslope along the D8 flow direction until one of the following occurs: a) A stream raster grid cell is encountered before traversing the “max\_dist” number of grid cells. In which case, the point is moved and the “dist\_moved” field is assigned a value indicating how many grid cells the point was moved. b) More than the “max\_number” of grid cells are traversed, or c) the traversal ends up going out of the domain (i.e., a “no data” D8 flow direction value is encountered). In which case, the point is not moved and the “dist\_moved” field is assigned a value of -1.

### Parameters

**D8 Flow Direction Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Stream Raster Grid [raster]** This output is an indicator grid (1, 0) that indicates the location of streams, with a value of 1 for each of the stream cells and 0 for the remainder of the cells. This file is produced by several different tools in the “**Stream Network Analysis**” toolset.

**Outlets Shapefile [vector: point]** A point shape file defining points of interest or outlets that should ideally be located on a stream, but may not be exactly on the stream due to the fact that the shapefile point locations may not have been accurately registered with respect to the stream raster grid.

**Maximum Number of Grid Cells to traverse [number]** This input parameter is the maximum number of grid cells that the points in the input outlet shapefile will be moved before they are saved to the output outlet shapefile.

Default: 50

## Outputs

**Output Outlet Shapefile [vector]** A point shape file defining points of interest or outlets. This file has one point in it for each point in the input outlet shapefile. If the original point was located on a stream, then the point was not moved. If the original point was not on a stream, the point was moved downslope according to the D8 flow direction until it reached a stream or the maximum distance had been reached. This file has an additional field “dist\_moved” added to it which is the number of cells that the point was moved. This field is 0 if the cell was originally on a stream, -1 if it was not moved because there was not a stream within the maximum distance, or some positive value if it was moved.

## Console usage

```
processing.runalg('taudem:moveoutletstostreams', -p, -src, -o, -md, -om)
```

## See also

## Peuker Douglas

### Description

Creates an indicator grid (1, 0) of upward curved grid cells according to the Peuker and Douglas algorithm.

With this tool, the DEM is first smoothed by a kernel with weights at the center, sides, and diagonals. The Peuker and Douglas (1975) method (also explained in Band, 1986), is then used to identify upwardly curving grid cells. This technique flags the entire grid, then examines in a single pass each quadrant of 4 grid cells, and unflags the highest. The remaining flagged cells are deemed “upwardly curved”, and when viewed, resemble a channel network. This proto-channel network generally lacks connectivity and requires thinning, issues that were discussed in detail by Band (1986).

### Parameters

**Elevation Grid [raster]** A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

**Center Smoothing Weight [number]** The center weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: 0.4

**Side Smoothing Weight [number]** The side weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: 0.1

**Diagonal Smoothing Weight [number]** The diagonal weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: *0.05*

### Outputs

**Stream Source Grid [raster]** An indicator grid (1, 0) of upward curved grid cells according to the Peuker and Douglas algorithm, and if viewed, resembles a channel network. This proto-channel network generally lacks connectivity and requires thinning, issues that were discussed in detail by Band (1986).

### Console usage

```
processing.runalg('taudem:peukerdouglas', elevation_grid, center_weight, side_weight, diagonal_we
```

### See also

- Band, L. E., (1986), “Topographic partition of watersheds with digital elevation models”, *Water Resources Research*, 22(1): 15-24.
- Peuker, T. K. and D. H. Douglas, (1975), “Detection of surface-specific points by local parallel processing of discrete terrain elevation data”, *Comput. Graphics Image Process.*, 4: 375-387.

## Slope Area Combination

### Description

Creates a grid of slope-area values =  $(S_m)^m (A_n)^n$  based on slope and specific catchment area grid inputs, and parameters  $m$  and  $n$ . This tool is intended for use as part of the slope-area stream raster delineation method.

### Parameters

**Slope Grid [raster]** This input is a grid of slope values. This grid can be obtained from the “**D-Infinity Flow Directions**” tool.

**Contributing Area Grid [raster]** A grid giving the specific catchment area for each cell taken as its own contribution (grid cell length or summation of weights) plus the proportional contribution from upslope neighbors that drain in to it. This grid is typically obtained from the “**D-Infinity Contributing Area**” tool.

**Slope Exponent [number]** The slope exponent ( $m$ ) parameter which will be used in the formula:  $(S_m)^m (A_n)^n$ , that is used to create the slope-area grid.

Default: *2*

**Area Exponent [number]** The area exponent ( $n$ ) parameter which will be used in the formula:  $(S_m)^m (A_n)^n$ , that is used to create the slope-area grid.

Default: *1*

### Outputs

**Slope Area Grid [raster]** A grid of slope-area values =  $(S_m)^m (A_n)^n$  calculated from the slope grid, specific catchment area grid,  $m$  slope exponent parameter, and  $n$  area exponent parameter.



## Console usage

```
processing.runalg('taudem:slopeareacombination', slope_grid, area_grid, slope_exponent, area_exponent)
```

## See also

## Stream Definition By Threshold

### Description

Operates on any grid and outputs an indicator (1, 0) grid identifying cells with input values  $\geq$  the threshold value. The standard use is to use an accumulated source area grid to as the input grid to generate a stream raster grid as the output. If you use the optional input mask grid, it limits the domain being evaluated to cells with mask values  $\geq 0$ . When you use a D-infinity contributing area grid (*\*sca*) as the mask grid, it functions as an edge contamination mask. The threshold logic is:

```
src = ((ssa >= thresh) & (mask >= s0)) ? 1:0
```

### Parameters

**Accumulated Stream Source Grid [raster]** This grid nominally accumulates some characteristic or combination of characteristics of the watershed. The exact characteristic(s) varies depending on the stream network raster algorithm being used. This grid needs to have the property that grid cell values are monotonically increasing downslope along D8 flow directions, so that the resulting stream network is continuous. While this grid is often from an accumulation, other sources such as a maximum upslope function will also produce a suitable grid.

**Threshold [number]** This parameter is compared to the value in the Accumulated Stream Source grid (*\*ssa*) to determine if the cell should be considered a stream cell. Streams are identified as grid cells for which ssa value is  $\geq$  this threshold.

Default: *100*

**Mask Grid [raster]** Optional.

This optional input is a grid that is used to mask the domain of interest and output is only provided where this grid is  $\geq 0$ . A common use of this input is to use a D-Infinity contributing area grid as the mask so that the delineated stream network is constrained to areas where D-infinity contributing area is available, replicating the functionality of an edge contamination mask.

### Outputs

**Stream Raster Grid [raster]** This is an indicator grid (1, 0) that indicates the location of streams, with a value of 1 for each of the stream cells and 0 for the remainder of the cells.

## Console usage

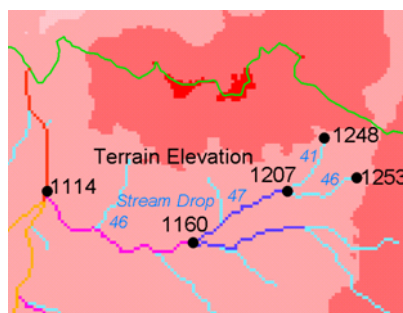
```
processing.runalg('taudem:streamdefinitionbythreshold', -ssa, -thresh, -mask, -src)
```

See also

## Stream Drop Analysis

### Description

Applies a series of thresholds (determined from the input parameters) to the input accumulated stream source grid (\*ssa) grid and outputs the results in the \*drp.txt file the stream drop statistics table. This function is designed to aid in the determination of a geomorphologically objective threshold to be used to delineate streams. Drop Analysis attempts to select the right threshold automatically by evaluating a stream network for a range of thresholds and examining the constant drop property of the resulting Strahler streams. Basically it asks the question: Is the mean stream drop for first order streams statistically different from the mean stream drop for higher order streams, using a T-test. Stream drop is the difference in elevation from the beginning to the end of a stream defined as the sequence of links of the same stream order. If the T-test shows a significant difference then the stream network does not obey this “law” so a larger threshold needs to be chosen. The smallest threshold for which the T-test does not show a significant difference gives the highest resolution stream network that obeys the constant stream drop “law” from geomorphology, and is the threshold chosen for the “objective” or automatic mapping of streams from the DEM. This function can be used in the development of stream network rasters, where the exact watershed characteristic(s) that were accumulated in the accumulated stream source grid vary based on the method being used to determine the stream network raster.



The constant stream drop “law” was identified by Broscocoe (1959). For the science behind using this to determine a stream delineation threshold, see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

### Parameters

**D8 Contributing Area Grid [raster]** A grid of contributing area values for each cell that were calculated using the D8 algorithm. The contributing area for a cell is the sum of its own contribution plus the contribution from all upslope neighbors that drain to it, measured as a number of cells or the sum of weight loadings. This grid can be obtained as the output of the “**D8 Contributing Area**” tool. This grid is used in the evaluation of drainage density reported in the stream drop table.

**D8 Flow Direction Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

**Pit Filled Elevation Grid [raster]** A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

**Accumulated Stream Source Grid [raster]** This grid must be monotonically increasing along the downslope D8 flow directions. It is compared to a series of thresholds to determine the beginning of the streams. It is often generated by accumulating some characteristic or combination of characteristics of the watershed with the “**D8 Contributing Area**” tool, or using the maximum option of the “**D8 Flow Path Extreme**” tool. The exact method varies depending on the algorithm being used.

**Outlets Shapefile [vector: point]** A point shapefile defining the outlets upstream of which drop analysis is performed.

**Minimum Threshold [number]** This parameter is the lowest end of the range searched for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 5

**Maximum Threshold [number]** This parameter is the highest end of the range searched for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 500

**Number of Threshold Values [number]** The parameter is the number of steps to divide the search range into when looking for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 10

**Spacing for Threshold Values [selection]** This parameter indicates whether logarithmic or linear spacing should be used when looking for possible threshold values using drop analysis.

Options:

- 0 — Logarithmic
- 1 — Linear

Default: 0

## Outputs

**D-Infinity Drop to Stream Grid [file]** This is a comma delimited text file with the following header line:

```
:: Threshold,DrainDen,NoFirstOrd,NoHighOrd,MeanDFirstOrd,MeanDHighOrd,StdDevFirstOrd,StdDevHighOrd,T
```

The file then contains one line of data for each threshold value examined, and then a summary line that indicates the optimum threshold value. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis, see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

## Console usage

```
processing.runalg('taudem:streamdropanalysis', d8_contrib_area_grid, d8_flow_dir_grid, pit_filled,
```

## See also

- Broscoc, A. J., (1959), "Quantitative analysis of longitudinal stream profiles of small watersheds", Office of Naval Research, Project NR 389-042, Technical Report No. 18, Department of Geology, Columbia University, New York.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1991), "On the Extraction of Channel Networks from Digital Elevation Data", Hydrologic Processes, 5(1): 81-100.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1992), "A Physical Basis for Drainage Density", Geomorphology, 5(1/2): 59-76.
- Tarboton, D. G. and D. P. Ames, (2001), "Advances in the mapping of flow networks from digital elevation data", World Water and Environmental Resources Congress, Orlando, Florida, May 20-24, ASCE, <http://www.engineering.usu.edu/dtarb/asce2001.pdf>.

## Stream Reach and Watershed

### Description

This tool produces a vector network and shapefile from the stream raster grid. The flow direction grid is used to connect flow paths along the stream raster. The Strahler order of each stream segment is computed. The sub-watershed draining to each stream segment (reach) is also delineated and labeled with the value identifier that corresponds to the WSNO (watershed number) attribute in the Stream Reach Shapefile.

This tool orders the stream network according to the Strahler ordering system. Streams that don't have any other streams draining in to them are order 1. When two stream reaches of different order join the order of the downstream reach is the order of the highest incoming reach. When two reaches of equal order join the downstream reach order is increased by 1. When more than two reaches join the downstream reach order is calculated as the maximum of the highest incoming reach order or the second highest incoming reach order + 1. This generalizes the common definition to cases where more than two reaches join at a point. The network topological connectivity is stored in the Stream Network Tree file, and coordinates and attributes from each grid cell along the network are stored in the Network Coordinates file.

The stream raster grid is used as the source for the stream network, and the flow direction grid is used to trace connections within the stream network. Elevations and contributing area are used to determine the elevation and contributing area attributes in the network coordinate file. Points in the outlets shapefile are used to logically split stream reaches to facilitate representing watersheds upstream and downstream of monitoring points. The program uses the attribute field "id" in the outlets shapefile as identifiers in the Network Tree file. This tool then translates the text file vector network representation in the Network Tree and Coordinates files into a shapefile. Further attributes are also evaluated. The program has an option to delineate a single watershed by representing the entire area draining to the Stream Network as a single value in the output watershed grid.

### Parameters

**Pit Filled Elevation Grid [raster]** A grid of elevation values. This is usually the output of the "**Pit Remove**" tool, in which case it is elevations with pits removed.

**D8 Flow Direction Grid [raster]** A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the "**D8 Flow Directions**" tool.

**D8 Drainage Area [raster]** A grid giving the contributing area value in terms of the number of grid cells (or the summation of weights) for each cell taken as its own contribution plus the contribution from upslope neighbors that drain in to it using the D8 algorithm. This is usually the output of the "**D8 Contributing Area**" tool and is used to determine the contributing area attribute in the Network Coordinate file.

**Stream Raster Grid [raster]** An indicator grid indicating streams, by using a grid cell value of 1 on streams and 0 off streams. Several of the "**Stream Network Analysis**" tools produce this type of grid. The Stream Raster Grid is used as the source for the stream network.

**Outlets Shapefile as Network Nodes [vector: point]** Optional.

A point shape file defining points of interest. If this file is used, the tool will only delineate the stream network upstream of these outlets. Additionally, points in the Outlets Shapefile are used to logically split stream reaches to facilitate representing watersheds upstream and downstream of monitoring points. This tool **REQUIRES THAT THERE BE** an integer attribute field "id" in the Outlets Shapefile, because the "id" values are used as identifiers in the Network Tree file.

**Delineate Single Watershed [boolean]** This option causes the tool to delineate a single watershed by representing the entire area draining to the Stream Network as a single value in the output watershed grid. Otherwise a separate watershed is delineated for each stream reach. Default is *False* (separate watershed).

Default: *False*

## Outputs

**Stream Order Grid [raster]** The Stream Order Grid has cells values of streams ordered according to the Strahler order system. The Strahler ordering system defines order 1 streams as stream reaches that don't have any other reaches draining in to them. When two stream reaches of different order join the order of the downstream reach is the order of the highest incoming reach. When two reaches of equal order join the downstream reach order is increased by 1. When more than two reaches join the downstream reach order is calculated as the maximum of the highest incoming reach order or the second highest incoming reach order + 1. This generalizes the common definition to cases where more than two flow paths reaches join at a point.

**Watershed Grid [raster]** This output grid identified each reach watershed with a unique ID number, or in the case where the delineate single watershed option was checked, the entire area draining to the stream network is identified with a single ID.

**Stream Reach Shapefile [vector]** This output is a polyline shapefile giving the links in a stream network. The columns in the attribute table are:

- LINKNO — Link Number. A unique number associated with each link (segment of channel between junctions). This is arbitrary and will vary depending on number of processes used
- DSLINKNO — Link Number of the downstream link. -1 indicates that this does not exist
- USLINKNO1 — Link Number of first upstream link. (-1 indicates no link upstream, i.e. for a source link)
- USLINKNO2 — Link Number of second upstream link. (-1 indicates no second link upstream, i.e. for a source link or an internal monitoring point where the reach is logically split but the network does not bifurcate)
- DSNODEID — Node identifier for node at downstream end of stream reach. This identifier corresponds to the "id" attribute from the Outlets shapefile used to designate nodes
- Order — Strahler Stream Order
- Length — Length of the link. The units are the horizontal map units of the underlying DEM grid
- Magnitude — Shreve Magnitude of the link. This is the total number of sources upstream
- DS\_Cont\_Ar — Drainage area at the downstream end of the link. Generally this is one grid cell upstream of the downstream end because the drainage area at the downstream end grid cell includes the area of the stream being joined
- Drop — Drop in elevation from the start to the end of the link
- Slope — Average slope of the link (computed as drop/length)
- Straight\_L — Straight line distance from the start to the end of the link
- US\_Cont\_Ar — Drainage area at the upstream end of the link
- WSNO — Watershed number. Cross reference to the `*w.shp` and `*w` grid files giving the identification number of the watershed draining directly to the link
- DOUT\_END — Distance to the eventual outlet (i.e. the most downstream point in the stream network) from the downstream end of the link
- DOUT\_START — Distance to the eventual outlet from the upstream end of the link
- DOUT\_MID — Distance to the eventual outlet from the midpoint of the link

**Network Connectivity Tree [file]** This output is a text file that details the network topological connectivity is stored in the Stream Network Tree file. Columns are as follows:

- Link Number (Arbitrary — will vary depending on number of processes used)
- Start Point Number in Network coordinates (`*coord.dat`) file (Indexed from 0)
- End Point Number in Network coordinates (`*coord.dat`) file (Indexed from 0)

- Next (Downstream) Link Number. Points to Link Number. -1 indicates no links downstream, i.e. a terminal link
- First Previous (Upstream) Link Number. Points to Link Number. -1 indicates no upstream links
- Second Previous (Upstream) Link Numbers. Points to Link Number. -1 indicates no upstream links. Where only one previous link is -1, it indicates an internal monitoring point where the reach is logically split, but the network does not bifurcate
- Strahler Order of Link
- Monitoring point identifier at downstream end of link. -1 indicates downstream end is not a monitoring point
- Network magnitude of the link, calculated as the number of upstream sources (following Shreve)

**Network Coordinates [file]** This output is a text file that contains the coordinates and attributes of points along the stream network. Columns are as follows:

- X coordinate
- Y Coordinate
- Distance along channels to the downstream end of a terminal link
- Elevation
- Contributing area

### Console usage

```
processing.runalg('taudem:streamreachandwatershed', -fel, -p, -ad8, -src, -o, -sw, -ord, -w, -net,
```

### See also

.

---

## Druckzusammenstellung

---

Mit der Druckzusammenstellung können Sie ansprechende Karten und Atlanten erstellen die als PDF-Datei, als Bild oder als SVG-Datei gedruckt oder gespeichert werden können. Dies ist eine leistungsstarke Methode geografische Informationen, die von QGIS erstellt werden in Berichten zu teilen oder zu veröffentlichen.

Die Druckzusammenstellung stellt immer mehr Layout- und Druckfunktionen bereit. Sie macht es möglich Elemente wie die QGIS Kartenanzeige, Beschriftungen, Legenden, Maßstäbe, Grundformen, Pfeile, Attributtabelle und HTML-Rahmen hinzuzufügen. Sie können bei jedem Element die Größe verändern, es gruppieren, ausrichten und positionieren als auch die Eigenschaften anpassen um Ihr Layout zu erstellen. Das Layout kann in Bildformate wie PostScript, PDF oder SVG (der Export in SVG funktioniert nicht mit einigen aktuellen Qt4 Versionen, Sie sollten dies individuell aus Ihrem System ausprobieren) gedruckt oder exportiert werden. Sie können das Layout als Vorlage speichern und es wieder in anderen Sitzungen laden. Schließlich kann das Erstellen mehrerer Karten basierend auf einer Vorlage mit der Atlaserzeugung durchgeführt werden. Sie finden eine Liste von Tools in [table\\_composer\\_1](#):






















































Icon	Zweck	Icon	Zweck
	Projekt speichern		Neue Zusammenstellung
	Zusammenstellungsduplizierung		Druckzusammenstellung verwalten
	Aus Vorlage laden		Als Vorlage speichern
	Drucken oder Exportieren als PostScript		Speichern als Rasterbild
	Speichern als SVG		Als PDF exportieren
	Letzte Änderung zurücknehmen		Letzte Änderung wiederherstellen
	Volle Ausdehnung		Zoom auf 100%
	Hineinzoomen		Hinauszoomen
	Aktualisiere Ansicht		Auf bestimmte Region zoomen
	Druckzusammenstellung verschieben		Den Elementinhalt verschieben
	Eintrag wählen/verschieben		Bild hinzufügen
	Neue Karte aus QGIS Kartenansicht hinzufügen		Neue Legende hinzufügen
	Neue Beschriftung hinzufügen		Einfaches Objekt hinzufügen
	Einen Maßstab zu Druckzusammenstellung hinzufügen		Fügt Attributtabelle hinzu
	Pfeil hinzufügen		Gruppe auflösen
	Einen HTML-Rahmen hinzufügen		Alle Elemente entsichern
	Elemente gruppieren		Versenken
	Gewählte Element fixieren		In den Hintergrund schicken
	Hervorholen		Linksbündig
	In den Vordergrund holen		Zentrieren
	Linksbündig		Oben bündig
	Zentrieren		Atlasvorschau
	Oben bündig		Vorheriges Objekt
	Atlasvorschau		Letztes Objekt
	Vorheriges Objekt		Atlas drucken
	Letztes Objekt		Atlas-Einstellungen
	Atlas als Bilder exportieren		

Tabelle Composer 1: Funktionen der Druckzusammenstellung


Alle Werkzeuge der Druckzusammenstellung stehen in Menüs und als Icons in der Werkzeugleiste zur Verfügung. Die Werkzeugleiste kann aus und angeschaltet werden, indem Sie rechte Maustaste in der Werkzeugleiste verwenden.

## 19.1 Erste Schritte

### 19.1.1 Ein neues Template öffnen

Bevor Sie mit der Druckzusammenstellung anfangen zu arbeiten müssen Sie einige Raster- und Vektorlayer in das QGIS Kartenfenster laden und Ihre Eigenschaften Ihren Wünschen anpassen. Wenn alles nach Ihren Wünschen



dargestellt und symbolisiert ist klicken Sie das  *Neue Druckzusammenstellung* Icon in der Werkzeugleiste oder wählen Sie *Projekt* → *Neue Druckzusammenstellung*. Sie werden aufgefordert einen Titel für die neue Zusammenstellung auszuwählen.

### 19.1.2 Überblick über die Druckzusammenstellung

Das Öffnen der Druckzusammenstellung bietet Ihnen eine leere Seite, die die Blattoberfläche beim Benutzen der Druckoption darstellt. Zunächst finden Sie Knöpfe auf der linken Seite der Seite um Druckzusammenstellungselemente hinzuzufügen; die aktuelle QGIS Kartenansicht, Beschriftungen, Bilder, Legenden, Maßstäbe, einfache Formen, Pfeile, Attributtabelle und HTML-Rahmen. In dieser Werkzeugleiste finden Sie auch Werkzeugleistenknöpfe zum Navigieren, Zoomen in einen Bereich und zum Verschieben der Ansicht in der Zusammenstellung und Werkzeugleistenknöpfe um ein Druckzusammenstellungselement auszuwählen und den Inhalt des Kartenelements zu verschieben.

[Figure\\_composer\\_overview](#) zeigt die anfängliche Ansicht der Druckzusammenstellung bevor Elemente hinzugefügt sind.

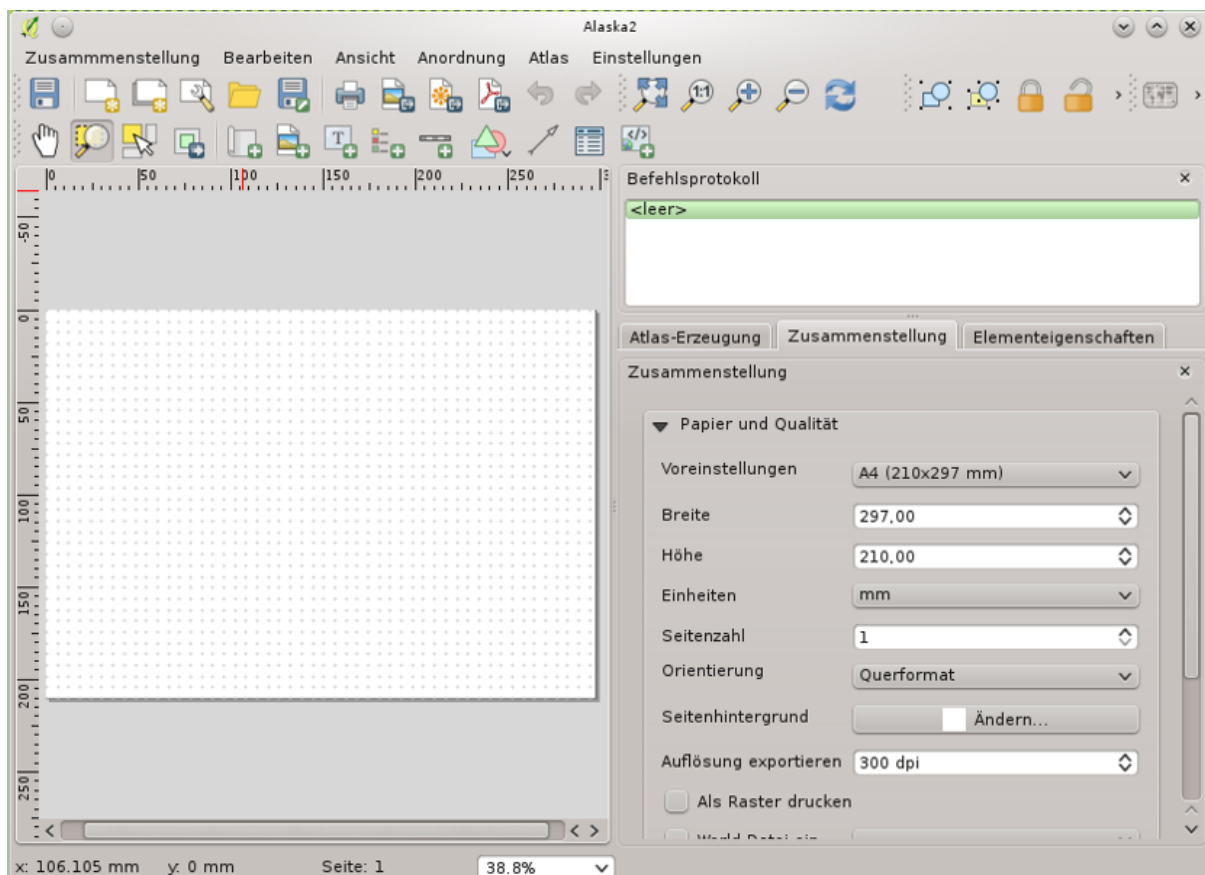





Abbildung 19.1: Druckzusammenstellung 

Rechts von der Seite finden Sie zwei Bedienfelder. Das obere Bedienfeld enthält die Reiter *Elemente* und *Befehlsprotokoll* und das untere Bedienfeld enthält die Reiter *Zusammenstellung*, *Elementeigenschaften* und *Atlas-Erzeugung*.

- Der *Elementeigenschaften* Reiter stellt eine Liste aller Kartenzusammenstellungselemente die der Seite hinzugefügt wurden.
- Der *Befehlsprotokoll* Reiter stellt einen Verlauf aller Änderungen, die am Druckzusammenstellungslayout durchgeführt wurden dar. Mit einem Mausklick ist es möglich Layoutschritte vorwärts oder rückwärts zu einem bestimmten Status rückgängig zu machen oder wiederherzustellen.








- Der Reiter *Zusammenstellung* ermöglicht Ihnen die Papiergröße, Orientierung, den Seitenhintergrund, Anzahl der Seiten und die Druckqualität der Ausgabedatei in dpi einzustellen. Darüberhinaus können Sie auch das  *Als Raster drucken* Kontrollkästchen aktivieren. Das heisst, dass alle Elemente vor dem Drucken oder Speichern als PostScript oder PDF gerastert werden. In diesem Reiter können Sie auch benutzerdefinierte Einstellungen für Gitter und Ausrichtungen vornehmen.
- The *Item Properties* tab displays the properties for the selected item. Click the  *Select/Move item* icon to select an item (e.g., legend, scale bar or label) on the canvas. Then click the *Item Properties* tab and customize the settings for the selected item.
- Der *Atlas-Erzeugung* Reiter ermöglicht es Ihnen die Erstellung eines Atlas für die aktuelle Zusammenstellung zu bewirken und bietet Zugang zu seinen Parametern.
- Schließlich können Sie Ihre Druckzusammenstellung mit dem  *Projekt speichern* Knopf speichern.

Im unteren Teil des Druckzusammenstellungsfensters können Sie eine Statusleiste mit der Mausposition, aktuellen Seitenanzahl und einer Kombobox zum Einstellen der Zoomstufe finden.

Sie können der Zusammenstellung mehrere Element hinzufügen. Es ist auch möglich, dass die Druckzusammenstellungsansicht mehr als eine Karte, Legende oder Maßstab enthält, dies auf einer oder mehreren Seiten. Jedes Element hat seine eigenen Eigenschaften und, im Fall der Karte, seine eigenen Ausmaße. Wenn Sie ein Element aus der Zusammenstellungsansicht entfernen wollen können Sie das mit *Entf* oder *Rücktaste* tun.

## In der Druckzusammenstellung navigieren




Um im Layout navigieren zu können bietet die Druckzusammenstellung einige grundlegende Tools:



-  Hineinzoomen
-  Hinauszoomen
-  Volle Ausdehnung
-  Zoom auf 100%
-  Aktualisiere Ansicht
-  Druckzusammenstellung verschieben
-  Zoom (zu einer bestimmten Region der Zusammenstellung zoomen)

Sie können den Zoomlevel auch verändern indem Sie das Mauseisen oder die Kombobox in der Statusleiste benutzen. Wenn Sie in den Verschieben Modus beim Arbeiten in der Zusammenstellungsansicht wechseln müssen können sie die *Leertaste* oder das Mauseisen gedrückt halten. Mit *Strg+Leertaste* können Sie zeitweise zum Zoommodus wechseln und mit *Strg+Umschalt+Leertaste* zum Hinauszoomen Modus wechseln.

### 19.1.3 Beispielsitzung

Um zu zeigen wie man eine Karte erstellt folgen Sie den nächsten Anweisungen.

1. Wählen Sie den  *Neue Karte hinzufügen* Werkzeugleistenknopf aus und zeichnen Sie ein Rechteck auf die Seite indem Sie die linke Maustaste gedrückt halten. Innerhalb des gezeichneten Rechteckes erscheint die QGIS Kartenansicht auf der Seite.
2. Wählen Sie den  *Neuen Maßstab hinzufügen* Werkzeugleistenknopf und platzieren Sie das Kartenelement mit der linken Maustaste auf der Druckzusammenstellungsseite. Es wird ein Maßstab der Seite hinzugefügt.
3. Wählen Sie den  *Neue Legende hinzufügen* Werkzeugleistenknopf und zeichnen Sie ein Rechteck auf die Seite indem sie die linke Maustaste gedrückt halten. Innerhalb des Rechtecks wird die Legende gezeichnet.

4. Wählen Sie das  Den Elementinhalt verschieben Icon um die Karte auszuwählen und verschieben Sie es ein bisschen.
5. Während das Kartenelement noch ausgewählt ist können Sie auch die Größe des Kartenelements verändern. Klicken Sie während Sie die linke Maustaste gedrückt halten in eine weißes kleines Rechteck in einer der Ecken des Kartenelements und verschieben Sie es an einen neuen Ort um seine Größe zu verändern.
6. Klicken Sie den *Elementeigenschaften* Reiter im linken unteren Bedienfeld und suchen Sie die Einstellung für die Orientierung. Ändern Sie den Wert der Einstellung *Kartenorientierung* auf '15.00°'. Sie sollten sehen wie die Orientierung des Kartenelements sich ändert.
7. Schließlich können Sie Ihre Druckzusammenstellung mit dem  Projekt speichern Knopf speichern.


### 19.1.4 Druckzusammenstellungsoptionen

Unter *Einstellungen* → *Zusammenstellungseinstellungen* ... können Sie einige Optionen, die als voreingestellt während Ihrer Arbeit benutzt werden, einstellen.

- Mit *Zusammenstellungsvoreinstellungen* können Sie die Standardschriftart, die verwendet werden soll, benutzen.
- Mit *Gitterdarstellung* können Sie den Gitterstil und seine Farbe festlegen.
- *Gittervoreinstellungen* definiert Zwischenräume, Gitterversatz und Fangtoleranz des Gitters. Es gibt drei Gitterstile: **Punkte**, **Ausgefüllt** und **Kreuze**.
- *Führungsvorgaben* definiert die Fangtoleranz für die Führungslinien.

### 19.1.5 Reiter Zusammenstellung - Allgemeines Zusammenstellung Setup

Im Reiter *Zusammenstellung* können Sie die globalen Einstellungen Ihrer Zusammenstellung definieren.

- Sie können eine der *Voreinstellungen* für Ihre Seite auswählen oder Ihre benutzerdefinierte *Breite* und *Höhe* eingeben.
- Die Zusammenstellung kann nun in mehrere Seiten aufgeteilt werden. Beispielsweise kann die erste Seite eine Kartenansicht zeigen und eine zweite Seite kann die zum Layer gehörende Attributtabelle zeigen während die dritte einen HTML-Rahmen, der zu der Webseite Ihrer Organisation zeigt, zeigt. Stellen Sie die *Seitenzahl* auf den gewünschten Wert ein. Sie können die Seiten *Orientierung* und Ihre exportierte Auflösung wählen. Wenn es aktiviert ist mit  *Als Raster drucken* gemeint, dass alle Elemente vor dem Drucken oder Speichern als PostScript oder PDF gerastert werden.
- Mit *Gitter* können Sie Gittereinstellungen wie *Abstand*, *Gitterversatz* und *Toleranz* an Ihre Anforderungen anpassen.
- Unter *Auf Ausrichtungen einrasten* können Sie die *Toleranz*, welches der maximale Abstand, unter dem ein Element auf die Ausrichtungen snappt, ist, ändern.

Das Snappen auf das Gitter und/oder auf die Ausrichtungen kann im *Ansicht* Menü aktiviert werden. In diesem Menü können Sie auch das Gitter und die Führungen verstecken oder anzeigen.

### 19.1.6 Gemeinsame Eigenschaften von Zusammenstellungselementen

Zusammenstellungselemente besitzen eine Reihe von allgemeinen Eigenschaften die Sie am unteren Ende des *Elementeigenschaften* Reiters finden: Position und Größe, Rahmen, Hintergrund, Elementkennung und Darstellung (siehe [figure\\_composer\\_common\\_1](#)).

- Der *Position und Größe* Dialog erlaubt es Ihnen die Größe und Position des Rahmens, der das Element beinhaltet, zu definieren. Sie können auch auswählen welcher Referenzpunkt an den vorher definierten **X** und **Y** Koordinaten gesetzt wird.

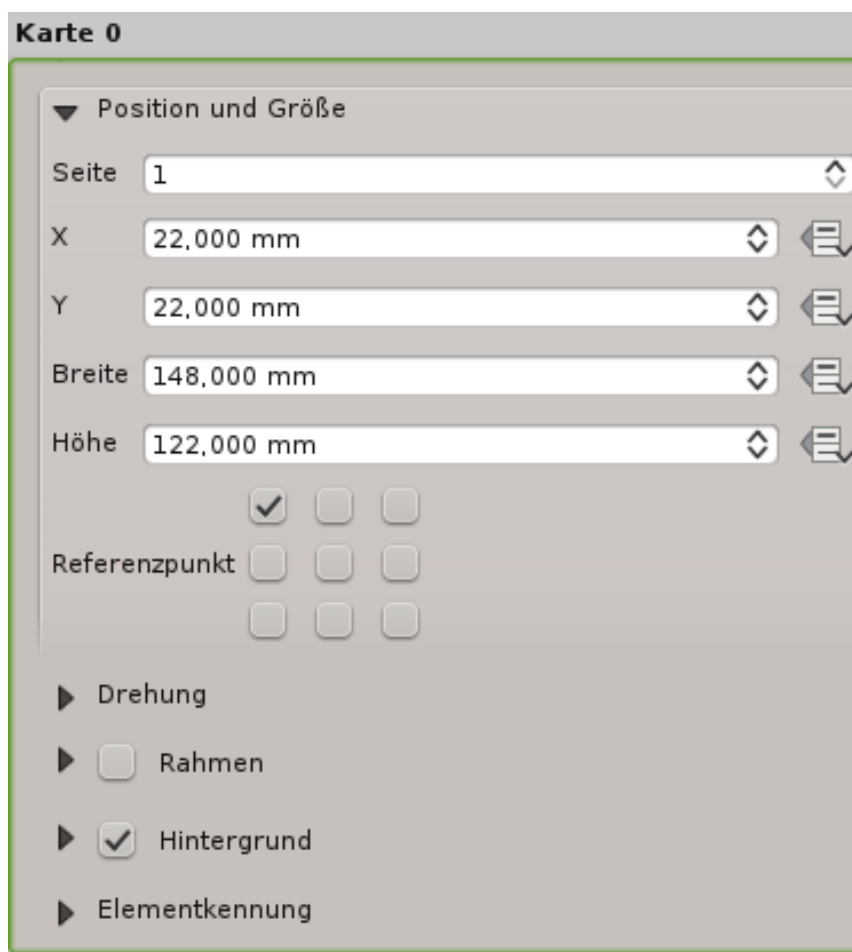




Abbildung 19.2: Allgemeiner Elementeigenschaften Dialog 

- Die *Drehung* stellt die Drehung des Elements (in Grad) ein.
- Der  *Rahmen* zeigt oder versteckt den Rahmen um die Beschriftung. Klicken Sie auf die [**Farbe**] und [**Dicke**] Knöpfe um die Eigenschaften anzupassen.
- Der  *Hintergrund* aktiviert oder deaktiviert eine Hintergrundfarbe. Klicken Sie auf den [**Farbe...**] Knopf um einen Dialog aufzurufen wo Sie eine Farbe auswählen oder aus einer Voreinstellung auswählen können. Die Transparenz kann ebenfalls mit dem **Alpha** Feld angepasst werden.
- Verwenden Sie die *Elementkennung* um eine Beziehung zu anderen Elementen der Druckzusammenstellung herzustellen. Dies wird beim QGIS Server und anderen potentiellen Webclients benutzt. Sie können eine ID an ein Element vergeben (z.B. eine Karte und eine Beschriftung) und dann kann der Webclient Daten zum Einstellen einer Eigenschaft (z.B. Beschriftung) für das bestimmte Element senden. Der GetProjectSettings Befehl führt auf welche Elemente und welche IDs in einem Layout zur Verfügung stehen.
- *Darstellung* kann im Optionen Feld ausgewählt werden. Siehe [Rendering\\_Mode](#).

**Bemerkung:**

- Wenn Sie  *Farbauswahldialoge verwenden, die laufend aktualisieren* in den QGIS Allgemein Optionen aktiviert haben, wird der Farbkopf sich aktualisieren sobald Sie eine neue Farbe in den **Farbdialog** Fenstern ausgewählt haben. Wenn nicht dann müssen Sie den **Farbdialog** schließen.
- Das  *Datendefinierte Übersteuerung* Icon neben einem Feld bedeutet dass Sie einem Feld Daten im Kartenelement zuweisen können oder dass Sie Ausdrücke verwenden können. Diese sind besonders hilfreich bei der Atlaserstellung (siehe [atlas\\_data\\_defined\\_overrides](#)).

## 19.2 Darstellung

QGIS ermöglicht jetzt eine erweiterte Darstellung für Zusammenstellungselement wie Vektor- und Rasterlayer.

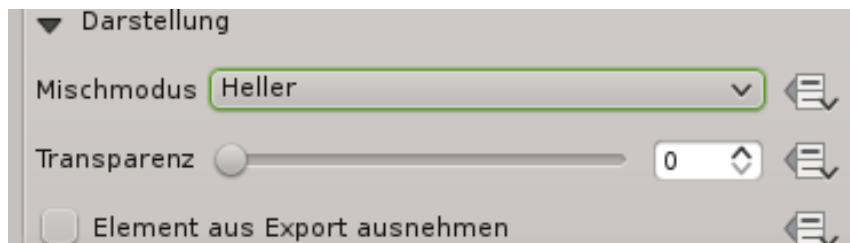

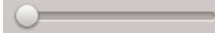



Abbildung 19.3: Darstellung 

- *Transparenz* : Sie können das unten liegende Element in der Zusammenstellung mit diesem Tool sichtbar machen. Verwenden Sie den Schieberegler um die Sichtbarkeit Ihres Elements an Ihre Bedürfnisse anzupassen. Sie können auch eine genaue Definition des Prozentsatzes der Sichtbarkeit im Menü neben dem Schieberegler machen.
- *Element aus Export ausnehmen*: Sie können sich entschließen ein Element in allen Exporten nicht sichtbar zu machen. Nach dem Aktivieren dieses Kontrollkästchens wird das Element nicht in PDF's, Drucken etc. enthalten sein
- *Mischmodus*: Sie können spezielle Darstellungseffekte mit diesen Tools, die Sie vorher vielleicht nur von Grafikprogrammen kannten, erzielen. Die Pixel Ihrer oben liegenden und darunterliegenden Elemente werden durch die unten beschriebenen Einstellungen gemischt.
  - Normal: Dies ist der Standardmischmodus, der den Alphakanal des oben liegenden Pixels mit dem darunter liegenden Pixel vermischt. Die Farben werden nicht vermischt.

- **Heller:** Dies wählt das Maximum jeder Komponente der Vordergrund- und Hintergrundpixel. Seien Sie sich bewusst dass die Ergebnisse zackig und hart aussehen können.
- **Bildschirm:** Helle Pixel der Quelle werden über die des Ziels gezeichnet, wohingegen dunkle Pixel nicht verwendet werden. Dieser Modus ist am nützlichsten für das Mischen der Textur eines Layers mit einem anderen Layer (z.B. kann man eine Schummerung dazu verwenden einen anderen Layer mit einer Textur zu versehen).
- **Abwedeln:** das Abwedeln erhellt und sättigt unten liegende Pixel auf Basis der Helligkeit des oben liegenden Pixels. Demzufolge erhöhen hellere oben liegende Pixel die Sättigung und Helligkeit des unten liegenden Pixels. Dies funktioniert am Besten wenn die oben liegenden Pixel nicht zu hell sind; andernfalls ist der Effekt zu extrem.
- **Addition:** Dieser Mischmodus fügt einfach die Pixelwerte eines Layers denen eines anderen Layers hinzu. Im Falle von Werten größer 1 (im Fall von RGB) wird weiß dargestellt. Dieser Modus ist dafür geeignet Objekte hervorzuheben.
- **Dunkler:** Dies erstellt ein Ergebnispixel das die kleinste Komponente der Vordergrund und Hintergrundpixel erhält. Wie das Aufhellen neigen die Ergebnisse dazu zackig und hart zu sein.
- **Multiplizieren:** Hier werden die Nummern für jedes Pixel des oben liegenden Layers mit den entsprechenden Pixeln des unteren Layers multipliziert. Das Ergebnis sind dunklere Bilder.
- **Einbrennen:** Dunklere Farben im oben liegenden Layer bewirken ein Verdunkeln des unten liegenden Layers. Einbrennen kann dazu benutzt werden um unten liegende Layer zu optimieren und zu colorieren.
- **Überlagern:** Dieser Modus kombiniert die Multiplizieren und Bildschirm Mischmodi. Im Ergebnispixel werden helle Bereiche heller und dunkle Bereiche dunkler.
- **Weiches Licht:** Dieses ist dem Überlagern sehr ähnlich nur dass anstelle Multiplizieren/Bildschirm Einbrennen/Abwedeln verwendet wird. Hier soll das Leuchten eines weichen Lichtes auf ein Bild nachgeahmt werden.
- **Hartes Licht:** Hartes Licht ist dem Überlagern Modus sehr ähnlich. Es soll die Projektion eines sehr intensiven Lichts auf das Bild nachahmen.
- **Unterschied:** Unterschied subtrahiert das oben liegende Pixel von dem unten liegenden Pixel oder andersherum um immer einen positiven Wert zu bekommen. Das Mischen mit Schwarz produziert keinen Unterschied, da die Differenz mit allen Farben Null ist.
- **Abziehen:** Dieser Mischmodus zieht einfach die Pixelwerte eines Layers von dem anderen ab. Im Fall von negativen Werten wird Schwarz dargestellt.


## 19.3 Zusammenstellungselemente



### 19.3.1 Das Kartenelement

Klicken Sie auf den  **Neue Karte hinzufügen** Werkzeugleistenknopf in der Druckzusammenstellung um die QGIS Kartenansicht hinzuzufügen. Ziehen Sie jetzt ein Rechteck mit der linken Maustaste über die Zusammenstellungsansicht um die Karte hinzuzufügen. Um die aktuelle Karte anzuzeigen können Sie zwischen drei verschiedenen Modi im *Elementeigenschaften* Reiter wählen:

- **Rechteck** ist die Standardeinstellung. Es wird eine leeres Kästchen mit der Meldung 'Karte wird hier gedruckt' angezeigt.
- **Cache** zeichnet die Karte in der aktuellen Bildschirmauflösung. Wenn Sie in das Zusammenstellungsfenster hinein- oder hinauszoomen, wird die Karte nicht wieder gerendert aber das Bild wird skaliert.
- **Zeichnen** meint dass wenn Sie in das Zusammenstellungsfenster hineinzoomen oder nicht, die Karte wieder gerendert wird, aber aus Platzgründen nur bis zu einer maximalen Auflösung.

**Cache** ist der voreingestellte Vorschaumodus für neu hinzugefügte Druckzusammenstellungskarten.

Sie können die Größe des Kartenelements durch Klicken auf den  Knopf verändern indem Sie das Element auswählen und eins der blauen Kästchen an der Ecke der Karte verschieben. Mit der ausgewählten Karte können Sie jetzt mehr Eigenschaften im *Elementeigenschaften* Reiter anpassen.

Um Layer innerhalb des Kartenelements zu verschieben wählen sie das Kartenelement aus, klicken Sie das  Icon und verschieben Sie die Layer innerhalb des Karteninhaltsrahmen mit der linken Maustaste. Nachdem Sie den richtigen Platz für das Element gefunden haben, können Sie die Elementposition innerhalb der Druckzusammenstellungsseite sichern. Wählen Sie das Kartenelement und verwenden Sie aus der Werkzeugleiste  oder das *Elemente* Bedienfeld. Einmal ausgewählt können Sie das *Elemente* Bedienfeld verwenden um individuelle Elemente zu entsichern.




### Haupteigenschaften

Der *Haupteigenschaften* Dialog des Karten *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten bereit (siehe *figure\_composer\_map\_1*):



Abbildung 19.4: Elementeigenschaften Reiter 



- Der **Vorschau** Bereich ermöglicht es Ihnen die Vorschau Modi 'Rechteck', 'Cache' und 'Zeichnen' wie oben beschrieben zu definieren. Wenn Sie die Ansicht im QGIS Kartenfenster verändern indem Sie Vektor- oder Rastereigenschaften verändern, können Sie die Druckzusammenstellung aktualisieren, indem Sie das Kartenelement in der Druckzusammenstellung auswählen und den [**Vorschau aktualisieren**] Knopf klicken.
- Das Feld *Maßstab*  stellt einen Maßstab manuell ein.
- Das Feld *Drehung*  ermöglicht es Ihnen die Kartenelementinhalte im Uhrzeigersinn in Grad zu drehen. Beachten Sie, dass ein Koordinatenrahmen nur beim Standardwert 0 hinzugefügt werden kann.
- Mit  *Kartenelemente zeichnen* können Sie Beschriftungen zeigen, die in der Kartenanzeige im QGIS Hauptfenster platziert worden sind.
- Sie können sich dazu entschließen Layer, die in einem Kartenelement gezeigt werden zu sichern. Aktivieren Sie  *Layer des Kartenelements festhalten*. Nachdem dies aktiviert ist wird jeder Layer, der im QGIS Hauptfenster angezeigt oder versteckt wird, nicht erscheinen oder versteckt sein im Kartenelement der Zusammenstellung. Stil und Beschriftung eines gesicherten Layers werden immer noch gemäß dem QGIS Hauptfenster aktualisiert.
- Der  Knopf ermöglicht es Ihnen schnell alle Voreinstellungen, die Sie in QGIS vorbereitet haben, hinzuzufügen. Wenn Sie auf den  Knopf drücken werden Sie die Liste aller Voreinstellungen sehen: wählen Sie einfach die Voreinstellung aus, die sie darstellen wollen. Die Kartenansicht wird automatisch die voreingestellten Layer sichern indem  *Layer des Kartenelements festhalten* aktiviert wird: wenn sie die Voreinstellung abwählen wollen, deaktivieren Sie die  und drücken Sie auf den  Knopf. Siehe *Legende* um herauszufinden wie man Voreinstellungen erstellt.

## Ausdehnung

Der *Ausdehnung* Dialog der Karten Elementeigenschaften stellt die folgenden Funktionalitäten bereit (siehe [figure\\_composer\\_map\\_2](#)):



Abbildung 19.5: Kartenausdehnung Dialog 

- Der **Kartenausdehnung** Bereich ermöglicht es Ihnen die Kartenausdehnung anhand von X und Y min/max Werten anzugeben indem Sie den [**Anzeigegrenzen übernehmen**] Knopf klicken. Dieser Knopf setzt die Kartenausdehnung der Zusammenstellungskarte auf die Ausmaße der aktuellen Kartenansicht in der QGIS Hauptansicht. Der Knopf [**Ausmaße in der Karte anzeigen**] macht genau das Gegenteil, es aktualisiert die Ausmaße der Kartenansicht in der QGIS Anwendung auf die Ausdehnung der Zusammenstellungskarte.



Wenn Sie die Ansicht in der QGIS Kartenanzeige ändern indem Sie Vektorlayer- oder Rasterlayereigenschaften ändern können Sie die Druckzusammenstellungsansicht aktualisieren indem Sie das Kartenelement in der Druckzusammenstellung auswählen und den **[Vorschau aktualisieren]** Knopf im *Elementeigenschaften* Reiter der Karte klicken (siehe [figure\\_composer\\_map\\_1](#)).

## Gitter

Der *Gitter anzeigen* Dialog des Karten *Elementeigenschaften* Reiters bietet die Möglichkeit dem Kartenelement mehrere Gitter hinzuzufügen.

- Mit dem Plus- und Minusknopf könne Sie ein ausgewähltes Gitter hinzufügen oder auswählen.
- Mit den Hoch und Runterknöpfen können Sie ein Gitter aus der Liste verschieben und die Zeichenpriorität festlegen.

Wenn Sie auf das hinzugefügte Gitter doppelklicken können Sie ihm einen anderen Namen geben.

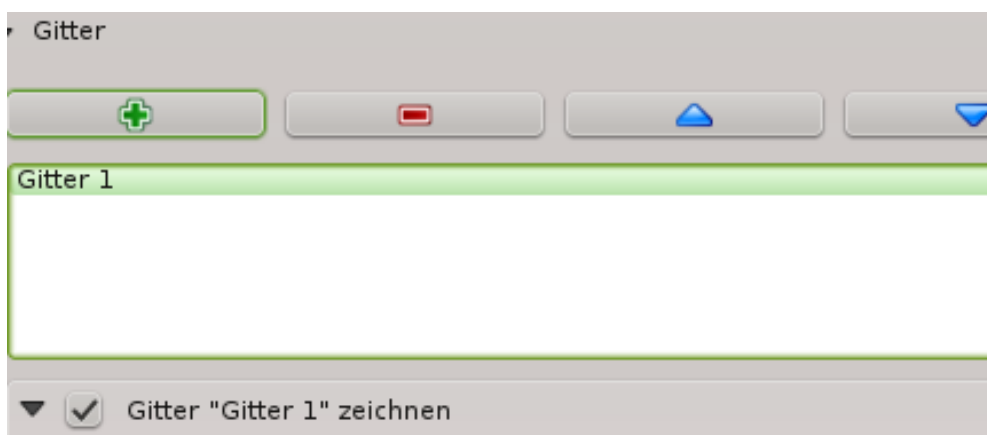


Abbildung 19.6: Kartengitterdialog 

Nachdem Sie ein Gitter hinzugefügt haben können Sie das Kontrollkästchen  *Gitter zeichnen* aktivieren um das Kartenelement mit einem Gitter zu überlagern. Rollen Sie diese Option aus um zu einer großen Menge von Konfigurationsoptionen zu gelangen, siehe [Figure\\_composer\\_map\\_4](#).

Als Gittertyp können Sie festlegen dass eine ausgefüllte Linie oder ein Kreuz verwendet werden. Die Darstellung des Gitters kann ausgewählt werden. Siehe Abschnitt [Rendering\\_Mode](#). Darüberhinaus können Sie ein Intervall in X und Y Richtung definieren, einen X- und Y-Versatz und die Breite die für den Kreuz- oder Liniengittertyp verwendet wird.

- Es gibt verschiedene Optionen um den Rahmen der die Karte hält zu gestalten. Folgende Optionen sind erhältlich: Kein Rahmen, Zebra, Markierungen innen, Markierungen außen, Innere und äußere Markierungen und Liniengrenze.
- Der erweiterte Darstellungsmodus steht auch für Gitter zur Verfügung (siehe Abschnitt [Rendering\\_mode](#)).
- Das  *Koordinaten zeichnen* Kontrollkästchen ermöglicht es Ihnen dem Kartenrahmen Koordinaten hinzuzufügen. Die Beschriftung kann innerhalb oder außerhalb des Kartenrahmens gezeichnet werden. Die Beschriftungsrichtung kann als horizontal, vertikal, horizontal und vertikal oder Grenzrichtung für jede Grenze individuell definiert werden. Die Einheiten können in Metern oder Grad sein. Schließlich können Sie die Gitterfarbe, die Schriftart, den Abstand zum Kartenrahmen und die Koordinatengenauigkeit definieren.

## Übersichten

Der *Übersichten* Dialog des *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten zur Verfügung:

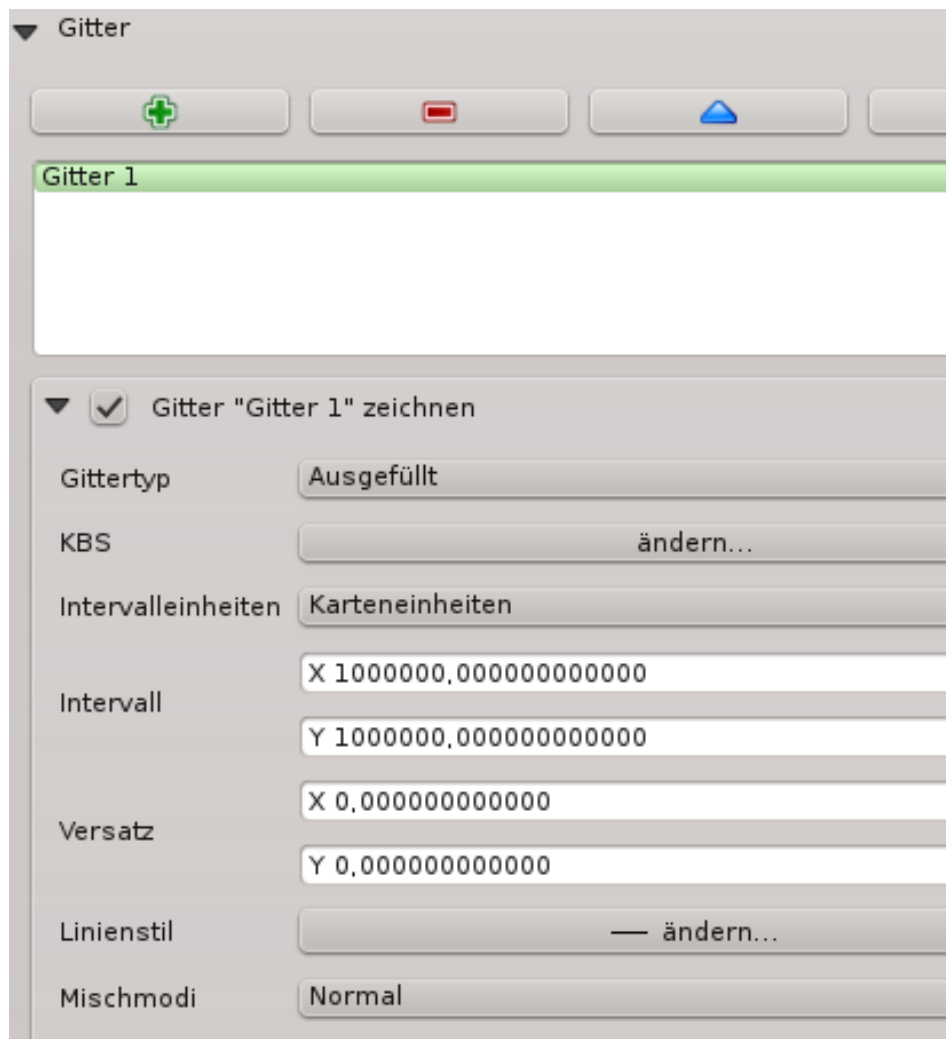


Abbildung 19.7: Gitter zeichnen Dialog 



Abbildung 19.8: Gitterrahmendialog 

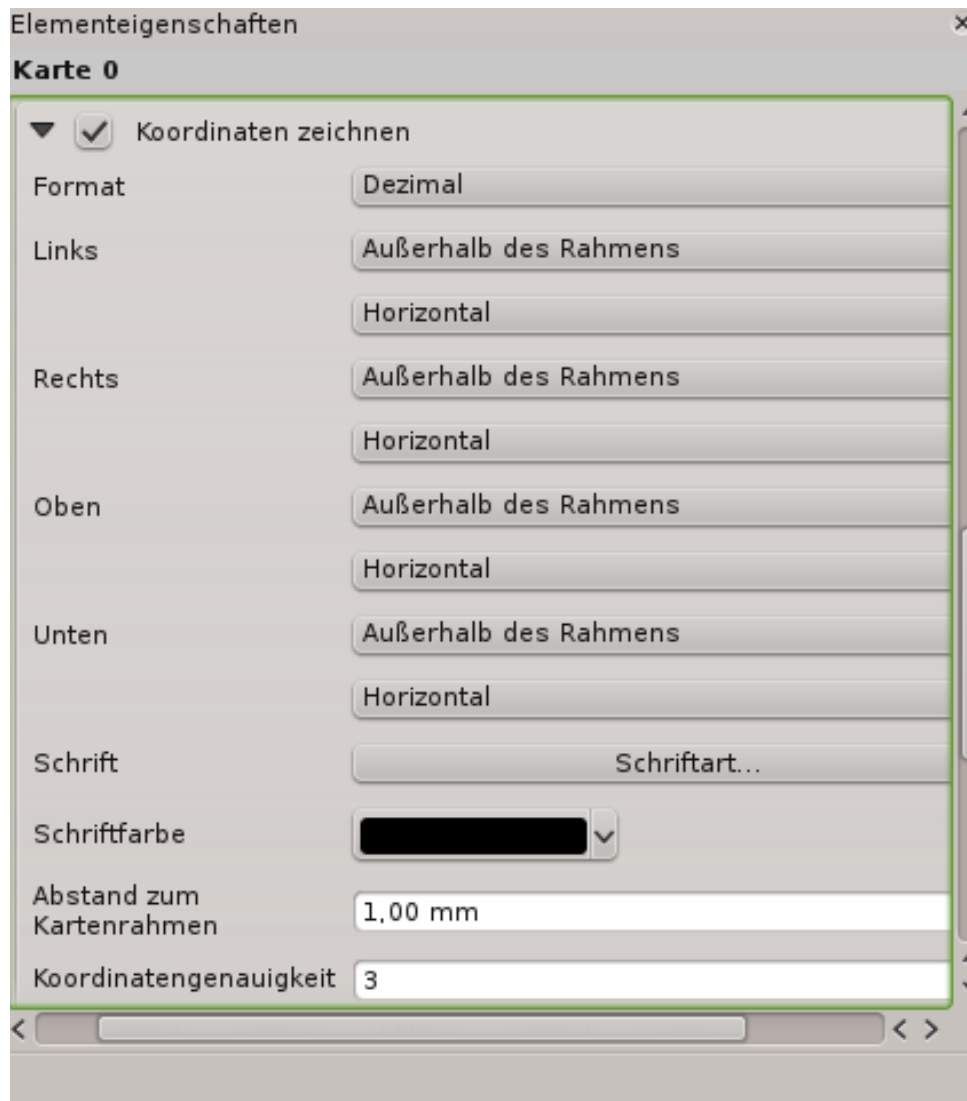


Abbildung 19.9: Gitter-Koordinaten-Zeichnen-Dialog 

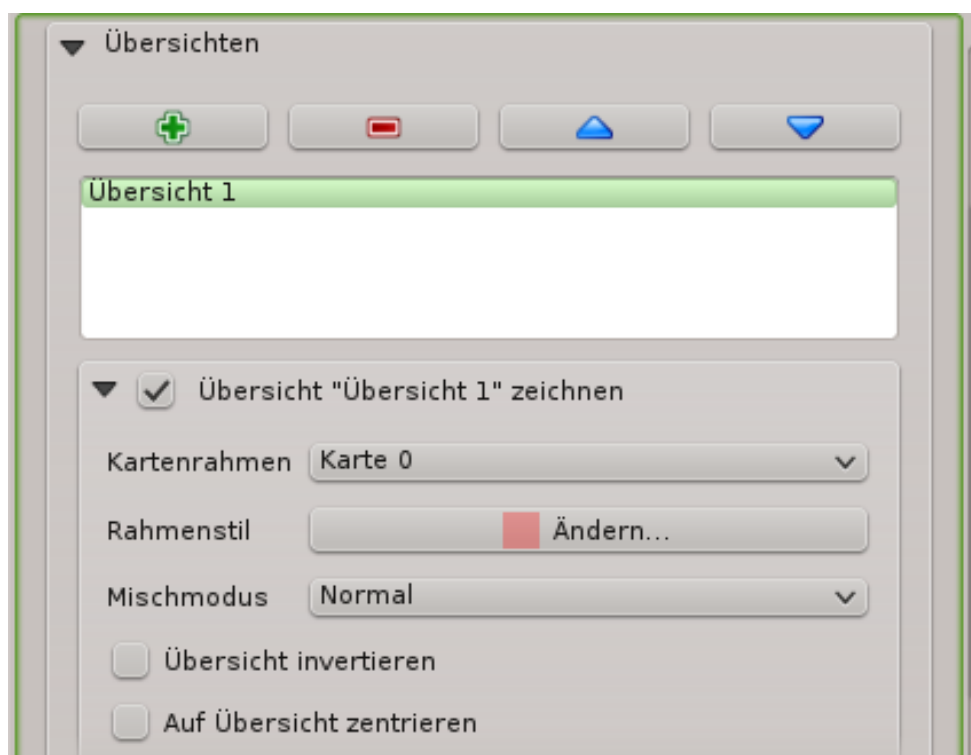


Abbildung 19.10: Kartenübersichtendialog 

Sie können sich entschließen eine Übersichtskarte zu erstellen welche die Ausmaße anderer Karten, die in der Zusammenstellung zur Verfügung stehen, zeigen. Als erstes müssen Sie die Karte(n), die sie in die Übersichtskarte einfügen wollen, erstellen. Als nächstes erstellen Sie die Karte die Sie wollen, um sie als Übersichtskarte zu benutzen, genau wie eine normale Karte.


- Mit dem Plus- und Minusknopf können Sie Übersichten hinzufügen oder entfernen.
- Mit dem Rauf- und Runterknopf können Sie eine Übersicht in der Liste verschieben und die Zeichenpriorität setzen.

Öffnen Sie *Übersichten* und drücken Sie den grünen Plusknopf um eine Übersicht hinzuzufügen. Zunächst wird diese Übersicht 'Übersicht 1' genannt (siehe [Figure\\_composer\\_map\\_7](#)). Sie können den Namen verändern wenn Sie auf das Übersichtelement in der Liste genannt 'Überblick 1' doppelklicken und ihm einen anderen Namen geben.

Wenn Sie da Übersichtelement in der Liste auswählen können Sie es anpassen.

- `:guilabel:Übersicht '<name_übersicht>'` muss aktiviert sein um die Ausmaße des gewählten Kartenrahmens zu zeichnen.
- Die *Übersichtsrahmen* Auswahlliste verweist auf das Kartenelement, dessen Ausmaße auf das aktuelle Kartenelement gezeichnet werden.
- Der *Rahmenstil* ermöglicht es Ihnen die Übersichtsrahmenfarbe zu ändern.
- Der *Mischmodus* ermöglicht verschiedene Transparenzmischmodi einzustellen. Siehe [Rendering\\_Mode](#).
- *Übersicht invertieren* erstellt eine Maske um die Ausmaße wenn es aktiviert ist: die referenzierte Kartenausmaße werden deutlich angezeigt, währenddessen alles andere mit einer Rahmenkarte ausgeblendet wird.
- *Auf Übersicht zentrieren* setzt die Ausmaße des Übersichtsrahmens in die Mitte der Übersichtskarte. Sie können nur eine Übersicht zum Zentrieren aktivieren wenn Sie mehrere Übersichten hinzugefügt haben.

## 19.3.2 Das Beschriftungselement

Um eine Beschriftung hinzuzufügen klicken Sie das  **Neue Beschriftung hinzufügen** Icon, platzieren Sie das Element mit der linken Maustaste auf der Druckzusammenstellungsansicht und positionieren und passen Sie sein Aussehen im Beschriftung *Elementeigenschaften* Reiter an.

Der *Elementeigenschaften* Reiter eines Beschriftungselements stellt die folgenden Funktionalitäten für das Beschriftungselement zur Verfügung (siehe [Figure\\_composer\\_label](#)):

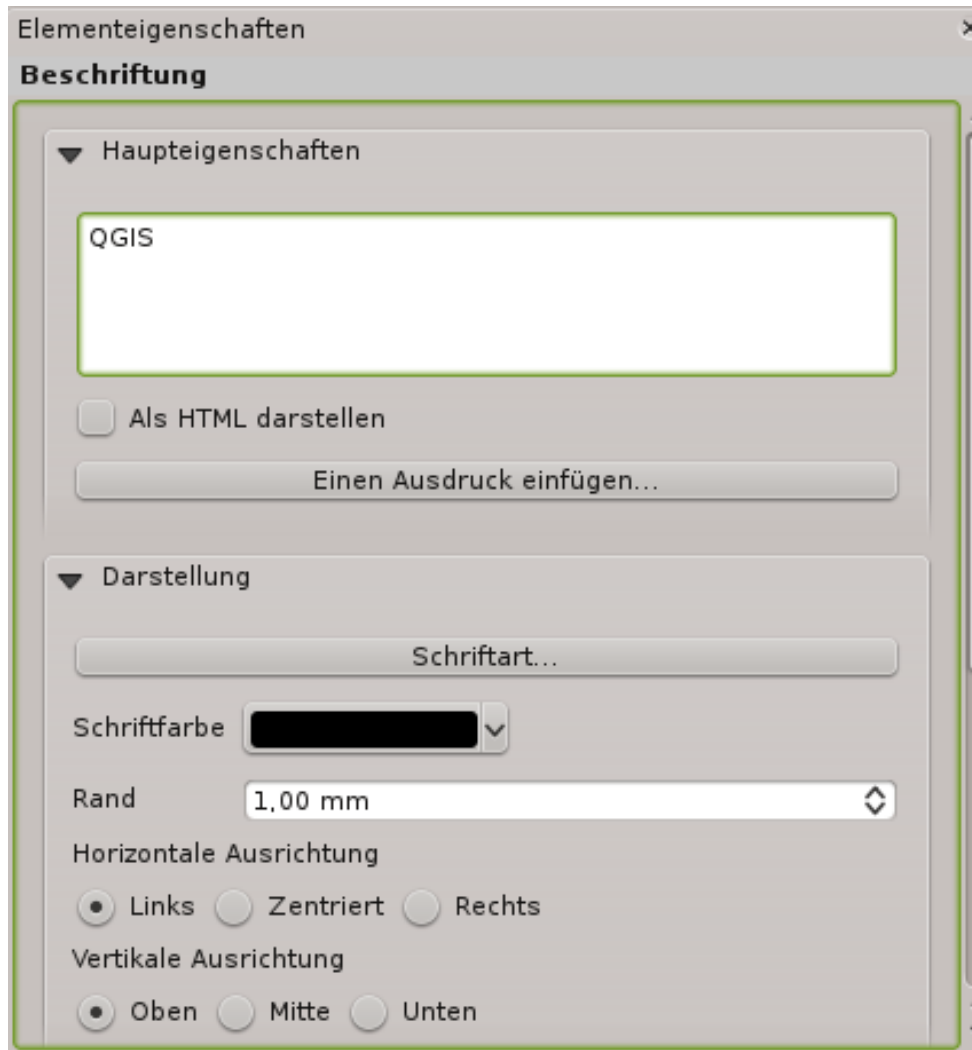


Abbildung 19.11: Beschriftungselementeigenschaften Dialog 

### Haupteigenschaften

- Im Haupteigenschaften Dialog wird der Text (HTML oder nicht) oder der Ausdruck, der benötigt wird um die Beschriftung auszufüllen, zur Zusammenstellungsansicht hinzugefügt.
- Beschriftungen können als HTML-Code interpretiert werden: aktivieren Sie  *Als HTML darstellen*. Sie können jetzt eine URL, ein klickbares Bild, das zu einer Webseite führt oder etwas komplexeres eingeben.
- Sie können auch einen Ausdruck eingeben. Klicken Sie auf [**Einen Ausdruck einfügen**] um einen neuen Dialog zu öffnen. Erstellen Sie einen Ausdruck, indem Sie auf die auf der linken Seite zur Verfügung stehenden Funktionen klicken. Zwei spezielle Kategorien können nützlich sein, insbesondere im Zusammenhang


mit der Atlasfunktionalität: Geometriefunktionen und Datensatzfunktionen. Unten wird eine Vorschau des Ausdrucks gezeigt.

- Definieren Sie *Schrift* indem Sie auf den **\*\*[Schriftart ...]** Knopf klicken oder eine *Schriftfarbe* indem Sie eine Farbe mit Hilfe des Farbauswahlwerkzeuges auswählen.

## Ausrichtung und Anzeigen



- Sie können die horizontale und vertikale Ausrichtung im Bereich *Ausrichtung* definieren.
- Im **Darstellung** Tag können Sie einen Rand in mm definieren. Dies ist der Rand von der Kante des Zusammenstellungselementes.

### 19.3.3 Das Bildelement

Um ein Bild hinzuzufügen klicken Sie das  **Bild hinzufügen** Icon, platzieren Sie das Element mit der linken Maustaste in der Druckzusammenstellungsanzeige und positionieren und passen Sie sein Aussehen im *Bild Elementeigenschaften* Reiter an.

Der *Bild Elementeigenschaften* Reiter stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_image\\_1](#)):

Sie müssen zuerst das Bild, das Sie anzeigen wollen auswählen. Es gibt mehrere Wege die *Bildquelle* in den **Haupteigenschaften** festzulegen.

1. Verwenden Sie den Suchknopf  von *Bildquelle* um eine Datei auf Ihrem Computer anhand des Suchdialogs auszuwählen. Der Browser starten in den SVG-Bibliotheken die mit QGIS zur Verfügung gestellt werden. Neben SVG können Sie auch noch andere Bildformate auswählen Sie `.png` oder `.jpg`.
2. Sie können die Quelle direkt in das *Bildquelle* Textfeld eingeben. Sie können für ein Bild sogar eine Remote URL-Adresse vergeben.
3. Im **Verzeichnisse durchsuchen** Bereich können Sie auch Bild durch *Lade Vorschauen ...* auswählen um die Bildquelle festzulegen.
4. Verwenden Sie den datendefinierten Knopf  um die Bildquelle aus einer Aufnahme festzulegen oder anhand eines regulären Ausdrucks festzulegen.

Mit der *Größenanpassungsmodus* Option können Sie festlegen wie das Bild dargestellt wird wenn der Rahmen verändert wird oder sich entschließen den Rahmen des Bildelementes in der Größe zu verändern so dass er auf die Originalgröße des Bildes passt.

Sie können einen der folgenden Modi auswählen:

- Zoom: Vergrößert das Bild im Rahmen wobei das Seitenverhältnis des Bildes beibehalten wird.
- Strecken: Streckt das Bild damit es in den Rahmen passt, ignoriert das Seitenverhältnis.
- Zuschneiden: Verwenden Sie diesen Modus nur für Rasterbilder, es setzt die Größe des Bildes auf die Original-Bildgröße ohne zu skalieren und der Rahmen wird verwendet das Bild ausschneiden, so dass nur der Teil des Bildes, der sich innerhalb des Rahmens befindet sichtbar ist.
- Rahmen zoomen und Größe anpassen: Vergrößert das Bild damit es in den Rahmen passt und verändert dann die Größe des Rahmens damit er zum Ergebnisbild passt.
- Rahmen auf Bildgröße setzen: Setzt die Größe des Rahmens damit der zur Originalgröße des Bildes passt ohne dass skaliert wird.


Der ausgewählte Größenanpassungsmodus kann die Elementoptionen 'Platzierung' und 'Bildrotation' deaktivieren. *Bildrotation* ist aktiv für die Größenanpassungsmodi 'Zoom' und 'Zuschneiden'.


Mit *Platzierung* können Sie die Position des Bildes innerhalb seines Rahmens auswählen. Der **Verzeichnisse durchsuchen** Bereich ermöglicht es ihnen Verzeichnisse mit Bildern im SVG-Format der Bilddatenbank



Abbildung 19.12: Bildelementeigenschaften Reiter 

hinzuzufügen und sie zu entfernen. Eine Vorschau der Bilder, die in den ausgewählten Verzeichnissen gefunden werden wird in einem Fenster gezeigt und kann dazu verwendet werden um die Bildquelle auszuwählen und zu setzen.

Bilder können mit dem *Bilddrehung* Feld gedreht werden. Das Aktivieren des  *Mit Karte abgleichen* Kontrollkästchens synchronisiert die Drehung eines Bildes in der QGIS Kartenansicht (z.B. ein gedrehter Nordpfeil) mit dem entsprechenden Druckzusammenstellungsbild.

Es ist auch möglich einen Nordpfeil direkt auszuwählen. Sie wählen zuerst einen Nordpfeil aus den **Verzeichnisse durchsuchen** aus und verwenden dann den Suchknopf  des Feldes *Bildquelle*, Sie können jetzt einen der Nordpfeile aus der Liste auswählen wie in [figure\\_composer\\_image\\_2](#) dargestellt.

**Bemerkung:** Vielen Nordpfeilen ist keine 'N' zugefügt, dieses ist mit Absicht so gehalten für Sprachen die kein 'N' für Norden verwenden, so dass sie einen anderen Buchstaben verwenden können.

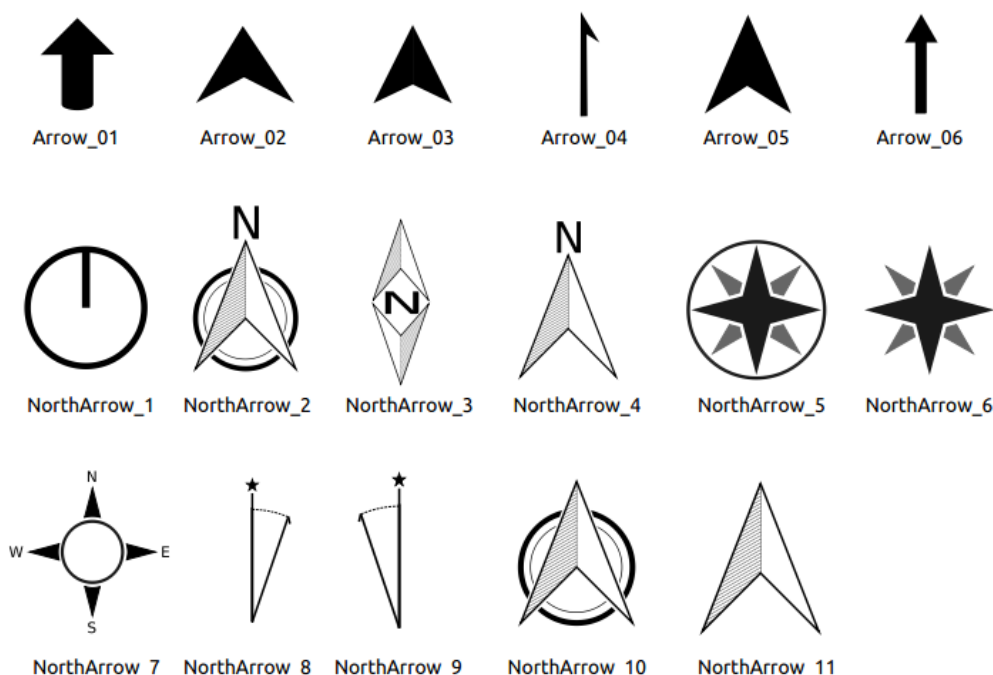



Abbildung 19.13: Für die Auswahl zur Verfügung stehende Nordpfeile, die von der SVG-Bibliothek zur Verfügung gestellt werden.

### 19.3.4 Das Legendenelement

Um ein Kartenelement hinzuzufügen, klicken Sie das  *Neue Legende hinzufügen* Icon, platzieren Sie das Element mit der linken Maustaste auf der Druckzusammenstellungsansicht und positionieren und passen Sie das Aussehen der Legende im *Elementeigenschaften* Reiter.

Die *Elementeigenschaften* eines Legendenelement-Reiters stellen die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_legend\\_1](#)):

#### Haupteigenschaften

Der *Haupteigenschaften* Dialog des Legenden *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_legend\\_2](#)):

In den Haupteigenschaften können Sie:



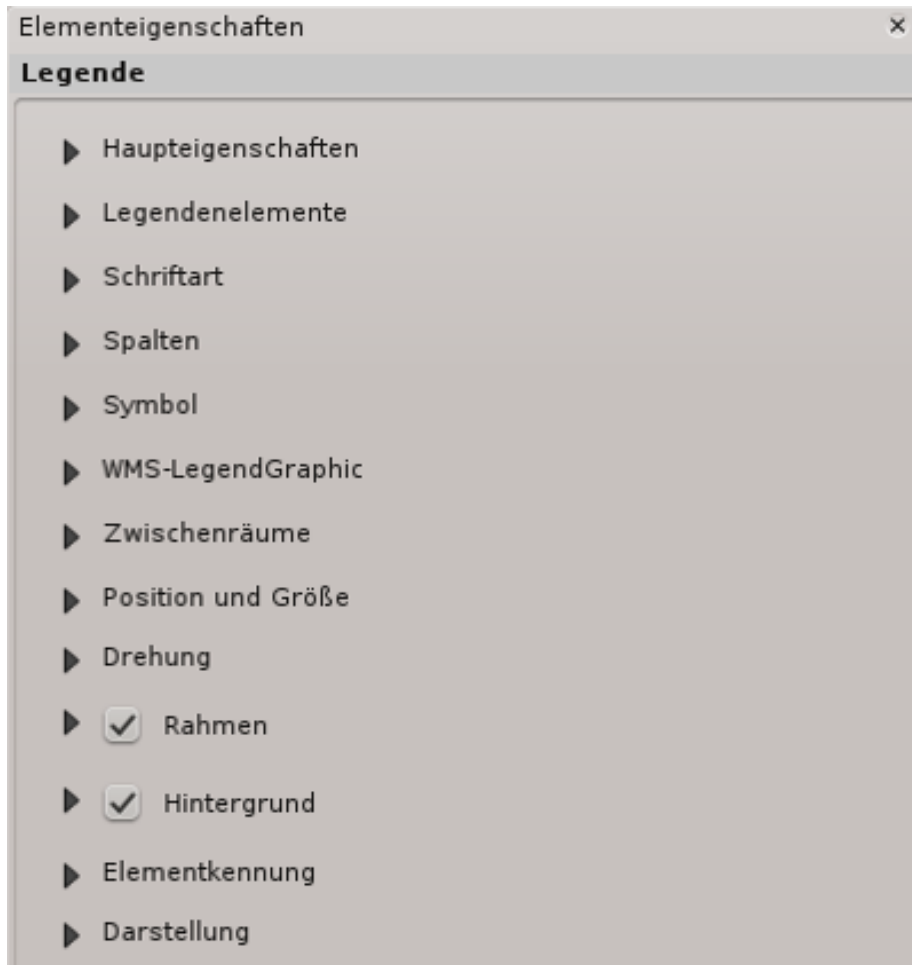



Abbildung 19.14: Legendenelementeigenschaften Reiter 



Abbildung 19.15: Legenden Haupteigenschaften Dialog 

- Den Titel der Legende.
- Die Titelausrichtung auf Links, Zentriert oder Rechts setzen.
- Sie können in der Liste auswählen auf welches *Karte* Element sich die aktuelle Legende bezieht.
- Sie können den Text des Legendentitels an einem vorgegebenen Zeichen umbrechen.

## Legendenelemente

Der **guilable:Legendenelemente** Dialog des Legendens *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_legend\\_3](#)):



Abbildung 19.16: Legendenelemente Dialog 🐧

- Die Legende wird automatisch geupdatet wenn  *Automatisch aktualisieren* aktiviert ist. Wenn *Automatisch aktualisieren* deaktiviert ist gibt dieses mehr Kontrolle über die Legendenelemente. Die Icons unter den Legendenelementenliste werden aktiviert.
- Das Legendenelementfenster führt alle Legendenelement auf und Sie können die Elementreihenfolge ändern, Layer gruppieren und Elemente in der Liste wiederherstellen, Namen bearbeiten und einen Filter hinzufügen.
  - Die Elementreihenfolge kann verändert werden mit **[Hoch]** und **[Runter]** Knöpfen oder mit der ‘Drag-und-Drop’ Funktionalität. Bei WMS Legendengrafiken kann die Reihenfolge nicht verändert werden.
  - Verwenden Sie den **[Gruppe hinzufügen]** Knopf um einen Legendengruppe hinzuzufügen.
  - Verwenden Sie den **[plus]** und **[minus]** Knopf um Layer hinzuzufügen oder zu entfernen.
  - Der **[Bearbeiten]** Knopf wird verwendet um den Layer, Gruppen- oder Titelnamen zu bearbeiten, zuerst müssen Sie das Legendenelement auswählen.
  - Der **[Sigma]** Knopf fügt eine Objektanzahl für jeden Vektorlayer hinzu.
  - Verwenden Sie den **[Legende nach Karteninhalt filtern]** Knopf um die Legende nach dem Karteninhalt zu filtern, nur die in der Karte sichtbaren Element werden in der Legende aufgeführt.

Nachdem Sie den Stil im QGIS Hauptfenster geändert haben, können Sie auf **[Alle aktualisieren]** klicken um die Änderungen im Legendenelement der Druckzusammenstellung anzupassen.

## Schriftarten, Spalten, Symbol

Die *Schriftarten*, *Spalten* und *Symbol* Dialoge des Legendens *Elementeigenschaften* Reiters stellen die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_legend\\_4](#)):

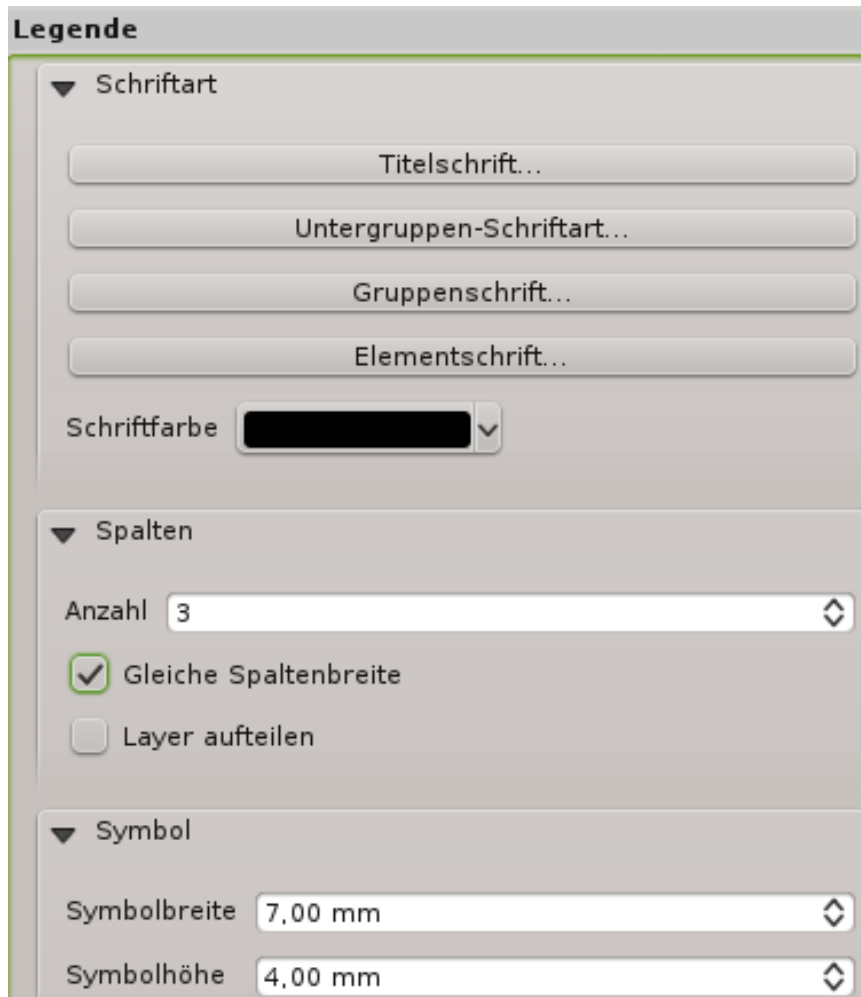


Abbildung 19.17: Legenden Schriftart, Spalten, Symbol und Zwischenräume Dialoge 

- Sie können die Schriftart des Legendentitels, der Gruppe, der Untergruppe und des Elements (Layers) ändern. Klicken Sie auf einen Kategorieknopf um einen **Schriftart auswählen** Dialog zu öffnen.
- Sie können die Beschriftungen mit einer **Farbe** versehen indem Sie die erweiterte Farbauswahl verwenden, die ausgewählte Farbe jedoch wird an alle Schriftelemente in der Legende vergeben.
- Legendenelemente können in mehrere Spalten angeordnet werden. Setzen Sie die Anzahl von Spalten im *Anzahl*  Feld.
  - *Gleiche Spaltenbreite* stellt ein, wie Legendenspalten angepasst werden sollen.
  - Die  *Layer aufteilen* Option ermöglicht es eine kategorisierte oder abgestufte Layerlegende in Spalten aufzuteilen.
- Sie können die Breite und Höhe des Legendensymbols in diesem Dialog ändern.

### WMS-LegendGraphic und Zwischenräume

Die Dialoge *WMS-LegendGraphic* und *Zwischenräume* Dialoge der Legenden *Elementeigenschaften* Reiters stellen die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_legend\\_5](#)):




Abbildung 19.18: WMS-LegendGraphic Dialoge 

Wenn Sie einen WMS Layer hinzugefügt haben und Sie fügen ein Legendenzusammenstellungselement ein wird ein Request an den WMS Server gesendet um eine WMS Legende zur Verfügung zu stellen. Diese Legende wird nur angezeigt wenn der WMS Server die GetLegendGraphic Capability übermitteln. Der WMS Legendeneinhalt wird als Rasterbild zur Verfügung gestellt.

*WMS legendGraphic* wird verwendet um die *Legendenbreite* und die *Legendenhöhe* des WMS Legendeneinhalt anzupassen.

Die Zwischenräume um Titel, Gruppe, Untergruppe, Symbol, Iconbeschriftung, Rahmenabstand oder Spaltenabstand können über diesen Dialog angepasst werden.

### 19.3.5 Das Maßstabselement

Um einen Maßstab hinzuzufügen klicken Sie das  Neuen Maßstab hinzufügen Icon, platzieren Sie das Element mit der linken Maustaste auf der Druckzusammenstellungsansicht und positionieren und passen Sie das Aussehen im Maßstab *Elementeigenschaften* Reiter an.

Die *Elementeigenschaften* eines Maßstabselementes bieten die folgenden Funktionalitäten (siehe [figure\\_composer\\_scalebar\\_1](#)):

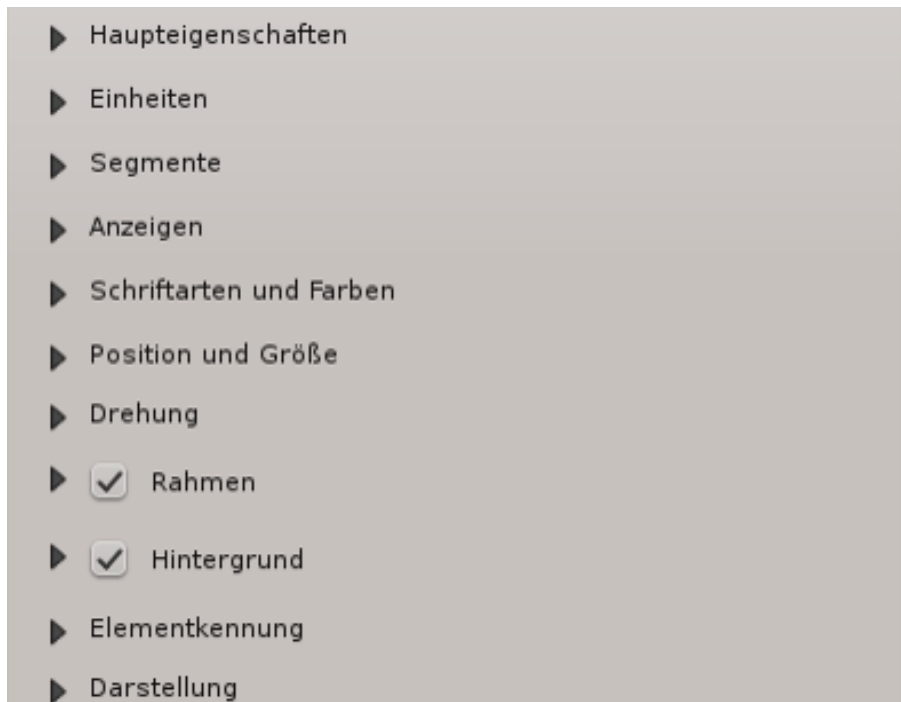


Abbildung 19.19: Maßstab Elementeigenschaften Reiter 

#### Haupteigenschaften

Der *Haupteigenschaften* Dialog des Maßstab *Elementeigenschaften* Reiters stellte die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_scalebar\\_2](#)):

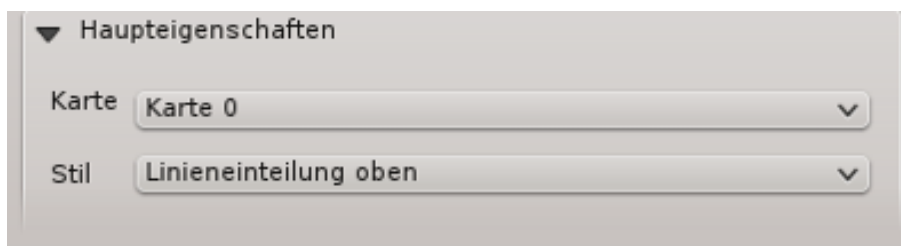


Abbildung 19.20: Maßstab Haupteigenschaften Dialog 

- Wählen Sie zuerst die Karte aus, an der der Maßstab befestigt werden soll.
- Wählen Sie dann den Stil des Maßstabs. Sechs Stile stehen zur Verfügung:

- **Einfacher Rahmen** und **Doppelter Rahmen** Stile, die eine oder zwei Linien von Rahmen mit abwechselnden Farben enthalten.
- **Mittige**, **Oben** oder **Unten** Linieneinteilung.
- **Numerisch**, wobei das Maßstabsverhältnis gedruckt wird (z.B. 1:50000).

## Einheiten und Segmente

Der *Einheiten* und *Segmente* Dialog des Maßstab *Elementeigenschaften* Reiters stellen die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_scalebar\\_3](#)):

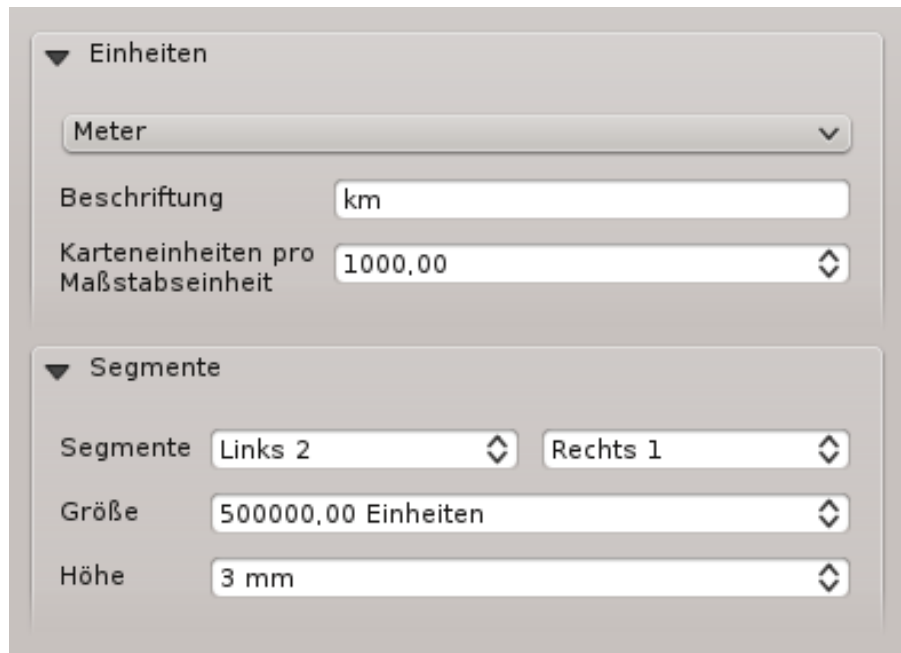



Abbildung 19.21: Maßstab Einheiten und Segmente Dialoge 

In diesen beiden Dialogen können Sie einstellen, wie der Maßstab dargestellt wird.

- Wählen Sie die verwendeten Karteneinheiten aus. Es gibt vier Möglichkeiten: **Karteneinheiten** ist die automatische Einheitenauswahl; **Meter**, **Fuß** oder **Seemeilen** erzwingen Einheitsumwandlungen.
- Das *Beschriftung* Feld definiert den Text, der verwendet wird, um die Einheiten des Maßstabs zu beschreiben.
- *Karteneinheit pro Maßstabseinheit* ermöglicht Ihnen das Verhältnis zwischen einer Karteneinheit und seiner Darstellung im Maßstab festzulegen.
- Sie können definieren wie viele *Segmente* links und rechts vom Maßstab gezeichnet werden und wie lang jedes Segment sein wird (*Größe* Feld). *Höhe* kann ebenfalls definiert werden.

## Anzeigen

Der *Anzeigen* Dialog des Maßstab *Elementeigenschaften* Reiters bietet die folgenden Funktionalitäten (siehe [figure\\_composer\\_scalebar\\_4](#)):

Sie können festlegen wie der Maßstab in seinem Rahmen dargestellt wird.

- *Rahmenrand*: Zwischenraum zwischen Text- und Rahmengrenzen
- *Beschriftungsrund*: Zwischenraum zwischen Text und Maßstabdarstellung
- *Linienbreite*: Linienbreite der Maßstabdarstellung

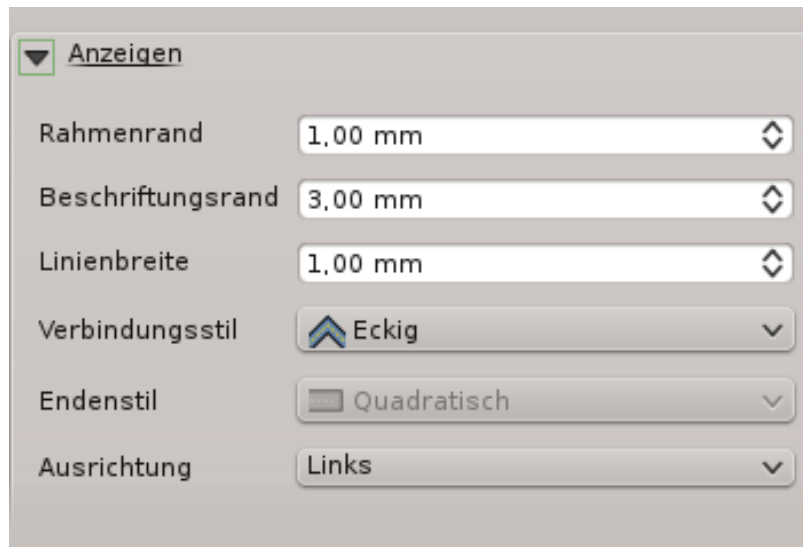


Abbildung 19.22: Maßstab Anzeigen 

- *Verbindungsstil*: Ecken am Ende des Maßstabs im Stil Rund, Abgerundet oder Eckig (nur für die Maßstabsstile Einfacher Rahmen & Doppelter Rahmen)
- *Endenstil*: Das Ende aller Linien im Stil Quadratisch, Rund Flach (nur erhältlich für die Maßstabsstile Linieneinteilung Oben, Unten und Mittig)
- *Ausrichtung*: Setzt Text auf auf die linke, mittlere oder rechte Seite des Rahmens (funktioniert nur für den Maßstabsstil Numerisch)

### Schriftarten und Farben

Der *Schriftarten und Farben* Dialog des Maßstab *Elementeigenschaften* Reiters bieten die folgenden Funktionalitäten (siehe [figure\\_composer\\_scalebar\\_5](#)):

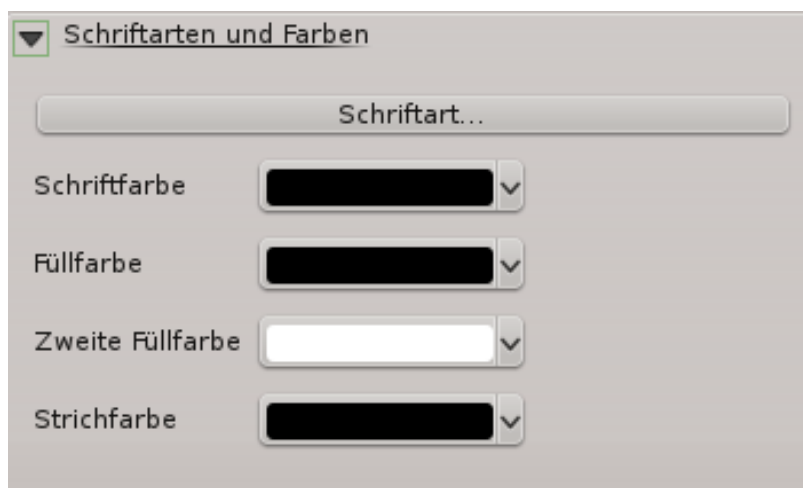


Abbildung 19.23: Maßstab Schriftarten und Farben Dialog 



Sie können die Schriftarten und Farben, die für den Maßstab verwendet werden definieren.

- Verwenden Sie den [**Schriftart ...**] Knopf um die Schriftart festzulegen.
- *Schriftfarbe*: setzen Sie die Schriftfarbe
- *Füllfarbe*: setzen Sie die erste Füllfarbe

- *Zweite Füllfarbe*: setzen Sie die zweite Füllfarbe
- *Strichfarbe*: legen Sie die Farbe der Linien des Maßstabs fest

Füllfarben werden nur für Maßstabsstile Einfacher Rahmen und Doppelter Rahmen verwendet. Um eine Farbe auszuwählen können Sie die Listenoption mit Hilfe des Dop-down-Pfeiles zum Öffnen einer einfachen Farbauswahloption oder der erweiterten Farbauswahloption verwenden, die gestartet wird wenn Sie in die farbige Box im Dialog klicken.

### 19.3.6 Die Grundformelemente

Um eine Grundform (Ellipse, Rechteck, Dreieck) hinzuzufügen, klicken Sie das  Ellipse hinzufügen Icon oder das  Pfeil hinzufügen Icon, platzieren Sie das Element mit dem Halten der linken Maustaste. Passen Sie das Aussehen im *Elementeigenschaften* Reiter an.

Wenn Sie auch die Umschalt Taste während des Platzierens der Grundform gerückt halten können Sie auch ein(en) perfektes Rechteck, Kreis oder Rechteck erstellen.

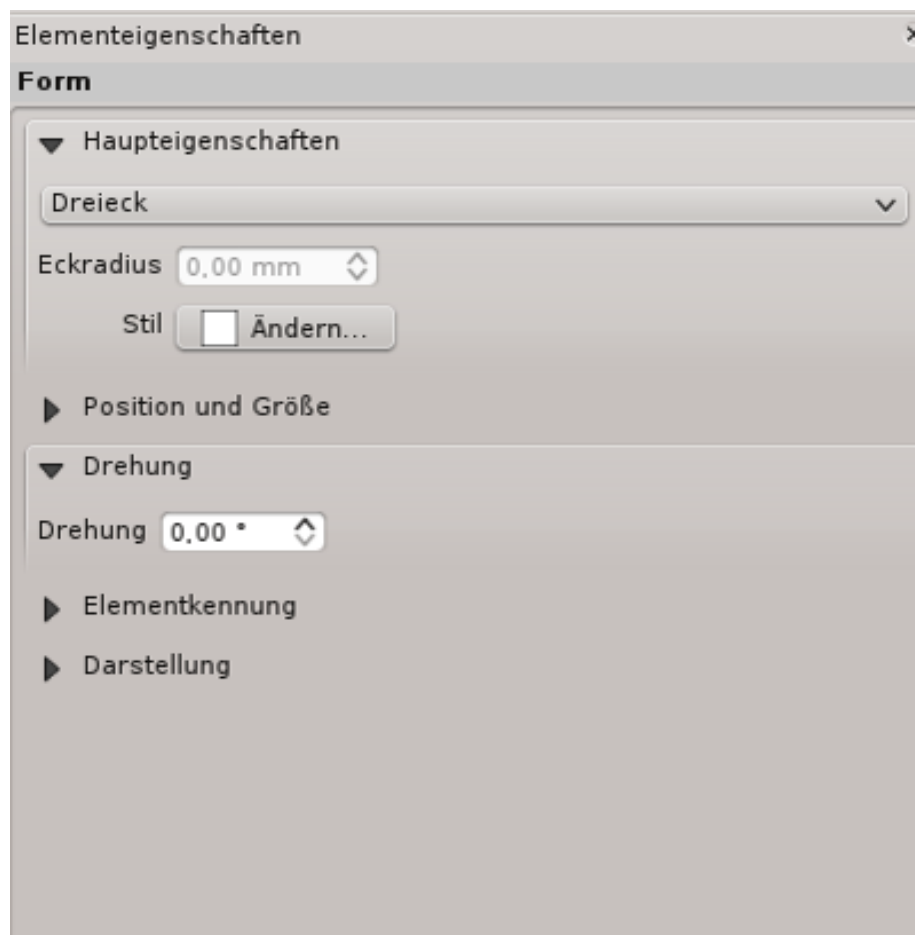


Abbildung 19.24: Formelementeigenschaften Reiter 

Der *Form* Elementeigenschaften Reiter ermöglicht es Ihnen auszuwählen wenn Sie eine Ellipse, Rechteck oder Dreieck innerhalb des vorgegebenen Rahmen zeichnen wollen.


Sie können den Stil der Form setzen indem Sie den erweiterten Symbolauswahldialog verwenden mit dem Sie seinen Randstil, Füllstil, Füllmuster und die Verwendung von Markierungen etc. definieren können.

Für die Rechteckform können Sie den Wert des Eckradius setzen um die Ecken abzurunden.



**Bemerkung:** Im Gegensatz zu anderen Elemente können Sie die Rahmen- oder die Hintergrundfarbe des Rahmens nicht setzen.

### 19.3.7 Das Pfeilelement

Um einen Pfeil hinzuzufügen klicken Sie das  Pfeil hinzufügen Icon, platzieren Sie das Element indem Sie die linke Maustaste drücken und zeichnen Sie einen Pfeil auf die Druckzusammenstellungsseite positionieren Sie ihn und passen Sie das Aussehen in den Pfeil *Elementeigenschaften* Reiter an.

Wenn Sie zusätzlich die Umschalt Taste beim Platzieren des Pfeils drücken wird er in einem Winkel von exakt 45° platziert.

Das Pfeilelement kann dazu verwendet werden eine Linie oder einen einfache Pfeil hinzuzufügen, der z.B. dafür verwendet werden kann um eine Beziehung zu anderen Druckzusammenstellungselementen herzustellen. Um einen Nordpfeil zu erstellen sollte man zuerst ein Bildelement in Betracht ziehen. QGIS besitzt einen Satz von Nordpfeilen im SVG Format. Des weiteren können Sie ein Bildelement mit einer Karte verbinden so dass es sich automatisch mit der Karte drehen kann (siehe [the\\_image\\_item](#)).

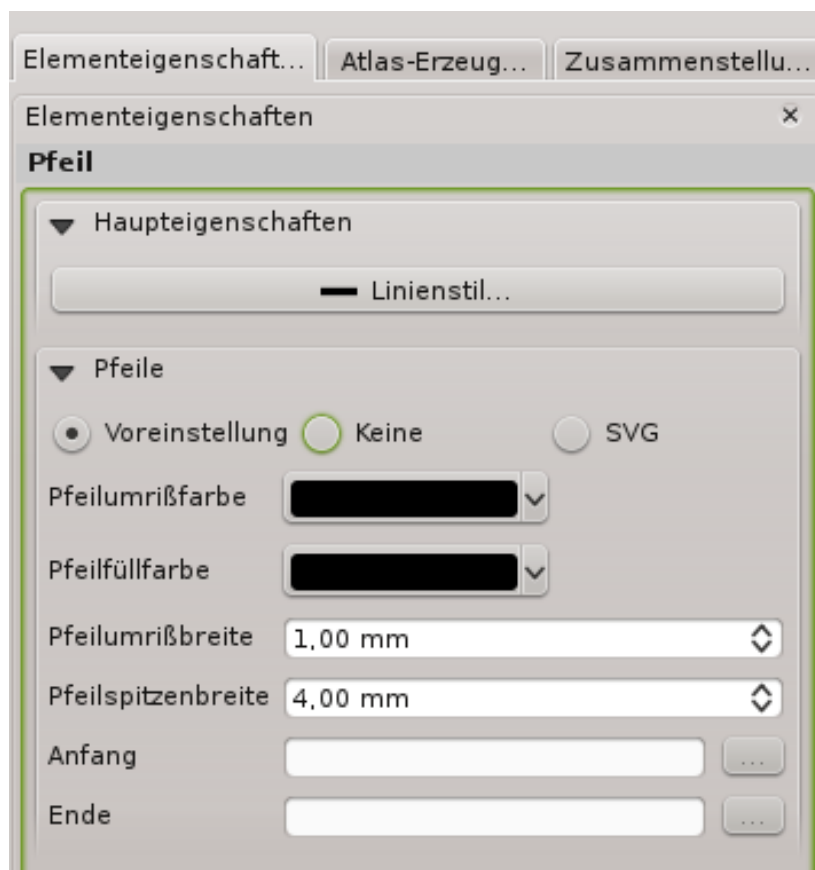



Abbildung 19.25: Pfeilelementeigenschaften Reiter 

#### Elementeigenschaften

Der *Pfeil* Elementeigenschaften Reiter ermöglicht es Ihnen ein Pfeilelement zu konfigurieren.

Der [**Linienstil ...**] Knopf kann dazu verwendet werden den Linienstil anhand einer Symbolauswahl festzulegen.

In *Pfeile* können Sie einen von drei Radiobuttons auswählen.

- *Voreinstellung*: zeichnet einen einfach Pfeil, vergibt eine Option um den Anfangsstil zu definieren

- *Keine*: Um eine Linie ohne Pfeilspitze zu zeichnen
- *SVG*: Um eine Linie mit einem SVG *Anfang* und/oder *Ende* zu zeichnen

Beim *Voreinstellung* Pfeilmarker können Sie die folgenden Optionen zum Gestalten der Pfeilspitze verwenden.


- *Pfeilumrissfarbe*: Setzt die Umrissfarbe des Pfeilkopfes
- *Pfeilfüllfarbe*: Setzt die Füllfarbe des Pfeilkopfes
- *Pfeilumrissbreite*: Setzt die Umrissbreite des Pfeilkopfes
- *Pfeilspitzenbreite*: Setzt die Größe der Pfeilspitze fest

Bei *SVG* können Sie die folgenden Optionen verwenden.

- *Anfang*: Suchen Sie ein SVG Bild aus das am Anfang der Linie gezeichnet wird
- *Ende*: Suchen Sie ein SVG Bild aus das am Ende der Linie gezeichnet wird
- *Pfeilspitzenbreite*: Legt die Größe des Start und/oder Headmarkers fest

SVG Bilder werden automatisch mit der Linie gedreht. Die Farbe des SVG Bildes kann nicht geändert werden.

### 19.3.8 Das Attributtabellelement

Es ist möglich Teile einer Vektorattributtabelle der Druckzusammenstellungsansicht hinzuzufügen: Klicken Sie das  Fügt Attributtabelle hinzu Icon, platzieren Sie das Element mit der linken Maustaste auf der Druckzusammenstellungsansicht und positionieren und passen Sie das Aussehen im *Elementeigenschaften* Reiter an.

Die *Elementeigenschaften* eines Attributtabellelement Reiters bieten die folgenden Funktionalitäten (siehe [figure\\_composer\\_table\\_1](#)):

#### Haupteigenschaften

Der *Haupteigenschaften* Dialog des Attributtabelle *Elementeigenschaften* Reiters bietet die folgenden Funktionalitäten (siehe [figure\\_composer\\_table\\_2](#)):

- Unter *Quelle* können Sie normalerweise nur 'Layerobjekte' auswählen
- Mit *Layer* können Sie aus Vektorlayern, die ins Projekt geladen sind, auswählen.
- Der Knopf [**Tabellendaten aktualisieren**] kann dazu verwendet werden die Tabelle zu aktualisieren wenn der tatsächliche Inhalt der Tabelle sich geändert hat.
- Der Knopf [**Attribute ...**] startet das *Attribute wählen* Menü, siehe [figure\\_composer\\_table\\_3](#), das dazu verwendet werden kann den sichtbaren Inhalt der Tabelle zu ändern. Nachdem die Änderungen vorgenommen wurden benutzen Sie den [**OK**] Knopf um die Änderungen auf die Tabelle anzuwenden.

Im *Spalten* Abschnitt können Sie:

- Ein Attribut entfernen, wählen Sie einfach eine Attributreihe indem Sie irgendwo in der Reihe klicken aus und drücken Sie den Minusknopf um das ausgewählte Atribut zu entfernen.
- Zum Hinzufügen eines Attributs verwenden Sie den Plusknopf. Am Ende erscheint eine neue leere Reihe und Sie können eine leere Zelle aus der Spalte *Attribut* auswählen. Sie können ein Feldattribut aus der Liste auswählen oder Sie können anhand eines regulären Ausdrucks ein neues Attribut erstellen.
- Benutzen Sie die Auf- und Ab-Pfeile um die Reihenfolge der Attribute in der Tabelle zu ändern.
- Wählen Sie eine Zelle in der Kopfzeile aus um den Kopf zu ändern, tippen Sie einfach einen neuen Namen ein.
- Wählen Sie eine Zelle in der Ausrichtungspalte und Sie können zwischen Links, Zentriert oder Rechts wählen.

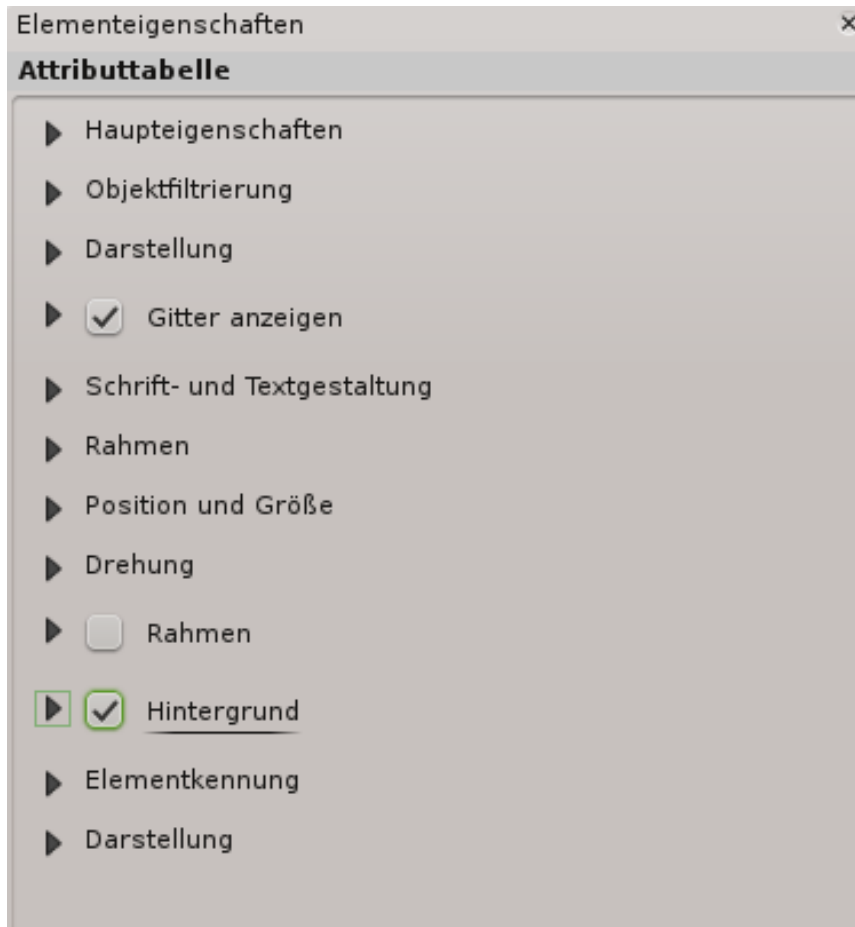


Abbildung 19.26: Attributtabellelementeigenschaften Reiter 

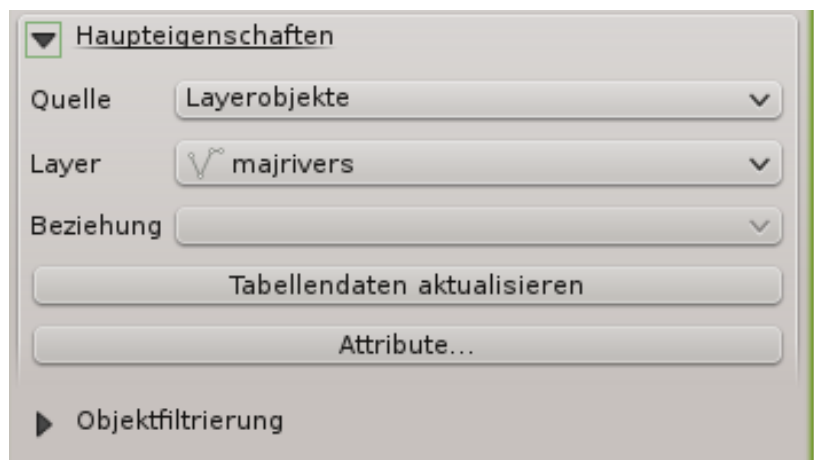


Abbildung 19.27: Attributtabellen Haupteigenschaften Dialog 

- Wählen Sie eine Zelle in der Breitespalte und Sie können von Automatisch zu einer Breite in mm wechseln, geben Sie einfach eine Nummer ein. Wenn Sie zurück zu automatisch wechseln wollen, verwenden Sie das Kreuz.
- Der [**Urücksetzen**] Knopf kann immer verwendet werden um zu den originalen Einstellungen zurückzuwechseln.

Im *Sortierung* Abschnitt können Sie:

- Ein Attribut, mit dem die Tabelle sortiert werden soll, hinzufügen. Wählen Sie ein Attribut aus und legen Sie die Sortierreihenfolge auf 'Aufsteigend' oder Absteigend' fest und drücken Sie den Plusknopf. Es wird eine neue Linie der Sortierungsreihenfolgenliste hinzugefügt.
- wählen Sie eine Reihe in der Liste und verwenden Sie die Auf- und Abknöpfe um den Sortierungsprioritätslevel auf Attributniveau zu ändern.
- verwenden Sie den Minusknopf um ein Attribut von der Sortierungsreihenfolgenliste zu entfernen

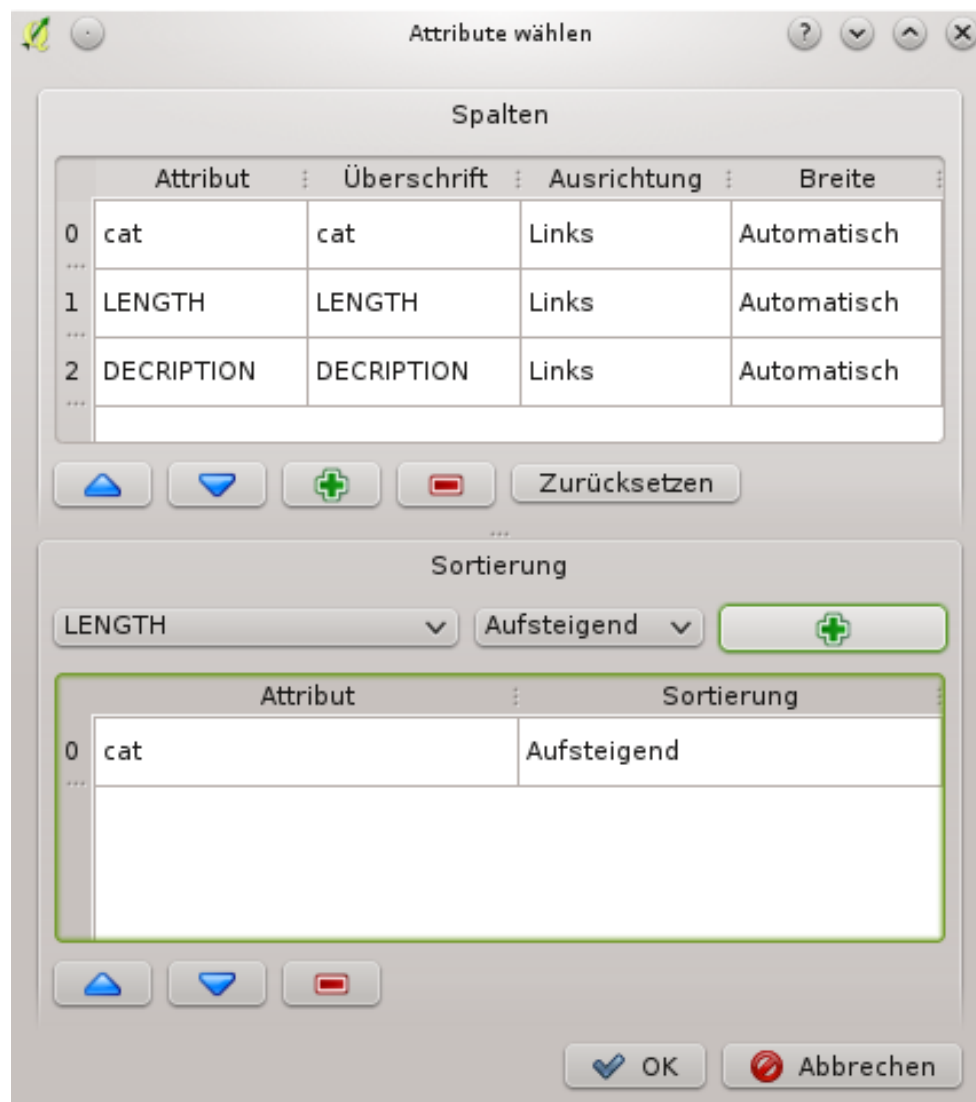


Abbildung 19.28: Attributtabelle Attribute wählen Dialog 

## Objektfiltrierung

Der *Objektfiltrierung* Dialog des Attributtabelle *Elementeigenschaften* Reiters bietet die folgenden Funktionalitäten (siehe [figure\\_composer\\_table\\_4](#)):

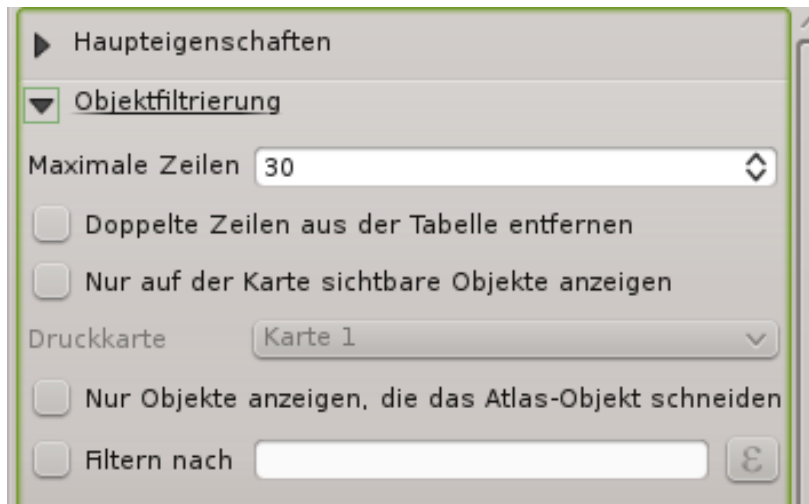



Abbildung 19.29: Attributtabelle Objektfiltrierung Dialog 

Sie können:

- Die *Maximalen Zeilen*, die dargestellt werden sollen, definieren.
- *Doppelte Zeilen aus der Tabelle entfernen* aktivieren um nur eindeutige Datensätze zu zeigen.
- Aktivieren Sie  *Nur auf der Karte sichtbare Objekte zeigen* und wählen Sie die entsprechende *Druckkarte* aus um die nur die Attribute von in der ausgewählten Karte sichtbaren Objekten anzuzeigen.
- Das Aktivieren des  *Nur Objekte zeigen, die das Atlasobjekt schneiden* ist nur möglich wenn  *Einen Atlas erzeugen* aktiviert ist. Wenn es aktiviert ist zeigt es eine Tabelle nur mit den Objekten, die auf der Karte der bestimmten Seite des Atlas gezeigt werden.
- Aktivieren Sie  *Filtern nach* und schaffen Sie einen Filter indem Sie eine Eingabezeile eingeben oder einen regulären Ausdruck über den gegebenen Ausdrucksknopf einfügen. Einige Beispiele von Filteranweisungen, die Sie verwenden können wenn Sie den airports Layer aus dem Beispieldatensatz geladen haben
  - ELEV > 500
  - NAME = ' ANIAK'
  - NAME NOT LIKE 'AN%
  - regexp\_match( attribute( \$currentfeature, 'USE' ) , '[i]')

Der letzte reguläre Ausdruck wird nur die Flughäfen einfügen die einen Buchstaben 'i' in der Attributspalte 'USE' haben.

## Darstellung

Die *Darstellung* Dialoge des Attributtabelle *Elementeigenschaften* Reiter stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_table\\_5](#)):

- Mit *Zellenränder* können Sie die Ränder um den Text in jeder Zelle der Tabelle definieren.
- Mit *Kopf anzeigen* können Sie von einer Liste eine der voreingestellten Optionen 'Im ersten Rahmen', 'In allen Rahmen' oder 'Kein Kopf' auswählen.
- Die Option *Leere Tabellen* kontrolliert was dargestellt wird wenn die Ergebnisauswahl leer ist.
  - **Nur Kopf zeichnen** zeichnet nur den Kopf ausser Sie haben 'Kein Kopf' bei *Kopf anzeigen* ausgewählt.

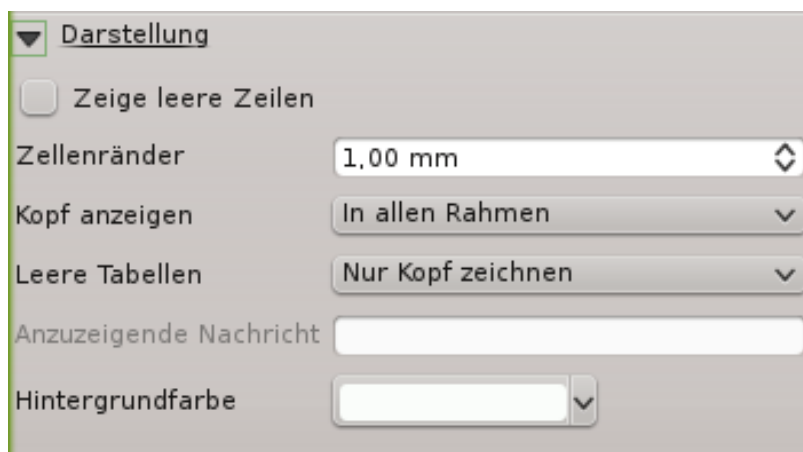


Abbildung 19.30: Attributtabelle Darstellung Dialog 

- **Ganze Tabelle ausblenden** zeichnet nur den Hintergrund der Tabelle. Sie können  *Hintergrund nicht anzeigen wenn Rahmen leer ist* in *Rahmen* aktivieren um die Tabelle komplett auszublenden.
- **Zeige leere Zeilen** füllt die Attributtabelle mit leeren Zellen, diese Option kann auch verwendet werden um zusätzliche leere Zellen zu schaffen wenn Sie ein Ergebnis anzeigen wollen!
- **Eingestellte Nachricht anzeigen** zeichnet den Kopf und fügt eine Zelle ein, die sich über alle Spalten erstreckt und eine Nachricht wie 'Kein Ergebnis' kann in der Option *Anzuzeigende Nachricht* bereitgestellt werden
- Die Option *Anzuzeigende Nachricht* ist nur aktiviert wenn Sie **Eingestellte Nachricht anzeigen** bei *Leere Tabelle* gewählt haben. Die angegebene Nachricht wird in der Tabelle in der ersten Zeile gezeigt wenn das Ergebnis eine leere Tabelle ist.
- Mit *Hintergrundfarbe* können Sie die Hintergrundfarbe der Tabelle festlegen.

## Gitter anzeigen

Der *Gitter anzeigen* Dialog des Attributtabelle *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten zur Verfügung (see [figure\\_composer\\_table\\_6](#)):

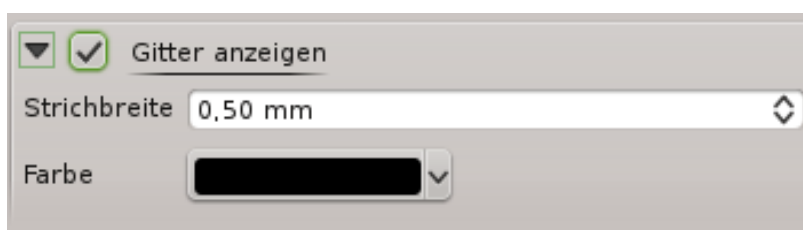


Abbildung 19.31: Attributtabelle Gitter anzeigen Dialog 

- Aktivieren Sie  *Gitter anzeigen* wenn Sie das Gitter, die Umrandungen der Tabellenzellen anzeigen wollen.
- Mit *Strichbreite* können Sie die Dicke der Linien, die im Gitter verwendet werden, festlegen.
- Die *Farbe* des Gitters kann anhand des Farbauswahldialogs eingestellt werden.

## Schrift- und Textgestaltung

Der *Schrift- und Textgestaltung* dialog des Attributtabelle *Elementeigenschaften* Reiter stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_table\\_7](#)):

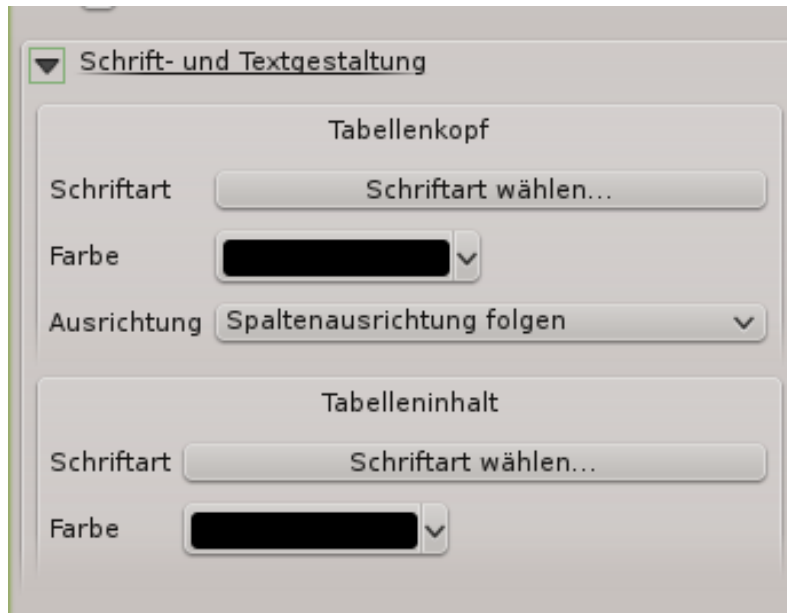


Abbildung 19.32: Attributtabelle Schrift- und Textgestaltung Dialog 

- Sie können *Schriftart* und *Farbe* für *Tabellenkopf* und *Tabelleninhalt* definieren.
- Für den *Tabellenkopf* können Sie zusätzlich die Ausrichtung‘ einstellen und zwischen *Spaltenausrichtung folgen*, *Links*, *Zentriert* und *Rechts* wählen. Die Spaltenausrichtung wird anhand des *Attribute wählen* Dialogs eingestellt (siehe [Figure\\_composer\\_table\\_3](#)).

## Rahmen

Der *Rahmen* Dialog des Attributtabelle *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_table\\_8](#)):

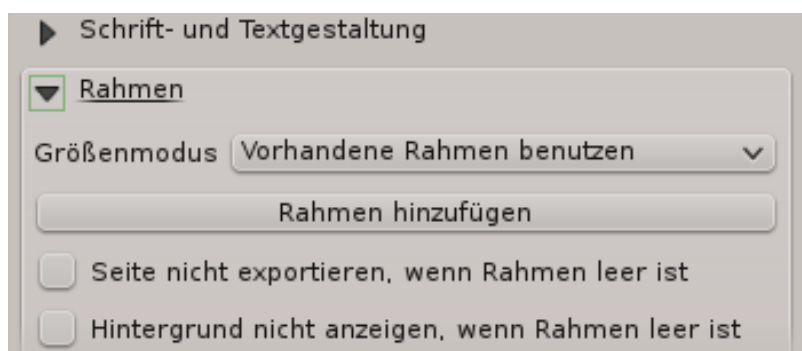



Abbildung 19.33: Attributtabelle Rahmen Dialog 

- Mit dem *Größenmodus* können Sie auswählen wie der Inhalt der Attributtabelle gerendert wird:
  - *Vorhandene Rahmen benutzen* stellt das Ergebnis nur im ersten Rahmen und hinzugefügten Rahmen dar.

- *Auf nächste Seite erweitern* erstellt so viele Rahmen (und entsprechende Seiten) wie nötig um die vollständige Auswahl der Attributtabelle darstellen zu können. Jeder Rahmen can im Layout bewegt werden. Wenn Sie die Größe eines Rahmens verändern wird die resultierende Tabelle auf andere Rahmen aufgeteilt. Der letzte Rahmen wird so zu geschnitten dass die Tabelle hineinpasst.
- *Bis zum Ende wiederholen* erstellt so viele Rahmen wie die *Auf nächste Seite erweitern* Option, nur dass alle Rahmen die gleiche Größe haben.
- Verwenden Sie den [**Rahmen hinzufügen**] Knopf um einen weiteren Rahmen mit der Größe wie der ausgewählte Rahmen hinzuzufügen. Das Ergebnis der Tabelle das nicht in den ersten Rahmen passt wird im nächsten Rahmen wiederholt wenn Sie den Größenmodus *Vorhandene Rahmen benutzen* verwenden.
- Das Aktivieren von  *Seite nicht exportieren wenn der Rahmen leer ist* verhindert dass die Seite exportiert wird wenn der Rahmen keinen Inhalt hat. Das heißt dass alle anderen Zusammenstellungselemente wie Elemente, Karten, Maßstäbe, Legenden etc. im Ergebnis nicht sichtbar sind.
- Das Aktivieren von  *Hintergrund nicht anzeigen wenn Rahmen leer ist* verhindert dass der Hintergrund gezeichnet wird wenn der Rahmen keinen Inhalt hat.

### 19.3.9 Das HTML-Rahmen Element

Es ist möglich einen Rahmen der den Inhalt einer Webseite anzeigt einzufügen oder sogar Ihre eigene HTML-Seite zu erstellen und sie anzuzeigen!

Klicken Sie das  HTML-Rahmen hinzufügen Icon, platzieren Sie das Element indem Sie das Rechteck mit gedrückter linker Maustaste auf der Druckzusammenstellungsseite aufziehen und positionieren und gestalten Sie die Erscheinung im *Elementeigenschaften* Reiter (siehe [figure\\_composer\\_html\\_1](#)).

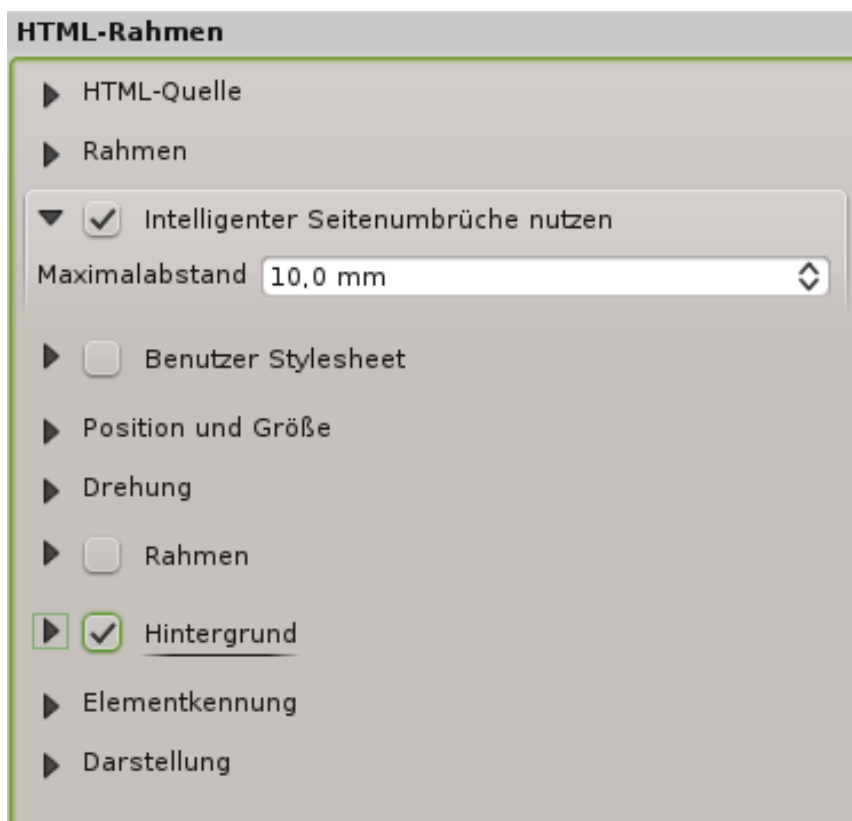



Abbildung 19.34: HTML-Rahmen, der Elementeigenschaften Reiter 





## HTML-Quelle

Als HTML-Quelle können Sie entweder eine URL setzen und den URL-Radiobutton aktivieren oder die HTML-Quelle direkt in die Textbox, die zur Verfügung gestellt wird, eingeben und den Quelle-Radiobutton aktivieren.

Der *HTML-Quelle* Dialog der HTML-Rahmen *Elementeigenschaften* Reiter stellt die folgenden Funktionalitäten zur Verfügung (siehe [figure\\_composer\\_html\\_2](#)):



Abbildung 19.35: HTML-Rahmen, die HTML-Quelle Eigenschaften 

- In *URL* können Sie die URL einer Webseite, die Sie aus Ihrem Internetbrowser kopiert haben eingeben oder eine HTML-Datei auswählen indem Sie den Suchknopf  verwenden. Es gibt zusätzlich die Option den datendefinierte Übersteuerung Knopf zu verwenden um eine URL von den Inhalten eines Attributfeldes einer Tabelle zu benutzen oder einen regulären Ausdruck zu verwenden.
- Unter *Quelle* können Sie Text mit einigen HTML-Tags ins Textfenster eingeben oder eine ganze HTML-Seite zur Verfügung zu stellen.
- Der [**Einen Ausdruck einfügen**] Knopf kann dazu verwendet werden einen Ausdruck wie [%Year(\$now) %] die das Quelle Textfenster einzugeben um das aktuelle Jahr darzustellen. Dieser Knopf ist nur aktive wenn der Radiobutton *Quelle* ausgewählt ist. Klicken Sie nach dem Eingeben des Ausdrucks irgendwo in das Textfenster bevor Sie den HTML-Rahmen auffrischen, sonst geht der Ausdruck verloren.
- Aktivieren Sie  *QGIS-Ausdrücke in HTML-Quelle auswerten* um das Ergebnis des Ausdrucks, den Sie eingefügt haben, zu sehen, andernfalls sehen Sie den Ausdruck.
- Verwenden Sie den [**HTML aktualisieren**] Knopf um den HTML-Rahmen zu aktualisieren und das Ergebnis der Änderung zu sehen.

## Rahmen

Der *Rahmen* Dialog des HTML-Rahmen *Elementeigenschaften* Reiters stellt die folgenden Funktionalitäten zur Verfügung (see [figure\\_composer\\_html\\_3](#)):

- Mit dem *Resize-Modus* können Sie auswählen wie der HTML Inhalt gerendert wird:
  - *Vorhandene Rahmen benutzen* stellt das Ergebnis nur im ersten Rahmen und hinzugefügten Rahmen dar.

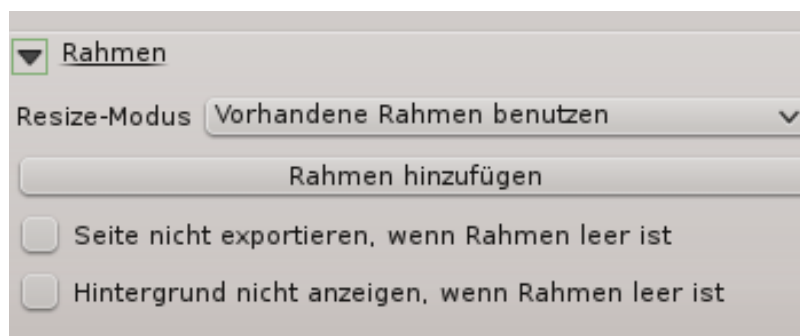


Abbildung 19.36: HTML-Rahmen, die Rahmeneigenschaften 

- *Auf nächste Seite erweitern* erstellt so viele Rahmen (und entsprechende Seiten) wie nötig um die Höhe der Webseite zu rendern. Jeder Rahmen kann im Layout verschoben werden. Wenn Sie die Größe eines Rahmens ändern, wird die Webseite auf die anderen Rahmen aufgeteilt. Der letzte Rahmen wird auf die Webseite zugeschnitten.
- *Auf jeder Seite wiederholen* wiederholt den oberen linken Abschnitt der Webseite auf jeder Seite in Rahmen der gleichen Größe.
- *Bis zum Ende wiederholen* erstellt so viele Rahmen wie die *Auf nächste Seite erweitern* Option, nur dass alle Rahmen die gleiche Größe haben.
- Verwenden Sie den [**Rahmen hinzufügen**] Knopf um einen weiteren Rahmen mit der gleichen Größe wie der ausgewählte Rahmen einzufügen. Wenn die HTML-Seite nicht in den ersten Rahmen passt wird sie im nächsten Rahmen weitergehen wenn Sie *Resize-Modus* oder *Vorhandene Rahmen benutzen* verwenden.
- Aktivieren Sie  *Seite nicht exportieren, wenn Rahmen leer ist* verhindert dass das Kartenlayout exportiert wird wenn der Rahmen keinen HTML-Inhalt hat. Das heißt, dass alle Zusammenstellungselemente, Karten, Maßstäbe, Legenden etc. im Ergebnis nicht sichtbar sind.
- Das Aktivieren von  *Hintergrund nicht anzeigen, wenn Rahmen leer ist* verhindert, dass der HTML-Rahmen gezeichnet wird wenn der Rahmen leer ist.

### Intelligente Seitenumbrüche und Benutzer Stylesheet nutzen

Der *Intelligente Seitenumbrüche* Dialog und *Benutzer Stylesheet* Dialog des HTML-Rahmen *Elementeigenschaften* Reiter bietet die folgenden Funktionalitäten (siehe [figure\\_composer\\_html\\_4](#)):

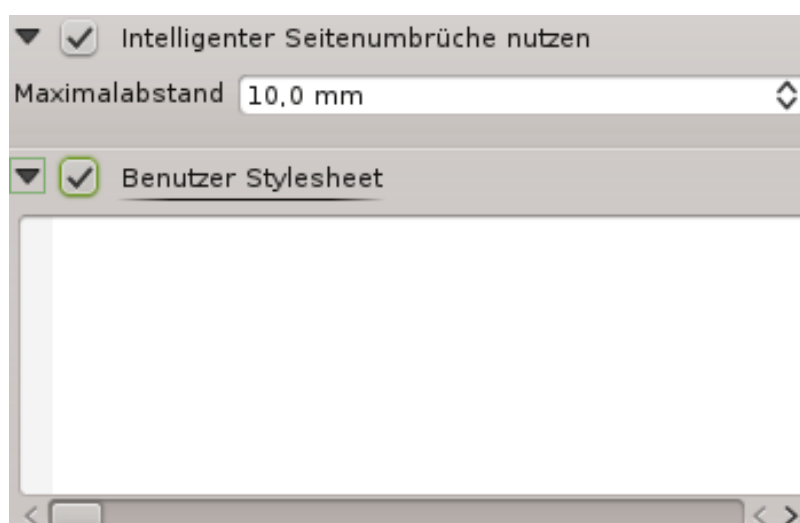





Abbildung 19.37: HTML-Rahmen, Intelligente Seitenumbrüche und Benutzer Stylesheet Eigenschaften 


- Aktivieren Sie  *Intelligente Seitenumbrüche* um zu verhindern, dass der HTML-Rahmen inmitten einer Textzeile umbricht so dass er weiter schön und glatt im nächsten Rahmen weitergeht.
- Setzen Sie den erlaubten *Maximalabstand* wenn ausgerechnet werden soll wo die Seitenumbrüche im HTML platziert werden sollen. Dieser Abstand ist der maximale Größe des leeren Raums am Fuß eines Rahmens nachdem der optimale Platz für den Seitenumbruch berechnet wurde. Das Setzen eines größeren Wertes resultiert in einer besseren Auswahl des Platzes für den Seitenumbruch, es resultiert aber auch in mehr nichtgenutztem Platz am Fuß des Rahmens. Dies wird nur genutzt wenn *Intelligenter Seitenumbrüche nutzen* aktiviert ist.
- Aktivieren Sie  *Benutzer Stylesheet* um HTML-Styles, die oft in Cascading Style Sheets bereitgestellt werden, anzuwenden. Ein Beispiel von Stylecode um die Farbe des `<h1>` Header Tags in grün zu setzen und die Schrift und Schriftgröße von Text, der in Paragraph Tags `<p>` enthalten ist, zu setzen.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
font-size: 20px;
}
```

- Verwenden Sie den **[HTML aktualisieren]** Knopf um das Ergebnis der Stylesheeteinstellungen zu sehen.


## 19.4 Elemente verwalten

### 19.4.1 Größe und Position

Jedes Element in der Zusammenstellung kann verschoben/in der Größe verändert werden um ein perfektes Layout zu erstellen. Bei beiden Operationen ist der erste Schritt das  **Eintrag wählen/verschieben** Werkzeug zu aktivieren und auf das Element zu klicken; Sie können es dann verschieben, indem Sie die linke Maustaste gedrückt halten. Wenn Sie die Bewegungen auf die horizontale oder vertikale Achse einschränken wollen, halten Sie einfach `Umschalt` während Sie die Maus bewegen. Wenn Sie eine höhere Genauigkeit brauchen können Sie das ausgewählte Element mit den Pfeiltasten auf der Tastatur verschieben; wenn die Bewegung zu langsam ist können Sie dies mit `Halten` von `Umschalt` verbessern.

Ein ausgewähltes Element zeigt Quadrate an seinen Grenzen; das Verschieben mit der Maus ändert die Größe des Elements in der entsprechenden Richtung. Beim Ändern der Größe behält man mit dem `Halten` von `Umschalt` das Seitenverhältnis. Das `Halten` von `Alt` ändert die Größe von der Elementmitte aus.

Die richtige Position für ein Element kann erzielt werden wenn man es an Führungen ausrichtet oder schlaue Führungen benutzt. Führungen werden gesetzt wenn man in den Führungen klickt und verschiebt. Führungen werden durch Klicken in das Lineal, auf eine Ebene mit der Führung gehen und verschieben an einen neuen Platz verschoben. Um eine Führung zu entfernen ziehen Sie sie außerhalb der Seite. Wenn Sie das an Führungen ausrichten deaktivieren wollen halten Sie einfach `Strg` während Sie die Maus bewegen.

Sie können mehrere Element auf einmal mit dem  **Eintrag wählen/verschieben** Knopf wählen. Halten Sie einfach den `Umschalt` Knopf und klicken Sie auf alle Elemente, die Sie brauchen. Sie können bei dieser Gruppe dann die Größe verändern oder Sie wie ein einziges Element verschieben.


Wenn Sie einmal die richtige Position für ein Element gefunden haben können Sie es sichern indem Sie die Elemente in der Werkzeugleiste verwenden oder die Box neben dem Element im *Elemente* Bedienfeld ticken. Gesicherte Elemente können auf der Seite **nicht** ausgewählt werden.

Gesicherte Elemente können entsichert weren indem man das Element im *Elemente* Bedienfeld auswählt und die Tickbox deaktiviert oder indem Sie die Icons in der Werkzeugleiste verwenden.

Um ein Element zu deselektieren klicken Sie einfach darauf während Sie die `Umschalt` Taste halten.

Innerhalb des *Bearbeiten* Menüs können Sie Aktionen um alle Elemente auszuwählen, um eine Auswahl aufzuheben und um die aktuelle Auswahl zu invertieren finden.

## 19.4.2 Ausrichtung

Anheben oder Absenkenfunktionalitäten für Elemente befinden sich innerhalb des  Ausgewählte Elemente in den Vordergrund bringen Pull-down-Menü. Suchen Sie ein Element auf der Druckzusammenstellungsseite und wählen sie die zutreffende Funktionalität um das ausgewählte Element verglichen mit anderen Elementen anzuheben oder abzusenken (siehe [table\\_composer\\_1](#)). Diese Reihenfolge wird im *Elemente* Bedienfeld gezeigt. Sie können ein Objekt auch anheben oder absenken im *Elemente* Bedienfeld indem Sie auf eine Objektbeschriftung klicken und diese in der Liste verschieben.

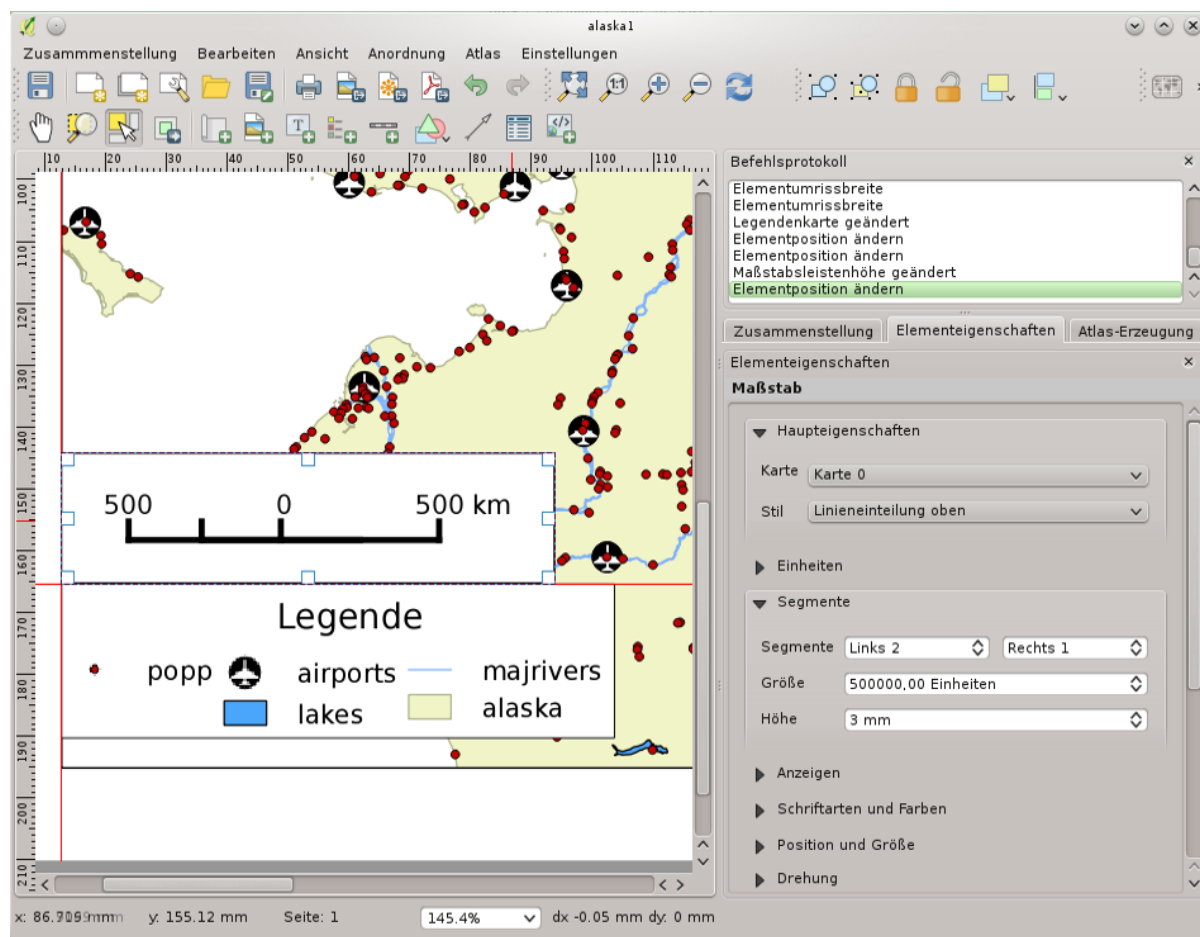



Abbildung 19.38: Ausrichtungshilfslinien in der Druckzusammenstellung 

Es gibt mehrere Ausrichtungsfunktionalitäten im  Alle gewählten Elemente linksbündig ausrichten Pull-downmenü (siehe [table\\_composer\\_1](#)). Um eine Ausrichtungsfunktionalität zu verwenden suchen Sie zuerst einige Elemente aus und klicken Sie dann auf das zutreffende Ausrichtungssymbol. Alle ausgewählten Elemente werden dann an Ihrem gemeinsamen Begrenzungsrahmen ausgerichtet.



## 19.4.3 Elemente Kopieren/Ausschneiden und Einfügen

Die Druckzusammenstellung enthält Aktionen um die gemeinsame Kopieren/Ausschneiden/Einfügen Funktionalität für die Elemente im Layout zu verwenden. Wie gewöhnlich müssen Sie erst die Elemente anhand einer der oben zu sehenden Optionen auswählen; an dieser Stelle können die Aktionen im *Bearbeiten* Menü gefunden werden. Wenn Sie die Einfügen Aktion verwenden werden die Elemente gemäß der aktuellen Mausposition eingefügt.

**Bemerkung:** HTML-Elemente können auf diese Weise nicht kopiert werden. Um dies zu umgehen benutzen Sie den **[Rahmen hinzufügen]** Knopf im *Elementeigenschaften* Reiter.


## 19.5 Schritte rückgängig machen und wiederherstellen

Während des Layoutens ist es möglich Änderungen zurückzunehmen und wiederherzustellen. Dies kann mit den Zurücknehmen und Wiederherstellen Werkzeugen getan werden:

-  Letzte Änderung zurücknehmen
-  Letzte Änderung wiederherstellen

Dies kann auch durch einen Mausklick innerhalb des *Befehlsprotokoll* Bedienfeldes durchgeführt werden (siehe [figure\\_composer\\_29](#)).




Abbildung 19.39: Befehlsprotokoll in der Druckzusammenstellung 

## 19.6 Atlas-Erzeugung

Die Druckzusammenstellung enthält Erstellungsfunktionen, die es Ihnen möglich machen ein Kartenbuch auf automatisierte Weise zu erstellen. Das Konzept ist es einen Abdeckungslayer zu verwenden, der Geometrien und Felder enthält. Für jede Geometrie im Abdeckungslayer wird eine neue Ausgabe erstellt in der der Inhalt einer Karte verschoben wird um die aktuelle Geometrie hervorzuheben. Felder die mit diesen Geometrien in Zusammenhang stehen können innerhalb von Beschriftungen verwendet werden.

Jede Seite wird mit jedem Objekt erzeugt. Um die Erstellung eines Atlas zu aktivieren und die Erstellungsparameter zu erreichen wechseln Sie zum *Atlas-Erzeugung* Reiter. Dieser Reiter enthält die folgenden Widgets (siehe [Figure\\_composer\\_atlas](#)):

- *Einen Atlas erzeugen*, was die Atlaserstellung aktiviert oder deaktiviert.
- Ein *Abdeckungslayer*  Kombobox mit der Sie einen (Vektor)Layer, der die Geometrien, über denen eine Iteration durchgeführt werden soll, enthält.
- Ein optionaler  *Abdeckungslayer verstecken* das, wenn es aktiviert ist, den Abdeckungslayer (aber nicht die anderen) während der Erstellung versteckt.
- Ein optionales *Filtern nach* Textfeld das es Ihnen möglich macht einen Ausdruck zum Filtern von Objekten aus dem Abdeckungslayer anzugeben. Wenn der Ausdruck nicht leer ist werden nur Objekte auf die `True` zutrifft ausgewählt. Mit dem Knopf auf der rechten Seite können Sie den Ausdrucksbasierten Filter aufrufen.
- Ein *Ausgabedateiausdruck* Textfeld, das dazu benutzt wird einen Dateinamen für jede Geometrie, falls benötigt, zu erzeugen. Es beruht auf Ausdrücken. Dieses Feld ist nur von Bedeutung wenn mehrere Dateien gerendert werden sollen.

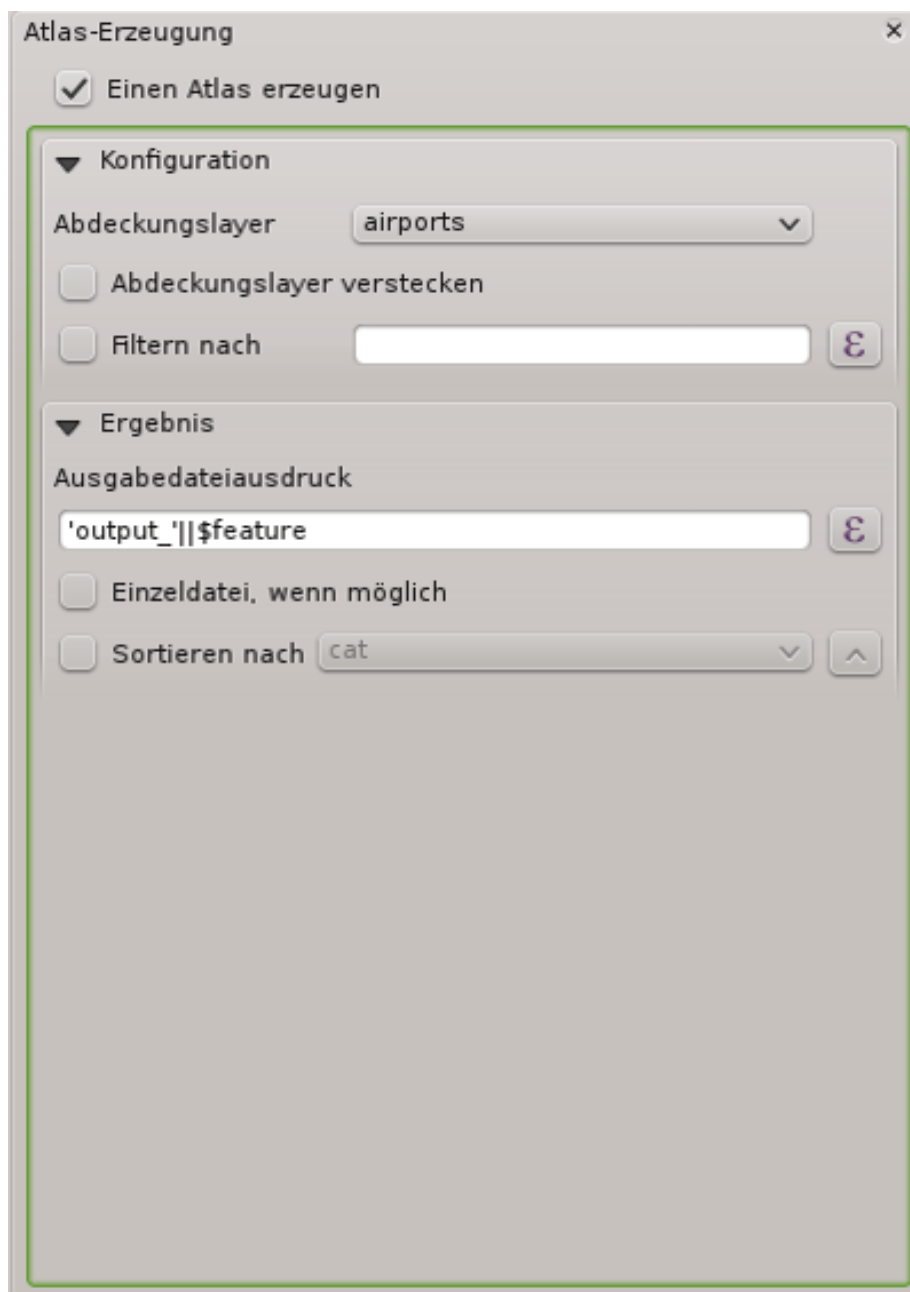



Abbildung 19.40: Atlas-Erzeugung Reiter 

- Ein  *Einzeldatei wenn möglich* ermöglicht die Erstellung einer einzelnen Datei wenn dies möglich ist mit dem ausgewählten Ausgabeformat (PDF beispielsweise). Wenn dieses Feld aktiviert ist, wird der Wert des *Ausgabedateiausdruck* nicht berücksichtigt.
- Ein optionales  *Sortieren nach* ermöglicht es Ihnen, falls aktiviert, Objekte des Abdeckungslayers zu sortieren. Die entsprechende Kombobox ermöglicht es Ihnen auszuwählen welche Spalte als Sortierschlüssel verwendet wird. Die Sortierreihenfolge (entweder aufsteigend oder absteigend) wird durch einen Zweizustandsknopf, der einen nach oben Pfeil oder nach unten Pfeil darstellt, festgelegt.

Sie können mehrere Kartenelemente mit der Atlaserzeugung verwenden; jede Karte wird gemäß den Abdeckungsobjekten dargestellt. Um die Atlaserstellung für ein bestimmtes Kartenelement einzustellen müssen Sie  *Gesteuert durch Atlas* in den Elementeigenschaften des Kartenelements aktivieren. Nach dem Aktivieren können Sie einstellen:

- Ein Eingabefenster *Rand um Objekte* das es Ihnen ermöglicht die Größe des Abstandes, der um jede Geometrie innerhalb der zugeordneten Karte hinzugefügt wird, auszuwählen. Sein Wert ist nur von Bedeutung wenn der auto-skalierungs Modus verwendet wird.
- Ein  *Fester Maßstab*, der es Ihnen ermöglicht zwischen Auto-Skalierung und festem Maßstab umzuschalten. Beim festen Maßstab wird die Karte nur für jede zu zentrierende Geometrie übersetzt. Bei der Auto-Skalierung werden die Kartenausschnitte so berechnet, dass jede Geometrie im Ganzen erscheint.

### 19.6.1 Beschriftungen


Um die Beschriftungen an das Objekt, über das das Plugi iteriert, anzupassen können Sie Ausdrücke einfügen. Zum Beispiel können Sie für einen city Layer mit den Feldern CITY\_NAME und ZIPCODE folgendes einfügen:

```
The area of [% upper(CITY_NAME) || ', ' || ZIPCODE || ' is ' format_number($area/1000000,2) %] km2
```

Die Information [% upper(CITY\_NAME) || ', ' || ZIPCODE || ' is ' format\_number(\$area/1000000,2) %] ist ein Ausdruck, der innerhalb der Beschriftung verwendet wird. Dies würde in einem erzeugten Atlas wie folgt resultieren:

*The area of PARIS,75001 is 1.94 km2*


### 19.6.2 Datendefinierte Übersteuerung Knöpfe

Es gibt mehrere Plätze wo Sie einen  *Datendefinierte Übersteuerung* Knopf zum Überschreiben von ausgewählten Einstellungen verwenden können. Diese Optionen sind besonders nützlich bei der Atlas-Erzeugung.


Für die folgenden Beispiele wird der *Regions* layer des QGIS Beispieldatensatzes verwendet und für die Atlas-Erzeugung ausgewählt. Wir nehmen ausserdem an, dass das Papierformat *A4(210x297)* im *:guilabel:Zusammenstellung* Reiter für das Feld *Voreinstellungen* ausgewählt ist.

Mit einem *Datendefinierte Übersteuerung* Knopf können Sie die Seitenorientierung dynamisch einstellen. Wenn die Höhe (Nord-Süd) des Ausmaßes einer region größer als seine Breite (Ost-West) ist wollen sie wohl lieber *Hochformat* als *Querformat* wählen um den Nutzen des Papiers zu optimieren.

In *Zusammenstellung* können Sie das Feld *Orientierung* einstellen und *Querformat* oder *Hochformat* auswählen. Wir wollen die Orientierung dynamisch anhand eines Ausdrucks, der von der Region geometry abhängt, einstellen.

Drücken Sie den  Knopf des Feldes *Orientierung*, wählen Sie *Bearbeiten ...* so dass der *Ausdruckseditor* Dialog sich öffnet. Geben Sie den folgenden Ausdruck ein:

```
CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 'Landscape' ELSE 'Portrait'
```


Jetzt orientiert sich das Papier automatisch für jede Region für die Sie die Position des Zusammenstellungsitems ebenfalls neu bestimmen müssen. Für das Kartenelement können Sie den  Knopf des Feldes *Breite* verwenden um sie dynamisch einzustellen indem Sie folgenden Ausdruck verwenden:



```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 297 ELSE 210 END) -
```

Verwenden Sie den  Knopf des *Höhe* Feldes um folgenden Ausdruck zu bereitzustellen:

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 210 ELSE 297 END) -
```

Wenn Sie eine Beschriftung über der Karte in der Mitte der Seite vergeben wollen fügen Sie ein Beschriftungselement über der Karte ein. Verwenden Sie erst die Elementeigenschaften des Beschriftungselementes um die horizontale Ausrichtung auf  *Zentriert* einzustellen. Aktivieren Sie als nächstes unter *Referenzpunkt* das obere mittlere Kontrollkästchen. Sie können den folgenden Ausdruck für das Feld *X* eingeben:

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 297 ELSE 210 END) /
```

Für alle anderen Zusammenstellungselemente können Sie die Position auf eine ähnliche Art und Weise einstellen so dass Sie korrekt positioniert sind wenn die Seite automatisch nach Hochformat oder Querformat gedreht wird.

Die zur Verfügung gestellten Informationen sind von dem hervorragenden Blog (in Englisch und Portugiesisch) über die Datendefinierte Übersteuerungsoptionen [Multiple\\_format\\_map\\_series\\_using\\_QGIS\\_2.6](#) abgeleitet.

Dieses ist nur ein Beispiel wie Sie Datendefinierte Übersteuerungen anwenden können.

### 19.6.3 Vorschau

Wenn Sie die Atlaseinstellungen konfiguriert haben und die Kartenelemente ausgewählt, können Sie eine Vorschau von allen Seiten machen indem Sie auf *Atlas → Atlasvorschau anzeigen* klicken und die Pfeile im gleichen Menü zum Navigieren durch alle Objekte verwenden.





### 19.6.4 Erstellung

Die Atlaserstellung kann auf verschiedene Arten durchgeführt werden. Zum Beispiel können sie ihn mit *Atlas → Atlas drucken ...* direkt drucken. Sie können auch ein PDF erstellen indem Sie *Atlas → Atlas als PDF exportieren* verwenden: Der Benutzer wird nach einem Verzeichnis gefragt, in das alle erstellten PDF-Dateien gespeichert werden sollen (außer Sie haben  *Einzeldatei, wenn möglich* ausgewählt). Wenn Sie nur eine Seite des Atlas drucken müssen, starten Sie einfach die Vorschaufunktion, wählen Sie die Seite, die Sie brauchen, aus und klicken Sie *Zusammenstellung → Drucken...* (oder PDF erstellen).

## 19.7 Eine Ausgabe erzeugen

[Figure\\_composer\\_output](#) zeigt die Druckzusammenstellung mit einem Beispieldrucklayout, das jeden Typ von Kartenelement wie in den vorangegangenen Abschnitten beschrieben enthält.

Die Druckzusammenstellung ermöglicht es Ihnen mehrere Ausgabeformate zu erstellen, und es ist möglich die Auflösung (Druckqualität) und Seitengröße zu definieren:

- Das  *Drucken* Icon ermöglicht es Ihnen das Layout in einen angeschlossenen Drucker oder in eine PostScript Datei zu drucken, je nach installierten Druckertreibern.
- Das  *Speichern als Rasterbild* Icon exportiert die Zusammenstellungsansicht in diverse Bildformate wie PNG, BPM, TIF, JPG, ...
- Das  *Als PDF exportieren* speichert die definierte Druckzusammenstellungsansicht direkt als PDF.
- Das  *Speichern als SVG* Icon speichert die Druckzusammenstellungsansicht als SVG (Scalable Vector Graphic).



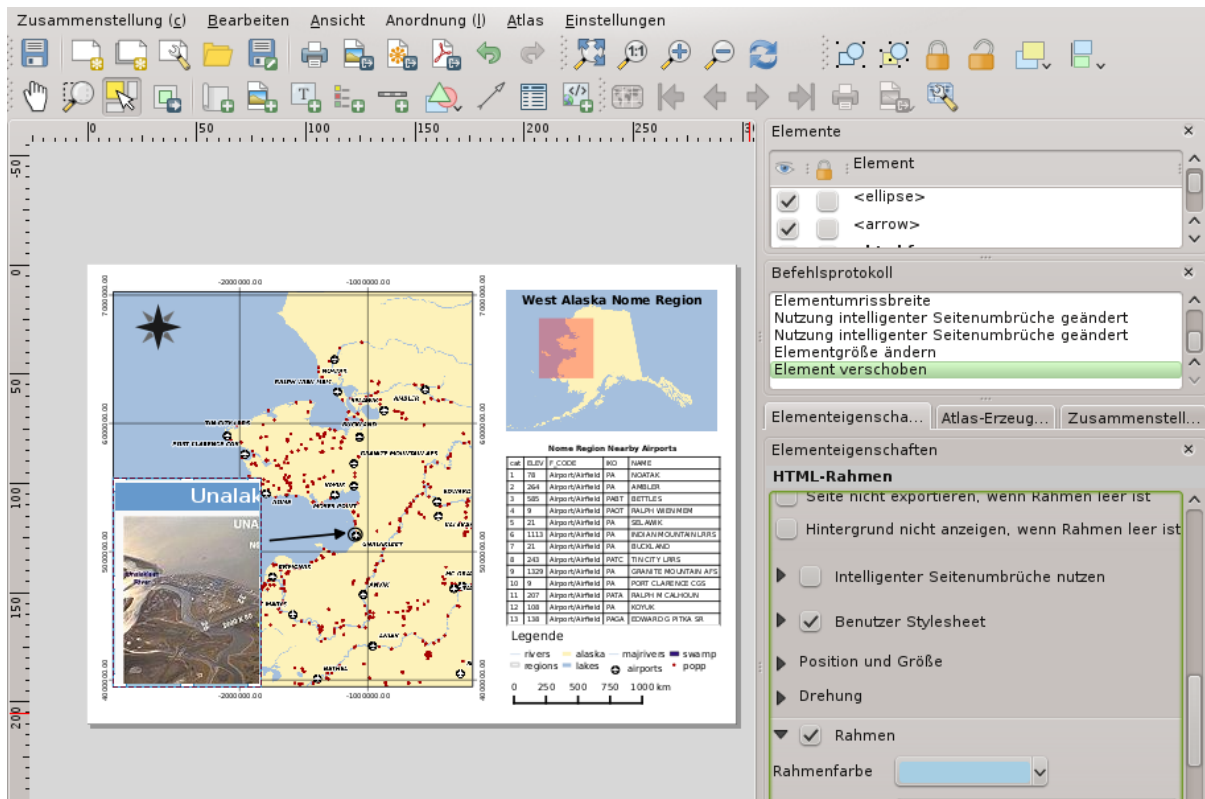


Abbildung 19.41: Druckzusammenstellung mit Kartenansicht, Legende, Bild, Maßstab, Koordinaten, Text und hinzugefügtem HTML-Rahmen

Wenn Sie Ihre Layouts als **georeferenziertes Bild** exportieren müssen (z.B. um es zurück in QGIS zu laden), müssen Sie dies im Zusammenstellung Reiter ermöglichen. Aktivieren Sie  *World-Datei ein* und wählen Sie das zu benutzende Kartenelement. Mit dieser Option erstellt die 'Als Bild exportieren' Aktion ebenfalls eine World-Datei.

**Bemerkung:**

- Zur Zeit ist die SVG Ausgabe sehr einfach gehalten. Dies ist kein QGIS Problem sondern ein Problem mit der darunterliegenden Qt Bibliothek. Dies wird hoffentlich in zukünftigen Versionen behoben.
- Das Exportieren von großen Rastern kann manchmal fehlschlagen, auch wenn es genug Speicher zu geben scheint. Dies ist ebenfalls ein Problem mit dem zugrundeliegenden Qt Management von Rastern.

## 19.8 Die Druckzusammenstellung verwalten

Mit den Icons Als Vorlage speichern und Elemente aus Vorlage hinzufügen können Sie den aktuellen Stand einer Druckzusammenstellungssitzung als ein Template mit der Dateiendung \*.qpt speichern und entsprechend in einer anderen Sitzung wieder laden.

Der Druckzusammenstellung verwalten Knopf in der QGIS Werkzeugleiste und unter *Zusammenstellung* → *Druckzusammenstellung verwalten* ermöglicht es Ihnen eine neue Zusammenstellungsvorlage hinzuzufügen, eine neue Zusammenstellung auf Basis einer vorher abgespeicherten Vorlage zu erstellen oder schon bestehende Vorlagen zu verwalten.

Standardmäßig sucht der Zusammenstellungsmanager nach Vorlagen in `~/qgis2/composer_template`.

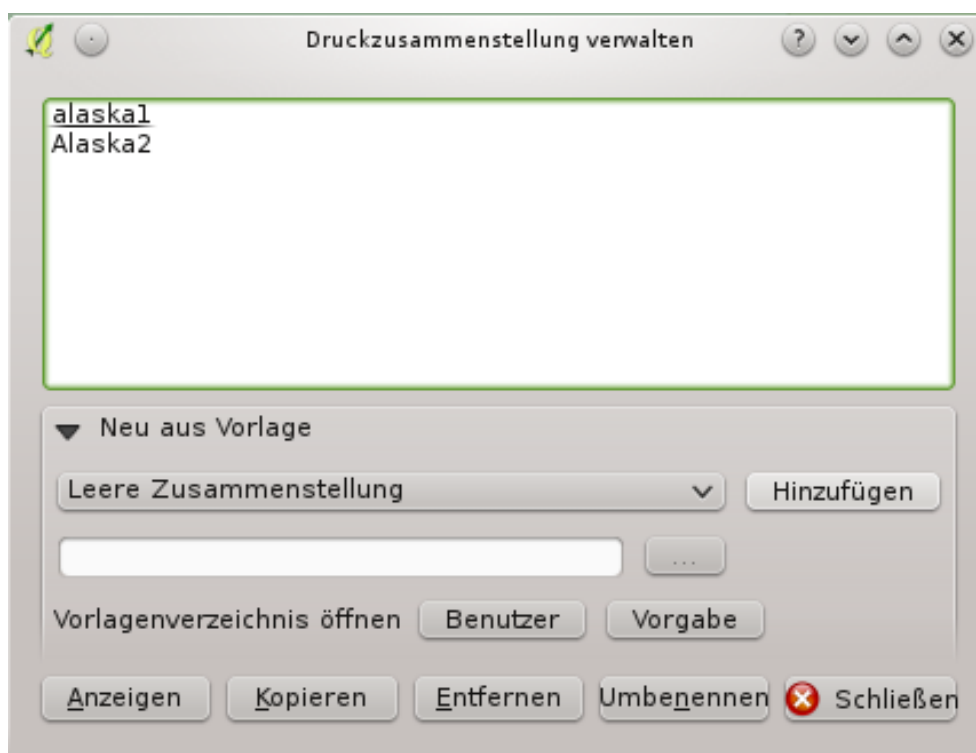






Abbildung 19.42: Die Druckzusammenstellung verwalten 

Die  *Neue Druckzusammenstellung* und  *Zusammenstellungsduplizierung* Knöpfe in der Werkzeugleiste und in *Zusammenstellung* → *Neue Zusammenstellung* sowie in *Zusammenstellung* → *Zusammenstellung kopieren* ermöglicht es Ihnen einen neuen Zusammenstellungsdialog zu öffnen oder eine bestehende Zusammenstellung von einer vorher erstellten zu duplizieren.

Schließlich können Sie Ihre Druckzusammenstellung mit dem  *Projekt speichern* Knopf speichern. Dies ist die gleiche Funktion wie im QGIS Hauptfenster. Alle Änderungen werden in einer QGIS Projektdatei gespeichert.

---

## Erweiterungen

---

### 20.1 QGIS Erweiterungen

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

Sie können Ihre Erweiterungen im Erweiterungen Dialog, der mit *Erweiterungen > Erweiterungen verwalten und installieren ...* geöffnet werden kann, verwalten.

Wenn eine Erweiterung geupdatet werden muss und die Erweiterungseinstellungen dementsprechend vorgenommen wurden kann das QGIS Hauptfenster einen blauen Link in der Statusleiste anzeigen um Ihnen mitzuteilen, dass Erweiterungsupdates angewendet werden können.

#### 20.1.1 Der Erweiterungen Dialog

Die Menüs im Erweiterungen Dialog ermöglichen es dem Anwender Erweiterungen auf verschiedene Arten zu installieren und zu aktualisieren. Für jede Erweiterung werden Metadaten im rechten Bedienfeld angezeigt:

- Informationen falls die Erweiterung experimentell ist
- Beschreibung
- Bewertungen (Sie können eine Bewertung für Ihre bevorzugte Erweiterung abgeben!)
- Elemente
- einige nützliche Links wie die Homepage, die Fehlerverfolgung und das Quellcode-Repository
- Autoren
- verfügbare Version

Sie können den Filter benutzen um eine bestimmte Erweiterung zu finden.



Hier sind alle zur Verfügung stehenden Plugins aufgelistet, darunter sowohl Kern- als auch externe Plugins. Verwenden Sie [**Alle aktualisieren**] um nach neuen Versionen für die Plugins zu schauen. Des Weiteren können Sie [**Erweiterung installieren**] verwenden wenn ein Plugin aufgelistet aber nicht installiert ist und [**Gewählte Erweiterung deinstallieren**] genauso wie [**Erweiterung erneut installieren**] wenn ein Plugin installiert ist. Wenn ein Plugin installiert ist kann es anhand des Kontrollkästchens de/aktiviert werden.



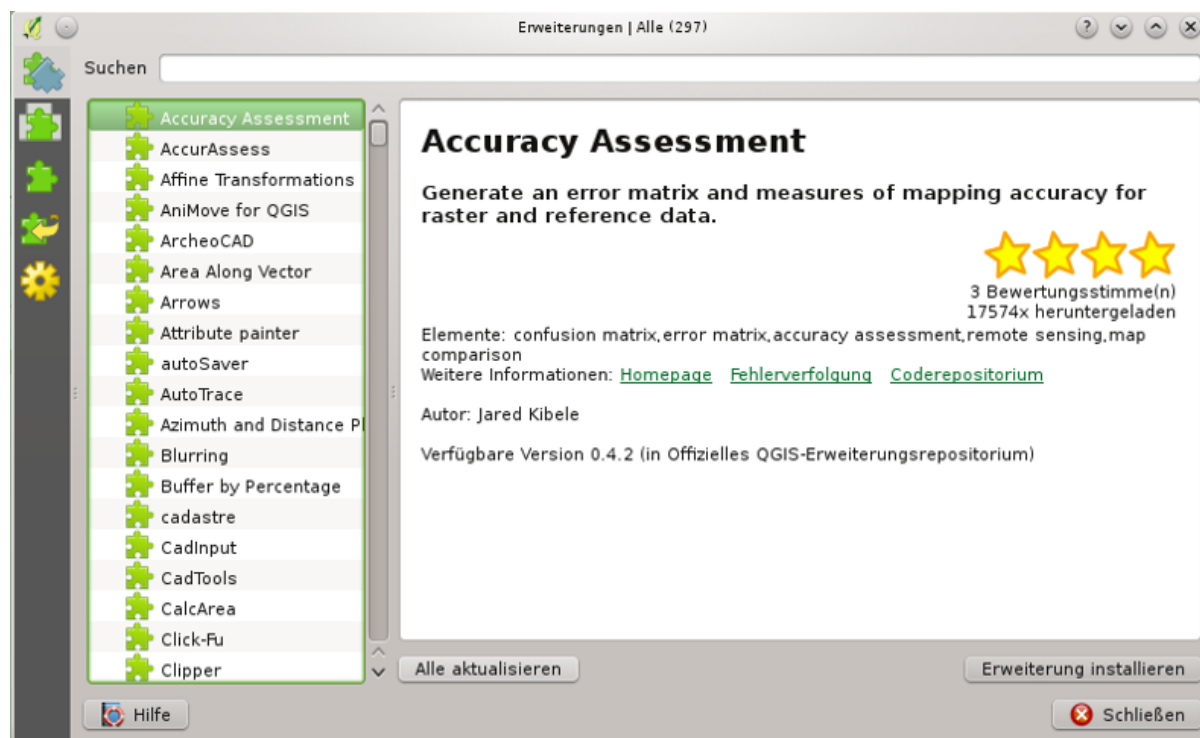



Abbildung 20.1: Das  Alle Menü 

In diesem Menü können Sie nur die installierten Plugins finden. Die externen Plugins können deinstalliert und erneut installiert werden mit den **[Gewählte Erweiterung deinstallieren]** und **[Erweiterung erneut installieren]**. Sie können hier auch **[Alle aktualisieren]**.

#### Nicht installiert

Dieses Menü listet alle zur Verfügung stehenden Plugins, die nicht installiert sind, auf. Sie können den **[Erweiterung installieren]** Knopf um eine Erweiterung in QGIS zu implementieren.

#### Erweiterbar

Wenn Sie  *Auch experimentelle Erweiterungen anzeigen* im  *Einstellungen* Menü aktivieren können Sie dieses Menü zum Suchen nach weiteren aktuellen Pluginversionen verwenden. Dies kann mit den **[Erweiterung erneut installieren]** oder **[Alle aktualisieren]** Knöpfen getan werden.

#### Einstellungen

In diesem Menü können Sie die folgenden Optionen verwenden:

- *Beim Start nach Aktualisierungen suchen.* Wann immer ein neues Plugin oder ein Pluginupdate zur Verfügung steht wird QGIS Sie ‘bei jedem QGIS-Start’, ‘einmal am Tag’, ‘alle drei Tage’, ‘jede Woche’, ‘alle zwei Wochen’ oder ‘jeden Monat’ informieren.
- *Auch experimentelle Erweiterungen zeigen.* QGIS wird Ihnen Plugins in frühen Entwicklungsphasen zeigen, die im Allgemeinen für den Produktiveinsatz ungeeignet sind.
- *Auch veraltete Erweiterungen anzeigen.* Diese Erweiterungen sind veraltet und im Allgemeinen für den Produktiveinsatz ungeeignet.

Um ein externes Anwender-Repository hinzuzufügen, klicken Sie **[Hinzufügen...]** im *Erweiterungsrepositorien* Abschnitt. Wenn Sie eins oder mehrere hinzugefügte Repositories nicht haben wollen können diese über den **[Bearbeiten...]** Knopf deaktiviert werden oder vollständig gelöscht mit dem **[Löschen]** Knopf werden.

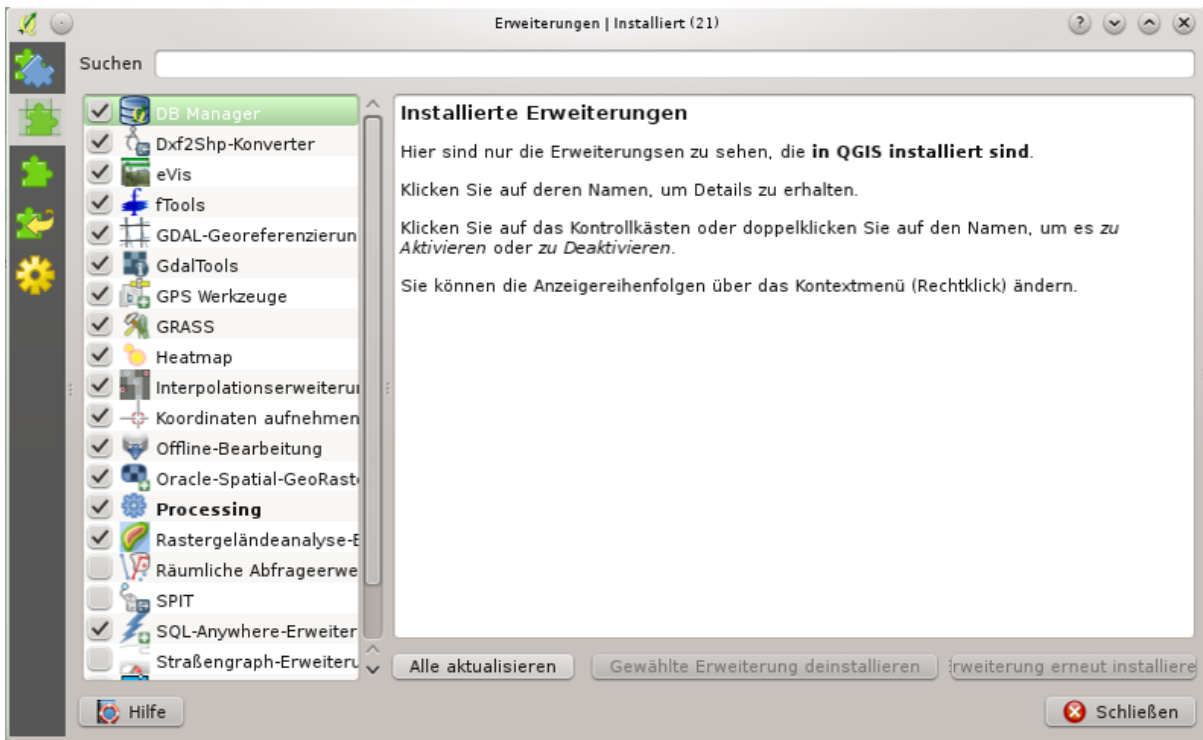


Abbildung 20.2: Das  *Installiert* Menü 

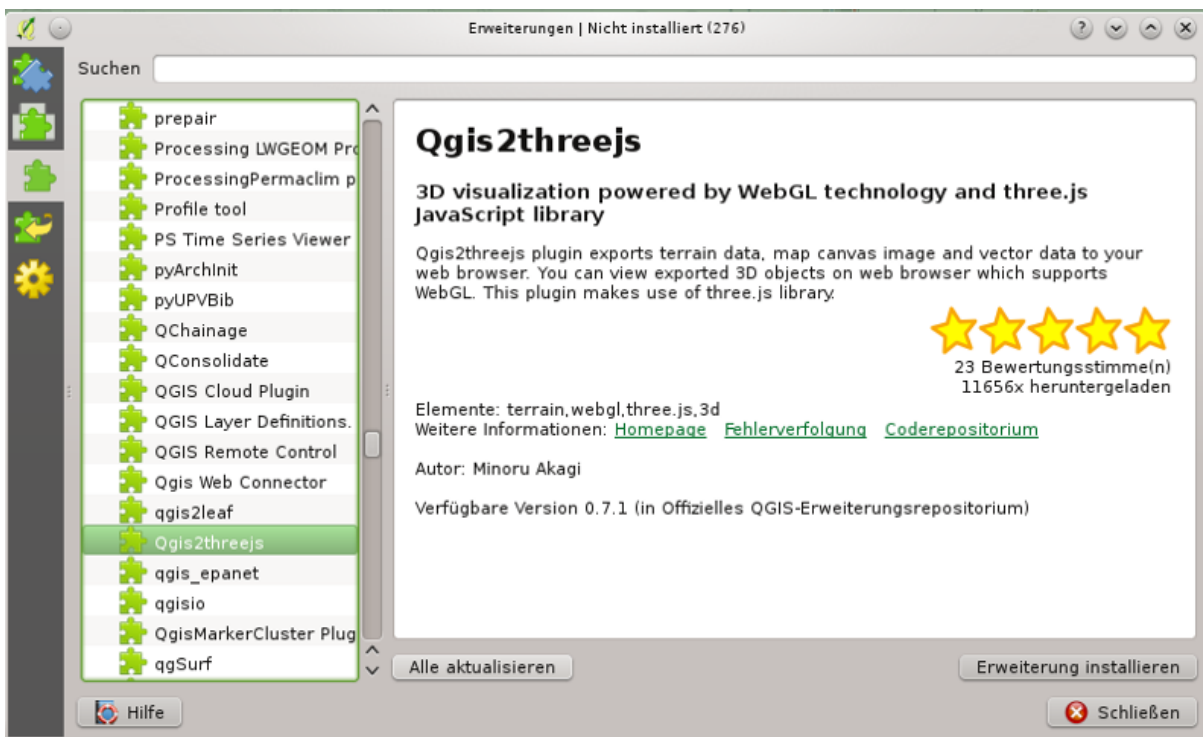


Abbildung 20.3: Das  *Nicht installiert* Menü 

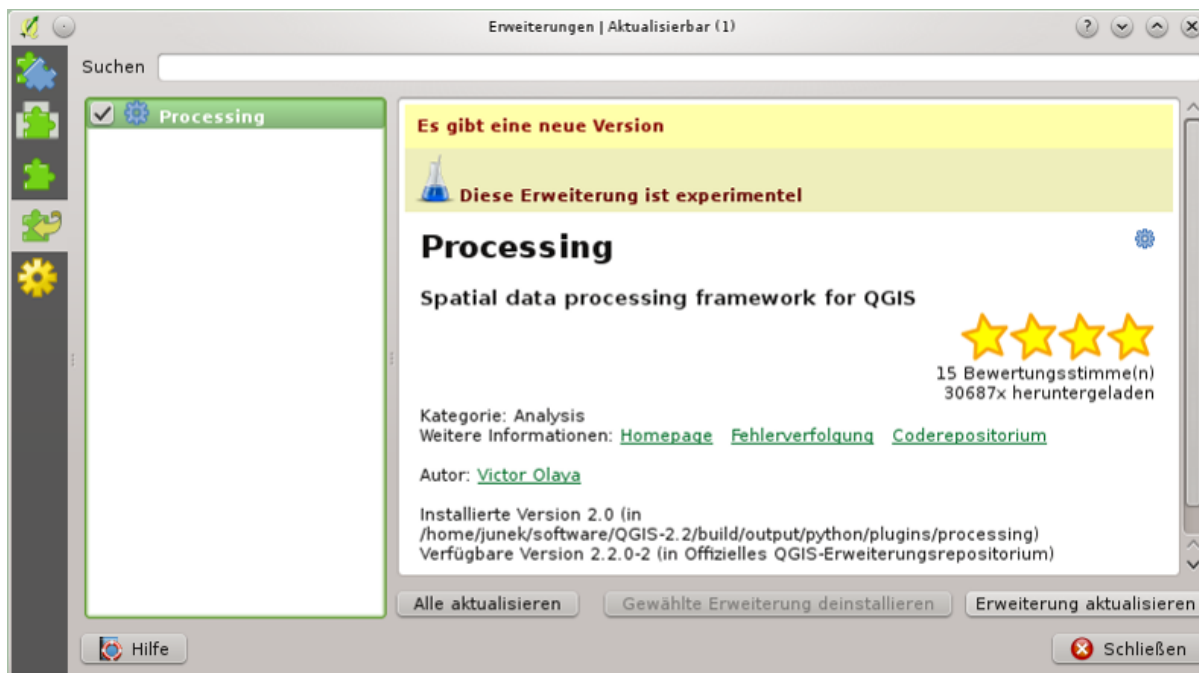



Abbildung 20.4: Das  Erweiterbar Menü 



Abbildung 20.5: Das  Einstellungen Menü 

Die *Suchen* Funktion steht in fast jedem Menü zur Verfügung (mit Ausnahme von  *Einstellungen*). Hier können Sie nach bestimmten Erweiterungen suchen.

---

**Tipp: Kernerweiterungen und externe Erweiterungen**

QGIS Plugins sind entweder als **Kern-Plugins** oder **Externe Plugins** implementiert. **Kern-Plugins** werden durch das QGIS Development Team gewartet und sind automatisch Teil jeder QGIS Distribution. Sie sind entweder in C++ oder in Python geschrieben. **Externe Plugins** sind derzeit alle in Python geschrieben. Sie sind in externen Repositories gespeichert und werden von den individuellen Autoren gewartet.

---

Eine ausführliche Dokumentation über die Verwendung, notwendige QGIS Version, Homepage, Autoren, und weitere Informationen werden für das 'Official' QGIS Repository unter der URL <http://plugins.qgis.org/plugins/> bereitgestellt. Für andere externe Repositories können Dokumentationen mit dem externen Plugin selber zur Verfügung stehen. Sie sind in diesem Handbuch im Allgemeinen nicht enthalten.

## 20.2 QGIS Kernplugins verwenden

Icon	Plugin	Beschreibung	Handbuch Referenz
	Koordinaten aufnehmen	Koordinaten in anderem KBS verfolgen	<i>Koordinaten aufnehmen Plugin</i>
	DB Manager	Datenbanken in QGIS verwalten	<i>DB Manager Plugin</i>
	DXF2Shape Konverter	Wandelt vom DXF- ins Shapeformat um	<i>Dxf2Shape Konverter Plugin</i>
	eVis	Ein Ereignisvisualisierungswerkzeug	<i>eVis Plugin</i>
	fTools	Werkzeuge für Vektoranalyse und Management	<i>fTools Plugin</i>
	GPS Werkzeuge	Werkzeuge zum Laden und Importieren von GPS-Daten	<i>GPS Plugin</i>
	GRASS	Einbinden von GRASS Daten und Modulen	<i>GRASS GIS Integration</i>
	GDALTools	Integration der GDAL Tools in QGIS	<i>GDALTools Plugin</i>
	GDAL Georeferenzierung	Rasterdateien mit GDAL georeferenzieren	<i>Georeferenzier Plugin</i>
	Heatmap	Erzeugt ein Heatmap-Raster für einen Eingabepunktlayer.	<i>Heatmap-Erweiterung</i>
	Interpolationserweiterung	Stützpunktinterpolation von Vektorlayern	<i>Interpolationsplugin</i>
	Offline Editing	Offline-Bearbeitung und Datensynchronisation	<i>Offline-Bearbeitung Plugin</i>
	Oracle-Spatial-Georaster	Auf OracleSpatial-GeoRaster zugreifen	<i>Oracle-Spatial-GeoRaster Plugin</i>
	Erweiterungsmanager	Kernerweiterungen und externe Erweiterungen verwalten	<i>Der Erweiterungen Dialog</i>
	Rastergeländeanalyse	Berechnung geomorphologischer Parameter auf Basis eines DGM	<i>Rastergeländeanalyse-Erweiterung</i>
	Road Graph Erweiterung	Lösen des Kürzeste Wege Problems	<i>Straßengraph Plugin</i>
	SQL-Anywhere	Speichert Vektorlayer in einer SQL-Anywhere Datenbank	<i>SQL-Anywhere Plugin</i>
	Räumliche Abfrage	Räumliche Abfrage von Vektorlayern	<i>Räumliche Abfrage Plugin</i>
	SPIT	Shapes in PostgreSQL/PostGIS importieren Werkzeug	<i>SPIT Plugin</i>
	Räumliche Statistik	Berechnung von Rasterstatik für Vektorflächen	<i>Zonenstatistikerweiterung</i>
	MetaSearch	Mit Metadata Catalogue Services (CSW) interagieren	<i>MetaSearch Katalog Client</i>

## 20.3 Koordinaten aufnehmen Plugin

Das Plugin Koordinaten aufnehmen ist einfach zu bedienen und erlaubt es, Koordinaten für zwei ausgewählte Koordinatenbezugssysteme (KBS) im Kartenfenster abzufragen.



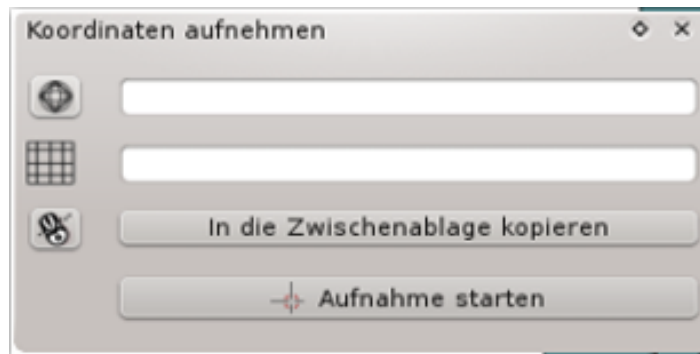








Abbildung 20.6: Koordinaten aufnehmen Erweiterung 

1. Starten Sie QGIS, wählen Sie  *Projekteigenschaften* aus dem *Einstellungen* Menü (KDE, Windows) oder aus dem *Datei* (Gnome, OSX) Menü und klicken Sie auf den *KBS* Reiter. Als Alternative können Sie auch auf das  *KBS-Status* Icon in der rechten unteren Ecke des Statusleiste klicken.
2. Aktivieren Sie die Checkbox  *On-the-Fly-KBS-Transformation aktivieren* und wählen Sie ein KBS Ihrer Wahl (siehe auch *Arbeiten mit Projektionen*).
3. Aktivieren Sie die Koordinaten aufnehmen Erweiterung im Erweiterungsmanager (siehe *Der Erweiterungen Dialog*) und stellen Sie sicher, dass der Dialog sichtbar ist indem Sie zu *Einstellungen* → *Bedienfelder* gehen und sich vergewissern dass  *Koordinaten aufnehmen* aktiviert ist. Der Koordinaten aufnehmen Dialog erscheint wie in Abbildung [figure\\_coordinate\\_capture\\_1](#). Alternativ dazu können Sie auch in das Menü *Vektor* → *Koordinaten aufnehmen* gehen und gucken ob  *Koordinaten aufnehmen* aktiviert ist.
4. Klicken Sie nun auf das Icon  Klicken Sie, um das KBS zur Koordinatenanzeige auszuwählen und wählen Sie ein anderes Koordinatenbezugssystem (KBS) als eben.
5. Klicken Sie nun auf [**Aufnahme starten**] und dann auf einen Punkt im Kartenfenster. Das Plugin zeigt ihnen die Koordinaten für diesen Punkt in beiden zuvor gewählten KBS an.
6. Um die Mausverfolgungs-Funktion zu starten, klicken Sie auf das Icon  *Mausverfolgung*.
7. Sie können die ausgewählten Koordinaten auch in die Zwischenablage kopieren.

## 20.4 DB Manager Plugin

Die DB Manager Erweiterung ist offiziell Teil des QGIS Kerns und soll das SPIT Plugin ersetzen und zusätzlich alle anderen Datenbankformate, die von QGIS unterstützt werden in einer Benutzeroberfläche integrieren. Die

 *DB Manager* Erweiterung bietet mehrere Funktionen. Sie können Layer aus dem QGIS Browser in den DB Manager ziehen und diese werden dann in Ihre räumliche Datenbank importiert. Sie können ein Drag-And-Drop mit Tabellen zwischen räumlichen Datenbanken machen und diese werden dann importiert. Sie können den DB Manager zum Ausführen von SQL Abfragen an Ihre räumliche Datenbank verwenden und dann die räumliche Ausgabe für Abfragen anschauen indem Sie QGIS die Ergebnisse als Abfragelayer hinzufügen.

Das *Datenbank* Menü ermöglicht es Ihnen sich mit einer bestehenden Datenbank zu verbinden, das SQL Fenster zu starten und die DB Manager Erweiterung zu verlassen. Nachdem Sie mit einer bestehenden Datenbank verbunden sind, erscheinen zusätzlich die Menüs *Schema* und *Tabelle*.

Das *Schema* Menü enthält Werkzeuge zum Erstellen und löschen von (leeren) Schemata und, falls es eine Topologie gibt (z.B. PostGIS 2), eins zum Starten von *TopoViewer*.

Das *Tabelle* Menü ermöglicht es Ihnen Tabellen zu erstellen und zu bearbeiten und Tabellen und Views zu löschen. Als weitere Funktionalität können Sie ein VACUUM und dann ein ANALYZE für jede ausgewählte Tabelle durch-

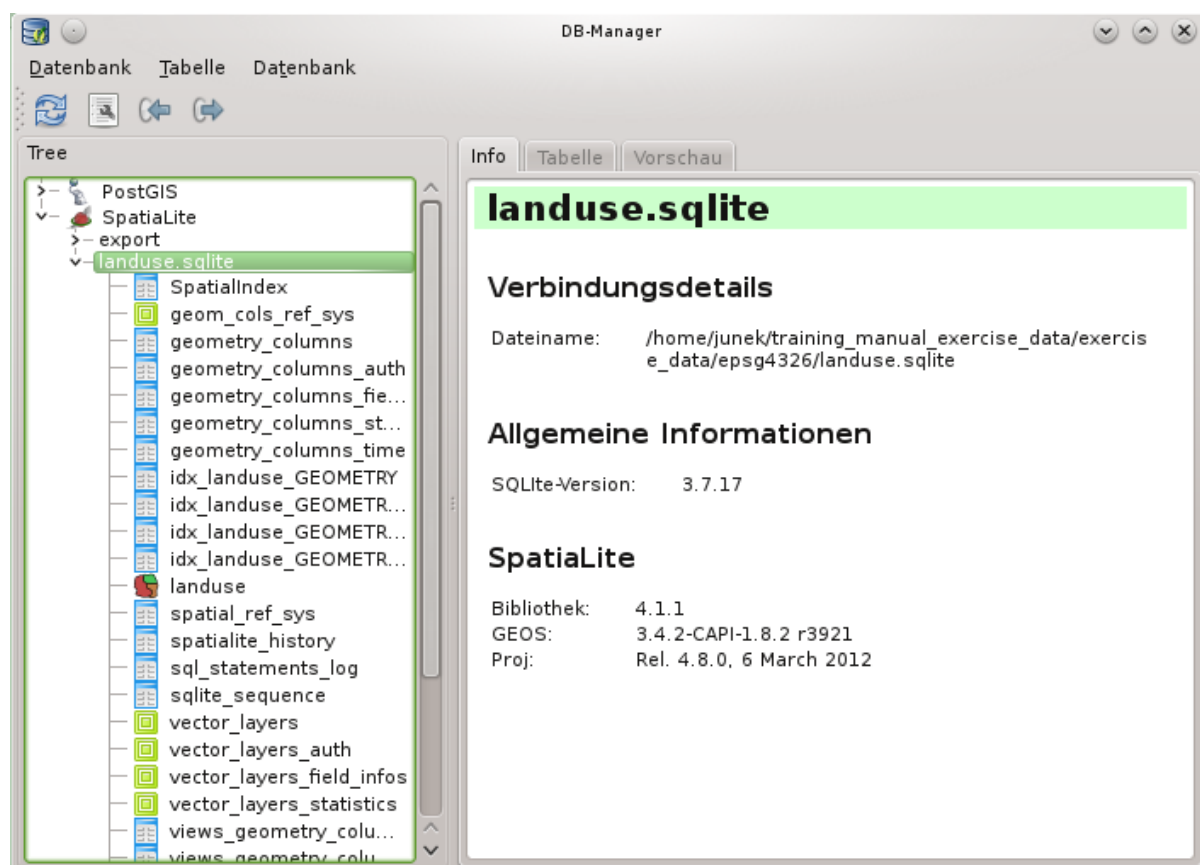


Abbildung 20.7: DB Manager Dialog 

führen. Ein einfaches VACUUM fordert einfach Platz und stellt ihn für das erneute Anwenden zu Verfügung. ANALYZE updatet Statistiken um den effizientesten Weg eine Abfrage durchzuführen zu bestimmen. Schließlich können Sie Layer/Dateien importieren, wenn diese in QGIS geladen sind oder im Dateisystem existieren. Und Sie können Datenbanktabellen nach Shape mit der Export File Funktion exportieren.

Das *Tree* Fenster listet alle existierenden Datenbanken auf, die in QGIS integriert wurden. Mit einem Doppelklick können Sie sich mit der jeweiligen Datenbank verbinden. Mit der rechten Maustaste können Sie eine existierende Tabelle oder Schema umbenennen oder löschen oder eine neue Tabellen hinzugefügen. Tabellen können auch mit dem Kontextmenü zum QGIS Kartenfenster hinzugefügt werden.

Wenn Sie mit einer Datenbank verbunden sind, bietet das **Hauptfenster** des DB Managers drei Reiter. Der *Info* Reiter zeigt Information zur Tabelle und Geometrie, zu existierenden Spalten, Constraints und Indices. Ausserdem kann man die Vacuum Analyze Funktion starten und einen räumlichen Index für eine ausgewählte Tabelle erzeugen. Der *Tabelle* Reiter zeigt die Attribute und der Reiter *Vorschau* zeigt eine Vorschau der Tabelle.

## 20.5 Dxf2Shape Konverter Plugin

Die Dxf2shape-Konverter Erweiterung kann dazu benutzt werden im Vektordaten vom DXF zum Shapedateiformat zu konvertieren. Es erfordert die folgenden Parameter, die vor dem Laufen angegeben werden müssen:



- **Eingabe-DXF-Datei:** Geben Sie den Pfad zur zu konvertierenden DXF-Datei ein.
- **Ausgabedatei:** Geben Sie den gewünschten Namen der Shapedatei, die erstellt werden soll, ein.
- **ausgabedateityp:** Geben Sie den Geometrietyp der Ausgabedatei an. Derzeit unterstützte Typen sind Polylinie, Polygon und Punkt.



Abbildung 20.8: Dxf2Shape-Konverter Erweiterung

- **Beschriftungen exportieren:** Wenn Sie dieses Kontrollkästchen aktivieren, wird ein zusätzlicher Shapefile Punktlayer erstellt, und die damit verknüpfte DBF-Datei enthält die Beschriftungen und Informationen dazu, die sich im 'TEXT'-Feld der Datei befinden.

### 20.5.1 Das Plugin anwenden

1. Starten Sie QGIS, laden Sie das Dxf2Shape Plugin mit dem Plugin Manager (siehe Kapitel *load\_core\_plugin*) und klicken Sie auf das Icon  Dxf2Shape Converter in der Werkzeugleiste. Der DXF-Import Dialog erscheint wie in Abbildung *Figure\_dxf2shape\_1*.
2. Geben Sie die Eingabe-DXF-Datei ein, einen Namen für die Ausgabedatei und den Ausgabedateityp.
3. Aktivieren Sie das Kontrollkästchen  *Beschriftungen exportieren*, wenn Sie einen zusätzlichen Shapefile Punktlayer mit den Beschriftungen erstellen wollen.
4. Klicken Sie [OK].

## 20.6 eVis Plugin

(Dieses Kapitel ist von Horning, N., K. Koy, P. Ersts. 2009. eVis (v1.1.0) User's Guide abgeleitet. American Museum of Natural History, Center for Biodiversity and Conservation. Erhältlich unter <http://biodiversityinformatics.amnh.org/>, and released under the GNU FDL.)

Der Biodiversitäts-Informatik Fachbereich am American Museum of Natural History's (AMNH) Center for Biodiversity and Conservation (CBC) hat das Event Visualization Tool (eVis) entwickelt, ein weiteres Softwaretool das der Reihe der Konservationsmonitoring- und Entscheidungstools für das Managen von Schutzgebieten und Landschaftsplanung hinzugefügt werden kann. Mit diesem Plugin kann der Benutzer auf einfache Weise georeferenzierte (z.B. mit Länge und Breite oder mit X und Y Koordinaten) Fotos und andere unterstützende Dokumente mit Vektordaten in QGIS verlinken.


Das eVis Plugin wird jetzt automatisch in neuen Versionen von QGIS installiert und aktiviert und wie bei allen Plugins kann es mit dem Erweiterungsmanager deaktiviert und aktiviert werden (siehe Abschnitt *Der Erweiterungen Dialog*).

Es besteht aus drei Modulen: der eVis-Datenbankverbindung, dem eVis-Ereignis-ID-Werkzeug und dem eVis-Ereignisbrowser. Diese Werkzeuge arbeiten zusammen, damit das Darstellen von georeferenzierten Fotos und anderen Dokumenten, die mit Vektorobjekten, Datenbanken oder Tabellen verlinkt sind, funktioniert.

### 20.6.1 Ereignisbrowser

Der Ereignisbrowser verfügt über Funktionen, um georeferenzierte Fotos anzuzeigen, die mit Vektorobjekten verknüpft sind, die im Kartenfenster dargestellt sind. Die Punktdaten können dabei z.B. ein in QGIS geladener Vektorlayer oder das Ergebnis einer Datenbankabfrage sein. Das Vektorobjekt muss Attribute mit Informationen zu dem Ort und dem Namen der darzustellenden Datei mit dem Foto enthalten und gegebenenfalls auch die Himmelsrichtung, in welche die Kamera gerichtet war, als das Bild geschossen wurde. Der Vektorlayer muss bereits in QGIS geladen sein, bevor Sie den Ereignisbrowser starten können.

#### Den eVis-Ereignisbrowser starten

Um den Ereignisbrowser zu starten, klicken Sie entweder auf das  eVis Event Browser Icon oder auf *Datenbank* → *eVis* → *eVis Ereignisbrowser*. Dies öffnet den *Ereignisbrowser* Dialog.

Der *Ereignisbrowser-Dialog* besteht aus drei Reitern am oberen linken Rand. Der Reiter *Anzeigen* wird verwendet, um das Foto anzuzeigen und das damit verbundene Datenattribut. Der Reiter *Optionen* enthält eine Reihe von Einstellungen, die angepasst werden, um das Verhalten des Plugins zu verändern. Schließlich gibt es noch den Reiter *Externe Anwendungen konfigurieren*, um eine Tabelle mit Dateierweiterungen und damit verbundenen Anwendungen zu managen, um andere Dokumente als Bilder anzeigen zu können.

#### Der Anzeigen-Reiter

Um das *Anzeigen* Fenster zu sehen, klicken Sie auf den *Anzeigen* Reiter im Dialog *Ereignisbrowser*. Der Reiter *Anzeigen* des Ereignisbrowsers wird verwendet, um georeferenzierte Fotos und damit verknüpfte Attribute anzuzeigen.

1. **Anzeigefenster:** Ein Fenster, in dem das Foto erscheint.
2. **Hineinzoomen:** Zoomen, um Details zu sehen. Wenn das gesamte Bild nicht im Anzeigefenster dargestellt werden kann, erscheinen Scrollbars auf der linken und unteren Seite des Fensters, um das Foto verschieben zu können.
3. **Herauszoomen:** Aus dem Bild herauszoomen, um einen Überblick zu haben.
4. **Zur vollen Ausdehnung zoomen:** Zeigt die volle Ausdehnung des Fotos an.
5. **Attributfenster:** Alle Attribute des Vektorpunktes, der mit dem Foto verlinkt ist werden hier angezeigt. Wenn der Datentyp, auf den verwiesen wird kein Foto ist, sondern ein anderer Datentyp, der im Reiter *Externe Applikationen konfigurieren* definiert ist, dann öffnet sich die Anwendung mit dem speziellen Datentyp, indem Sie auf das Feld mit dem Pfad doppelklicken. Die Anwendung wird dann gestartet und Sie können den Inhalt anschauen oder anhören. Wenn die Datenendung erkannt wird, wird sie automatisch in grün angezeigt.
6. **Navigation:** Verwenden Sie den 'Vorheriges' und 'Nächstes' Knopf, um weitere Objekte zu laden, falls vorhanden und selektiert.

#### Der Optionen-Reiter

1. **Dateipfad:** Eine Dropdown-Liste, um das Attributfeld zu definieren, welches den Verzeichnispfad oder die URL für die Fotos oder anderen Dokumente enthält. Wenn es ein relativer Pfad ist, dann wählen Sie das Kontrollkästchen  *Pfad ist relativ*, links neben dem Dropdown-Menü. Der Grundpfad für den relativen Pfad kann in dem *Grundpfad* eingegeben werden. Informationen über die verschiedenen Optionen zur Angabe des Speicherortes der Datei werden weiter unten im Abschnitt *Den Ort und Namen eines Fotos festlegen* beschrieben.

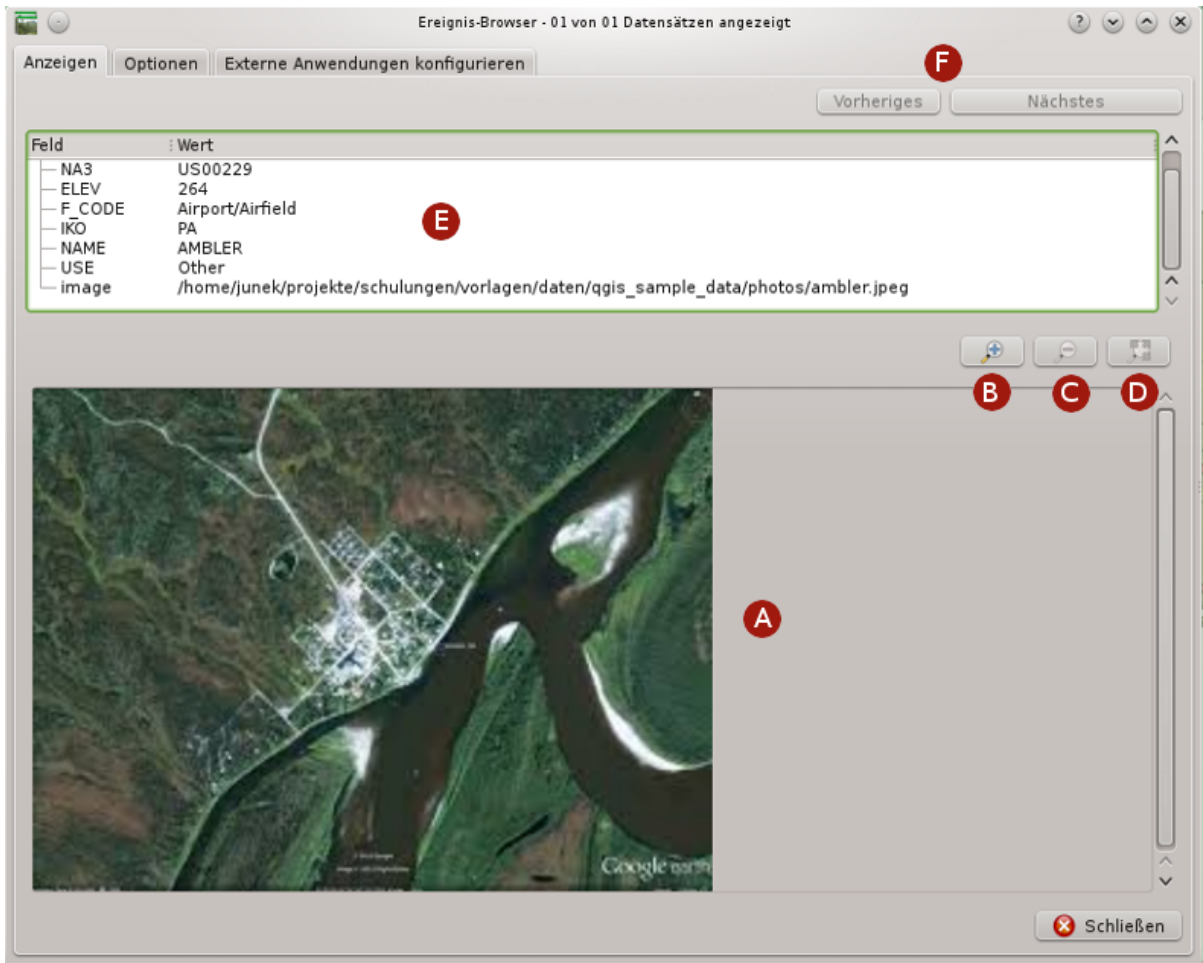


Abbildung 20.9: Das eVis Anzeigen Fenster

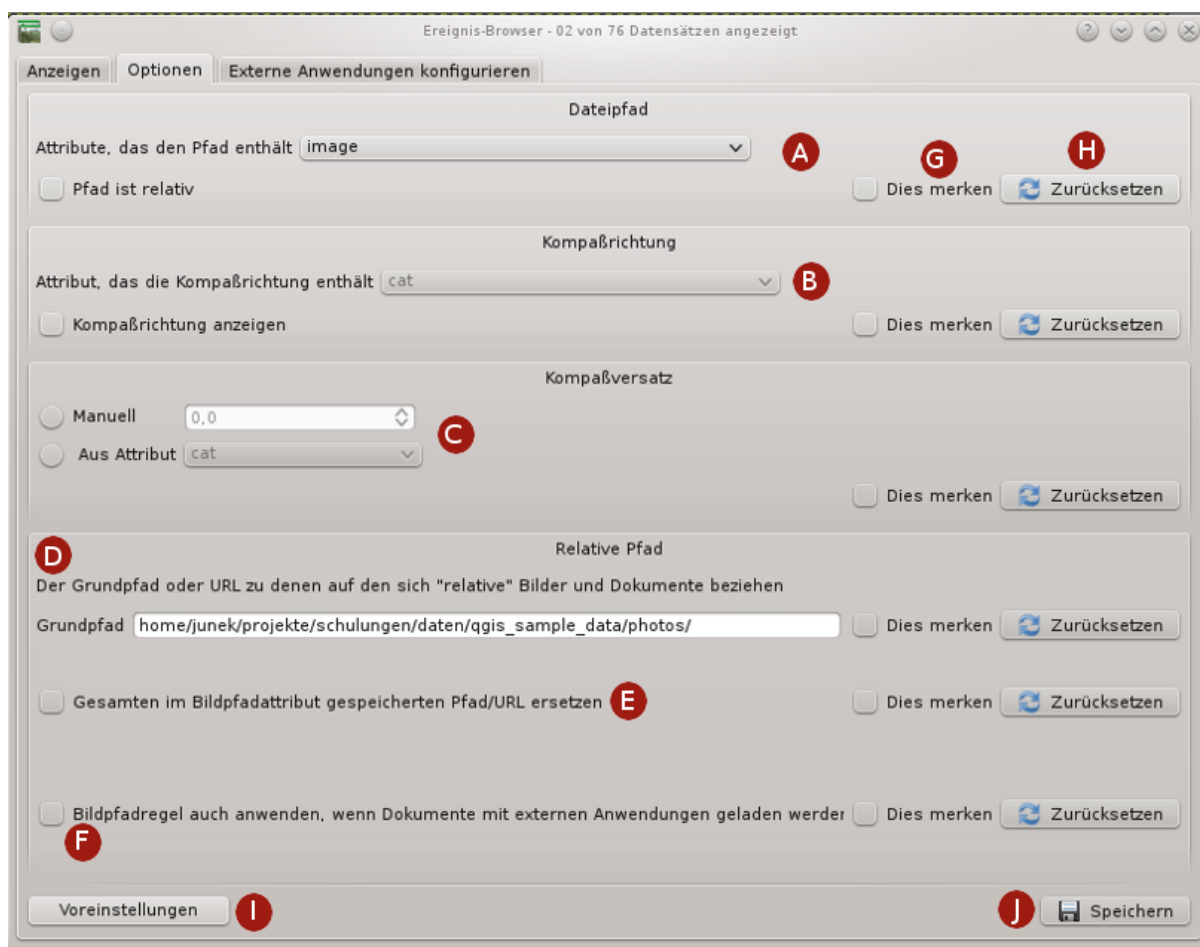


Abbildung 20.10: Das *eVis* Optionen Fenster

2. **Kompassrichtung:** Eine Dropdownliste um das Attributfeld anzugeben, das die Kompassrichtung des angezeigten Fotos enthält. Wenn Kompassversatzinformationen zur Verfügung stehen, ist es erforderlich das Kontrollkästchen unter dem Dropdownmenütitel zu klicken.
3. **Kompassversatz:** Kompassversätze können dazu benutzt werden Deklinationen zu kompensieren (um Versätze, die unter Anwendung von Magnetlagern gesammelt wurden, zu geographisch Nord anzupassen). Klicken Sie den  *Manuell* Radiobutton um den Versatz in das Textfenster einzugeben oder klicken Sie den  *Aus Attribut* Radiobutton um das Attributfeld, das die Versätze enthält, auszuwählen. Für beide dieser Optionen sollten Ostdeklinationen mit positiven Werten und Westdeklinationen mit negativen Werten eingegeben werden.
4. **Relativer Pfad:** Der Grundpfad, auf den sich die relativen Pfade der Fotos und Dokumente beziehen wie in Abbildung [Figure\\_eVis\\_2](#) (A) und wird angehängt.
5. **Gesamten Pfad ersetzen:** Wenn dieses Kontrollkästchen aktiviert ist, wird nur der Dateiname an den Grundpfad angehängt, der als Attributspalte im Bereich Dateipfad definiert wurde.
6. **Bildpfad auf alle Dokumente anwenden:** Wenn dieses Kontrollkästchen aktiviert ist, wird derselbe Grundpfad, der für Fotos verwendet wird, auch auf alle anderen Dokumente wie Filme, Texte, usw. verwendet. Ansonsten werden die definierten Pfadangaben nur für die Fotos verwendet. Alle anderen Dokumente ignorieren den Grundpfad.
7. **Dies merken:** Wenn dieses Kontrollkästchen aktiviert ist, werden alle damit in Verbindung stehenden Parameter für die nächste Sitzung gespeichert. Sie können dazu auch global den Knopf [**Speichern**] drücken.
8. **Zurücksetzen:** Setzt die Werte der jeweiligen Zeile auf die Standardwerte zurück.
9. **Voreinstellungen.** This setzt alle Felder auf ihre voreingestellten Einstellungen. Es hat den gleichen Effekt wie das Klicken auf die [**Zurücksetzen**] Knöpfe.
10. **Speichern:** Dieser Knopf speichert alle Einstellungen, ohne den Dialog zu schließen.

## Der Externe Applikationen konfigurieren Reiter

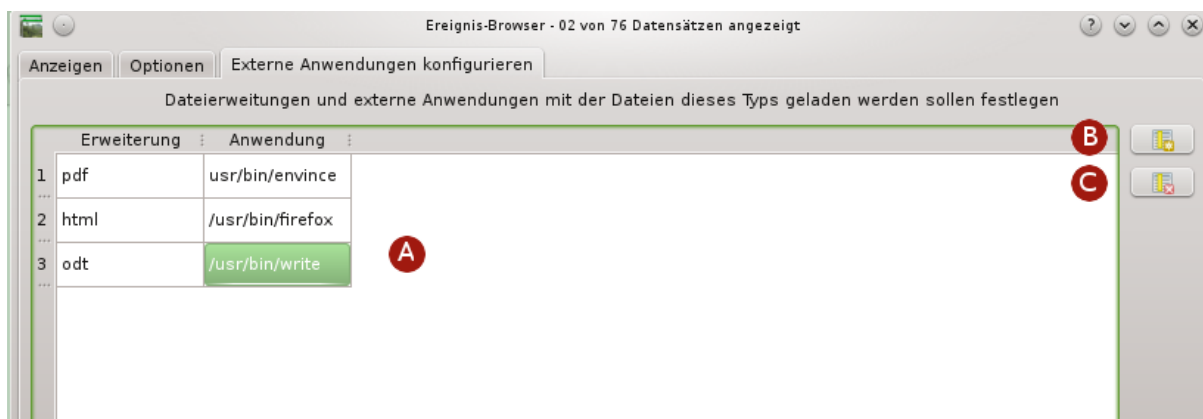


Abbildung 20.11: Das *eVis* Externe Anwendungen konfigurieren Fenster

1. **Datei Referenztable:** Eine Tabelle mit den Datentypen, die mit *eVis* geöffnet werden können. Jeder Datentyp benötigt eine Dateiendung, und den Pfad zu einer Anwendung, die den entsprechenden Datentyp anzeigen oder ausführen kann. Dies schließt neben Fotos u.a. auch Textdokumente, Filme oder Hörspiele mit ein.
2. **Neuen Dateityp hinzufügen:** Füge einen neuen Dateityp mit einer einzigartigen Dateiendung hinzu, sowie einen Pfad zu der Applikation, mit der dieser Datentyp geöffnet werden kann.
3. **Aktuelle Zeile löschen:** Löschen Sie den ausgewählten Dateityp aus der Referenztable.



## 20.6.2 Den Ort und Namen eines Fotos festlegen

Der Ort und Name eines Fotos kann über einen absoluten oder relativen Pfad festgelegt werden. Wenn das Foto auf einem Webserver liegt, kann auch eine URL verwendet werden. Beispiele für die verschiedenen Varianten finden Sie in Tabelle [evis\\_examples](#).

X	Y	FILE	BEARING
780596	1784017	C:\Workshop\eVis_Data\groundphotos\DSC_0168.JPG	275
780596	1784017	/groundphotos/DSC_0169.JPG	80
780819	1784015	http://biodiversityinformatics.amnh.org/\ evis_testdata/DSC_0170.JPG	10
780596	1784017	pdf:http://www.testsite.com/attachments.php?\ attachment_id-12	76

## 20.6.3 Den Ort und Namen anderer Dokumente und Dateien festlegen

Belege wie Text-Dokumente, Videos und Sound-Clips können ebenfalls angezeigt oder abgespielt werden. Dazu ist es notwendig, einen Eintrag in der Referenztabelle zu erstellen, über den die Quelle dann geöffnet werden kann. Außerdem ist es notwendig, den Pfad oder eine URL in die Attributtabelle des Vektorlayers für das entsprechende Objekt einzutragen. Eine weitere Regel, die für URLs verwendet werden kann, die keine Dateierweiterung für das Dokument enthalten, das Sie öffnen möchten, besteht darin, die Dateierweiterung vor der URL anzugeben. Das Format ist dann — `Dateierweiterung:URL`. Vor die URL wird die Dateierweiterung mit einem Doppelpunkt geschrieben. Dies ist besonders nützlich für den Zugriff auf Dokumente in Wikis und anderen Webseiten, die eine Datenbank verwenden, um die Webseiten zu verwalten (siehe Tabelle [evis\\_examples](#)).

## 20.6.4 Den eVis-Ereignisbrowser anwenden

Wenn Sie den *Ereignisbrowser* starten, öffnet sich ein Foto und wird im Display angezeigt, wenn in der Attributtabelle auf das entsprechende Foto verwiesen wird, und wenn der Speicherort der Datei im Reiter *Optionen* richtig eingestellt ist. Wenn Sie ein Foto erwarten, es aber nicht angezeigt wird, kontrollieren Sie die Parameter nochmals im *Optionen* Fenster.

Wenn auf ein Dokument (oder ein Bild, das nicht über eine eVis bekannte Dateierweiterung verfügt) in der Attributtabelle verwiesen wird, wird das Feld mit dem Dateipfad grün hervorgehoben dargestellt, wenn die Dateierweiterung sich in der definierten Referenztabelle im Reiter *Externe Applikationen konfigurieren* befindet. Um das Dokument zu öffnen, doppelklicken Sie auf die grün markierte Zeile im Attribut Informationsfenster. Wenn auf ein Objekt in der Attributtabelle verwiesen wird und der Dateipfad nicht grün markiert ist, dann müssen Sie einen Eintrag für die Dateierweiterung entsprechend ergänzen. Wenn der Dateipfad grün markiert ist, sich aber nicht bei einem Doppelklick öffnet, müssen Sie die Parameter im Reiter *Optionen* einstellen, damit die Datei von eVis gefunden und dargestellt werden kann.

Wenn keine Kompassrichtung vorhanden ist, wird im Reiter *Optionen* ein rotes Sternchen angezeigt, sofern das Vektorobjekt, das mit dem Foto verlinkt ist, angezeigt wird. Wenn eine Kompassrichtung vorhanden ist, erscheint ein Pfeil und weist in die Richtung, die durch die der Wert in der Anzeige Kompassrichtung definiert ist. Der Pfeil wird über dem Vektorpunkt, der mit dem Foto oder einem anderen Dokument verknüpft ist, zentriert dargestellt.


Um den Ereignisbrowser zu schließen, klicken Sie auf den Knopf [**Schließen**] im Dialogfenster.

## 20.6.5 Ereignis-ID-Werkzeug

Das 'Ereignis-ID-Werkzeug' ermöglicht es Ihnen, ein Foto oder Dokument anzuzeigen, indem Sie auf ein Objekt eines aktivierten Vektorlayers im QGIS Kartenfenster klicken. Das Vektorobjekt muss auf Attribute verweisen, welche Informationen zum Pfad und Namen der Datei und gegebenenfalls der Kompassrichtung während der Aufnahme mit einer Kamera enthalten. Der Vektorlayer muss vor der Ausführung des 'Ereignis-ID-Werkzeugs' in QGIS geladen worden sein.



## Starten des Ereignis-ID-Werkzeugs

Um das 'Ereignis-ID' Modul zu öffnen klicken Sie entweder auf das  eVis-Ereignis-ID Icon oder *Datenbank* → *eVis* → *eVis-Ereignis-ID-Werkzeug*. Dies bewirkt, dass der Cursor in einen Pfeil mit einem 'i' am oberen Ende wechselt, was bedeutet, dass das ID Werkzeug aktiv ist.


Um die mit den Vektorobjekten verlinkten Fotos in dem aktiven Vektorlayer anzuzeigen, bewegen Sie den Ereignis-ID Mauszeiger über das Objekt und klicken dann auf die linke Maustaste. Nach einem Klick auf das Objekt, wird der *Ereignisbrowser* geöffnet und die Fotos auf oder in der Nähe des angeklickten Ortes werden angezeigt. Wenn mehr als ein Foto zur Verfügung steht, können Sie durch die verschiedenen Objekte wandeln, indem Sie auf die Schaltflächen [**Vorheriges**] und [**Nächstes**] drücken. Die anderen Steuerelemente sind in Kapitel *Ereignisbrowser* beschrieben.

## 20.6.6 Datenbankverbindung

Der Dialog Datenbankverbindung ermöglicht es, sich mit einer Datenbank oder anderen ODBC Quelle, z.B. Excel-Tabellen zu verbinden.

eVis kann sich direkt mit den folgenden Typen von Datenbanken verbinden: PostgreSQL, MySQL und SQLite; es kann auch ODBC-Verbindungen auslesen (z.B. MS Access). Wenn Sie eine ODBC Datenbank auslesen (wie eine Excel Tabellenblatt) ist es erforderlich Ihren ODBC-Treiber für das Betriebssystem, das Sie verwenden, zu konfigurieren.

### Starten des Dialogs Datenbankverbindung

Um das 'Datenbankverbindung' Modul zu starten klicken Sie auf das entsprechende Icon  eVis Datenbankverbindung oder klicken Sie auf *Datenbank* → *eVis* → *eVis-Datenbankverbindung*. Dies öffnet das *Datenbankverbindung* Fenster. Das Fenster hat drei Reiter: *Vordefinierte Abfragen*, *Datenbankverbindung* und *SQL-Abfrage*. Das *Ausgabekonsolle* Fenster im unteren Teil des Fensters stellt den Status der Aktionen, die von verschiedenen Abschnitten dieses Moduls initiiert wurden, dar.

### Verbinden mit einer Datenbank

Drücken Sie auf den Reiter *Datenbankverbindung* und definieren Sie dann im Dropdown-Menü den *Datenbanktyp*, mit dem Sie sich verbinden wollen. Wenn ein Passwort oder Benutzername erforderlich ist, können diese Informationen in den Feldern *Benutzername* und *Passwort* eingegeben werden.

Geben Sie den Datenbank-Host in das *Datenbank-Host* Textfenster ein. Diese Option steht nicht zur Verfügung wenn Sie 'MS Access' als Datenbanktyp ausgewählt haben. Wenn die Datenbank auf Ihrem Desktop vorgehalten wird sollten Sie "localhost" eingeben.

Geben Sie den Namen der Datenbank in das *Datenbankname* Textfeld ein. Wenn Sie 'ODBC' als Datenbanktyp eingegeben haben, müssen Sie den Datenquellenname eingeben.

Nachdem Sie alle Parameter angegeben haben, klicken Sie auf den [**Verbinden**] Knopf. Wenn die Verbindung erfolgreich ist, wird dies in der *Ausgabekonsolle* angezeigt. Im anderen Fall müssen Sie die Parameter nochmal überprüfen, die Sie eingegeben haben.

1. **Datenbanktyp:** Eine Dropdown-Liste, um den zu verwendenden Datenbanktyp festzulegen.
2. **Datenbank-Host:** Der Name des Datenbank-Hosts.
3. **Port:** Die Portnummer, wenn MYSQL oder PostgreSQL benutzt wird.
4. **Datenbankname:** Der Name der Datenbank.
5. **Verbinden:** Ein Knopf, um sich mit der Datenbank, die die oben definierten Parameter verwendet, zu verbinden.
6. **Ausgabekonsolle:** Hier werden Informationen zu den Prozessen angezeigt.

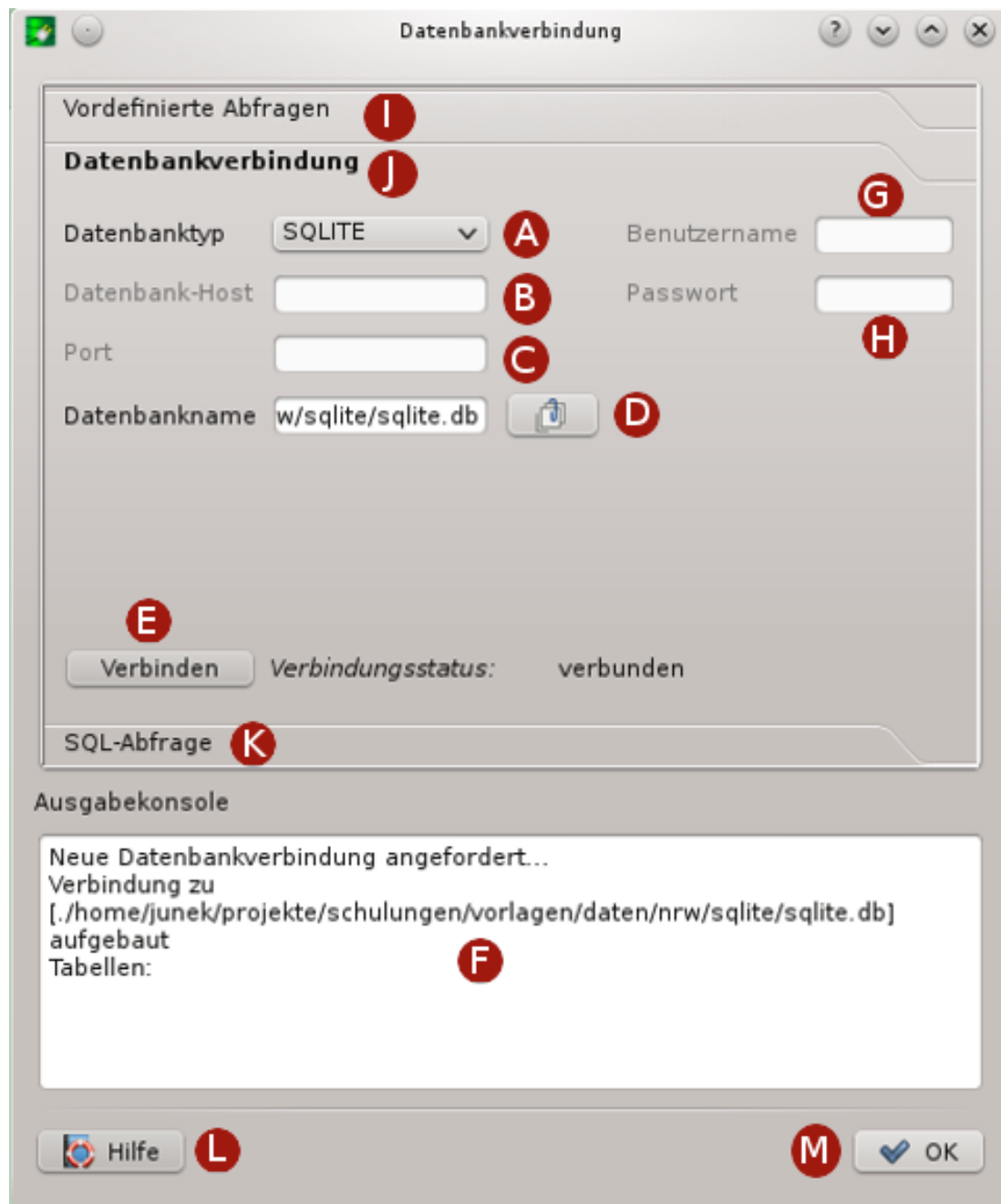


Abbildung 20.12: Das *eVis* Datenbankverbindung Fenster

7. **Benutzername:** Benutzername, um sich mit einer passwortgeschützten Datenbank zu verbinden.
8. **Passwort:** Passwort, wenn die Datenbank passwortgeschützt ist.
9. **Vordefinierte Abfragen:** Reiter, um den Dialog ‘Vordefinierte Abfragen’ zu öffnen.
10. **Datenbankverbindung:** Reiter, um den Dialog ‘Datenbankverbindung’ zu öffnen.
11. **SQL-Abfrage:** Reiter, um den Dialog ‘SQL-Abfrage’ zu öffnen.
12. **Hilfe:** Zeigt die Online-Hilfe an.
13. **OK:** Schließt das Fenster ‘Datenbankverbindung’.



## SQL-Abfrage durchführen

SQL-Abfragen werden verwendet, um Informationen aus einer Datenbank oder ODBC-Ressource zu extrahieren. In eVis wird die Abfrage als Vektorlayer zum QGIS Kartenfenster hinzugefügt. Klicken Sie auf den Reiter *SQL-Abfrage*, um die SQL-Abfrage-Schnittstelle zu öffnen. Sie können dann SQL-Kommandos in dem Textfenster eingeben. Ein hilfreiches Tutorial über SQL-Befehle finden Sie unter der <http://www.w3schools.com/sql>. Um z.B. einen Auszug aller Daten aus einer Excel-Datei zu erstellen, können Sie eingeben `select * from [Blatt1\]$]` wobei ‘Blatt1’ der Name des Arbeitsblattes ist.

Klicken Sie dann auf die Knopf [**Abfrage ausführen**], um den Befehl auszuführen. Wenn die Abfrage erfolgreich ist, wird ein *Auswahlfenster* angezeigt. Wenn die Abfrage nicht erfolgreich ist, erscheint eine Fehlermeldung in der *Ausgabekonsole*.

Im *Datenbank Dateiauswahl* Fenster geben Sie den Namen des Layers an, der aus dem Ergebnis der Abfrage erstellt werden soll.

1. **SQL-Abfrage:** Ein Fenster zur Eingabe von SQL-Abfragen.
2. **Abfrage ausführen:** Knopf, um die SQL-Abfrage auszuführen.
3. **Ausgabekonsole:** Hier werden Informationen zu den Prozessen angezeigt.
4. **Hilfe:** Zeigt die Online-Hilfe an.
5. **OK:** Schließt das Fenster *Datenbankverbindung*.

Verwenden Sie die Dropdown-Menüs *X Koordinate*  und *Y Koordinate* , um die Spalten der Datenbank zu selektieren, in denen die X (oder Longitude) and Y (oder Latitude) Koordinaten abgelegt sind. Wenn Sie nun auf den [OK] Knopf drücken, wird ein Vektorlayer entsprechend der SQL-Abfrage erstellt und als neuer Layer in QGIS angezeigt.

Sie können den Layer als Shapefile speichern, wenn Sie mit dem Mauszeiger auf den Layernamen in der Legende zeigen und dann auf die rechte Maustaste klicken. Dort erscheint dann die Option ‘Speichern als...’

---

### **Tipp: Einen Vektorlayer von einem Microsoft Excel Arbeitsblatt erstellen**

Beim Erstellen eines Vektorlayers aus einem Microsoft Excel-Arbeitsblattes entdecken Sie vielleicht, dass unerwünschte Nullwerte (“0”) gemeinsam mit gültigen Daten in den Tabellenzeilen vorhanden sind. Dies kann verursacht werden, wenn Sie die Felder in Excel mit der Rücktaste Taste gelöscht haben. Um dieses Problem zu beheben, müssen Sie die Excel-Datei öffnen. (Sie müssen dazu QGIS und die Verbindung zur Excel-Tabelle zuerst beenden.) Dann können Sie die Datei in Excel bearbeiten und die leeren Zeilen über das *Bearbeiten* → *Löschen* entfernen und dann wieder abspeichern. Um dieses Problem zu verhindern können Sie einfach mehrere Zeilen in der Exceltabelle löschen indem Sie *Bearbeiten* → *Löschen* vor dem Speichern der Datei verwenden.

---

## Vordefinierte Abfragen starten

Über den Reiter *Vordefinierte Abfragen* können Sie zuvor schriftlich erstellte SQL-Abfragen, die als Datei im XML-Format gespeichert werden, wieder aufrufen. Dies ist besonders hilfreich, wenn Sie nicht vertraut mit SQL-Befehlen sind. Klicken Sie auf den Reiter *Vordefinierte Abfragen*, um die vordefinierte Abfrage-Schnittstelle zu öffnen.

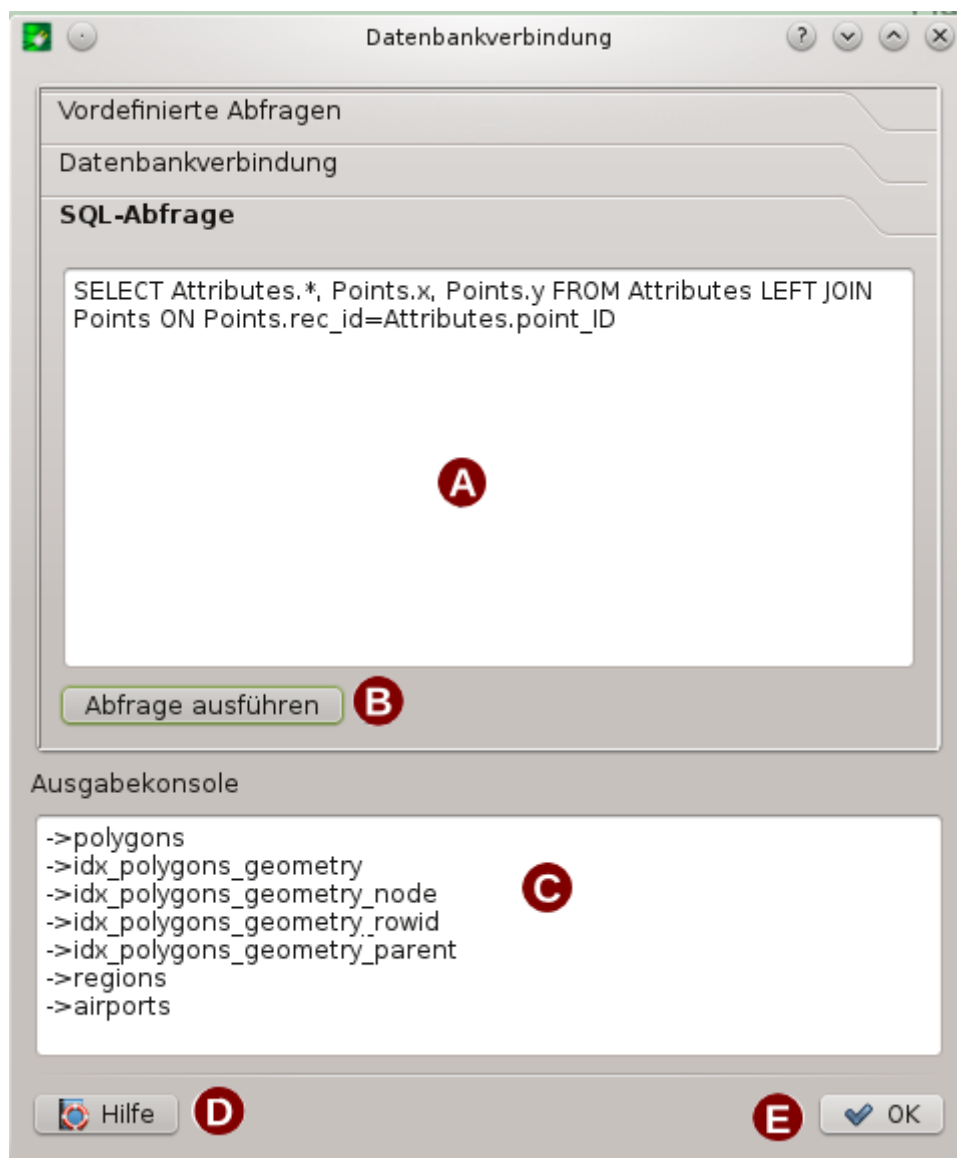




Abbildung 20.13: Der eVis SQL-Abfrage Reiter

Um eine Reihe von vordefinierten Abfragen zu laden, klicken Sie auf das  Datei öffnen Icon. Dies öffnet das *Datei öffnen* Fenster, das dazu verwendet wird die Datei mit den SQL-Abfragen ausfindig zu machen. Wenn die Abfragen geladen sind, erscheinen ihre Titel wie in der XML-Datei definiert im Dropdownmenü gerade unter dem  Datei öffnen Icon. Die ausführliche Beschreibung der Abfrage wird im Textfenster unter dem Dropdownmenü dargestellt.

Wählen Sie die Abfrage, die Sie aus dem Dropdownmenü ausführen wollen und klicken Sie dann auf den *SQL-Abfrage* Reiter um zu sehen, dass die Abfrage in das Abfragefenster geladen worden ist. Wenn Sie das erste Mal eine vordefinierte Abfrage ausführen oder die Datenbank wechseln müssen Sie sich vergewissern, dass Sie mit der Datenbank verbunden sind.

Klicken Sie auf den Knopf [**Abfrage ausführen**] im Reiter *SQL-Abfrage*. Wenn die Abfrage erfolgreich ist, wird ein *Auswahlfenster* angezeigt. Wenn die Abfrage nicht erfolgreich ist, erscheint eine Fehlermeldung in der *Ausgabekonsole*.

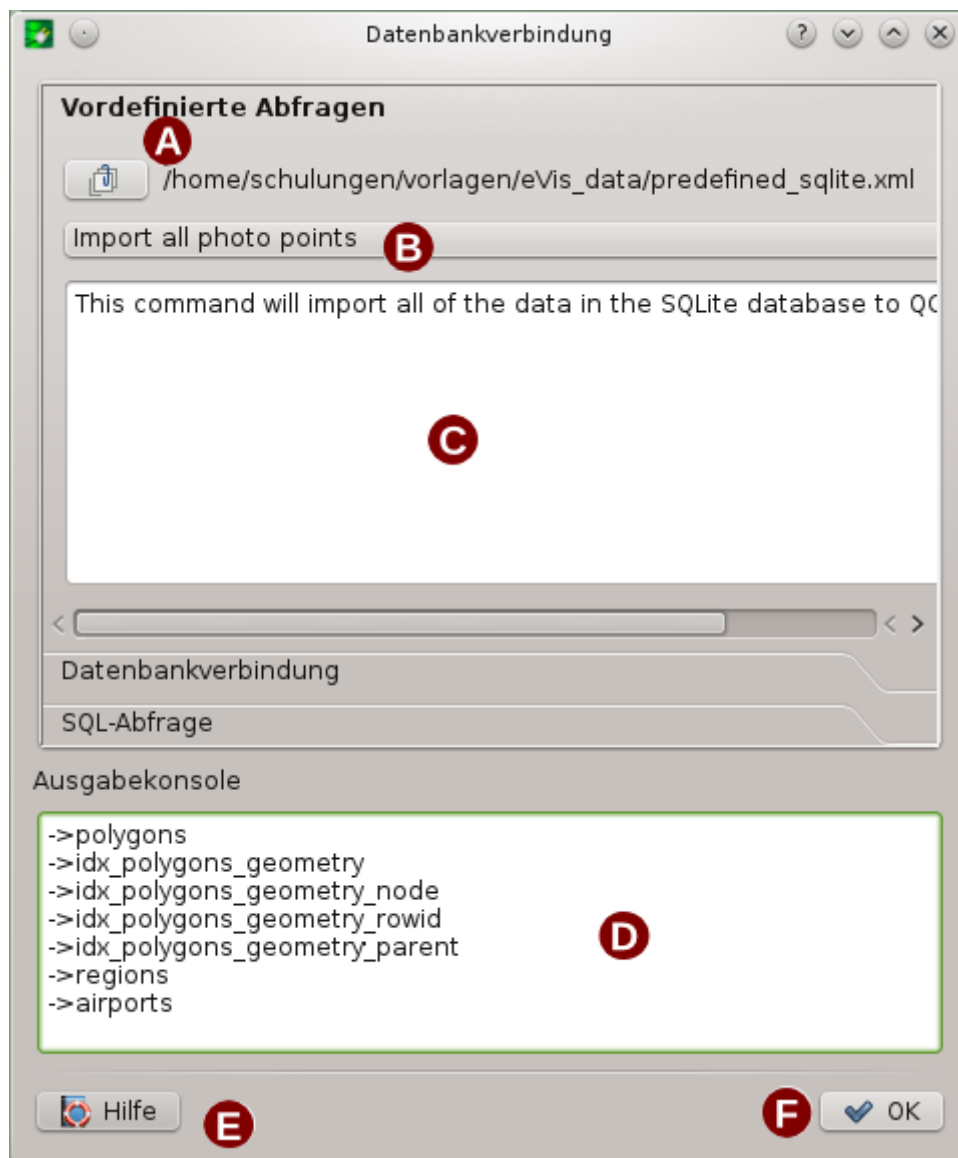


Abbildung 20.14: Der *eVis* Vordefinierte Abfragen Reiter

1. **Datei öffnen:** Öffnet den "Datei öffnen" Dateibrowser um nach der XML-Datei, die die vordefinierten Abfragen vorhält, zu suchen.
2. **Vordefinierte Abfragen:** Eine Dropdown-Liste mit allen Abfragen der bereits erstellten XML-Datei.

3. **Abfragebeschreibung:** Eine Beschreibung der Abfrage. Diese Beschreibung ist in der XML-Datei enthalten.
4. **Ausgabekonsole:** Hier werden Informationen zu den Prozessen angezeigt.
5. **Hilfe:** Zeigt die Online-Hilfe an.
6. **OK:** Schließt das Fenster 'Datenbankverbindung'.

### XML-Format für vordefinierte Abfragen in eVis

Die von eVis gelesenen XML-Tags

Tag	Beschreibung
query	Definiert den Anfang und das Ende einer Abfrage.
shortdescription	Eine kurze Beschreibung der Abfrage, die im eVis Dropdown-Menü erscheint.
description	Eine detailliertere Beschreibung der Abfrage, die in der eVis Abfragebeschreibung angezeigt wird.
databasetype	Der Datenbanktyp, definiert im Datenbanktyp Dropdownmenü im Datenbankverbindung Reiter.
databaseport	Der Port, wie er in der Port Textfenster im Datenbankverbindung Reiter definiert ist.
databasehost	Der definierte Datenbankname aus dem Datenbankname Textfeld im Reiter Datenbankverbindung.
databaseusername	Der Datenbank Benutzername wie im Username Textfeld im Datenbankverbindung Reiter definiert.
databasepassword	Das Datenbank Passwort wie im Passwort Textfeld im Reiter Datenbankverbindung definiert.
sqlstatement	Die SQL-Abfrage.
autoconnect	Eine Option ("true" oder "false"), die festlegt, ob automatisch anhand der oben angegeben Tags mit der Datenbank verbunden werden soll, ohne die Routine aus dem Reiter <i>Datenbankverbindung</i> zu durchlaufen.

Ein Beispiel einer XML-Datei mit drei Abfragen ist unten dargestellt:

```
<?xml version="1.0"?>
<doc>
  <query>
    <shortdescription>Import all photograph points</shortdescription>
    <description>This command will import all of the data in the SQLite database to QGIS
      </description>
    <databasetype>SQLITE</databasetype>
    <databasehost />
    <databaseport />
    <databasehost>C:\textbackslash Workshop\textbackslash
eVis\_Data\textbackslash PhotoPoints.db</databasehost>
    <databaseusername />
    <databasepassword />
    <sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
      Points ON Points.rec_id=Attributes.point_ID</sqlstatement>
    <autoconnect>false</autoconnect>
  </query>
  <query>
    <shortdescription>Import photograph points "looking across Valley"</shortdescription>
    <description>This command will import only points that have photographs "looking across
      a valley" to QGIS</description>
    <databasetype>SQLITE</databasetype>
    <databasehost />
    <databaseport />
    <databasehost>C:\Workshop\eVis_Data\PhotoPoints.db</databasehost>
```

```

<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
    Points ON Points.rec_id=Attributes.point_ID where COMMENTS='Looking across
    valley'</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
<query>
  <shortdescription>Import photograph points that mention "limestone"</shortdescription>
  <description>This command will import only points that have photographs that mention
    "limestone" to QGIS</description>
  <databasetype>SQLITE</databasetype>
  <databasehost />
  <databaseport />
  <databasename>C:\Workshop\eVis_Data\PhotoPoints.db</databasename>
  <databaseusername />
  <databasepassword />
  <sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
    Points ON Points.rec_id=Attributes.point_ID where COMMENTS like '%limestone%'
  </sqlstatement>
  <autoconnect>>false</autoconnect>
</query>
</doc>

```

## 20.7 fTools Plugin

Das Ziel des fTool Plugins besteht darin, eine Quelle für alltägliche, vektorbasierte GIS Aufgaben bereitzustellen, ohne zusätzliche Software, Bibliotheken oder komplexe Workarounds notwendig zu machen. Das Plugin bietet eine wachsende Zahl effektiver Werkzeuge, um räumliche Daten zu verwalten und zu analysieren.

Die fTools sind automatisch in QGIS aktiviert, können aber natürlich wie alle Plugins mit dem Plugin Manager geladen und entladen werden (Abschnitt *Der Erweiterungen Dialog*). Die Funktionen befinden sich im Menü *Vektor*, in dem Funktionen zu Analyse, Forschung, Geoverarbeitung, Geometrie und Datenmanagement bereitgestellt werden.

## 20.7.1 fTools Funktionen









Icon	Werkzeug	Zweck
	Distanzmatrix	Messen von Distanzen zwischen zwei Punktlayers. Ausgabe ist a) Standard Distanzmatrix, b) Lineare Distanzmatrix oder c) Zusammenfassende Distanzmatrix. Die Distanzen können auf die jeweils nächsten Ziele rediziert werden.
	Linienlängen summieren	Berechnet die Gesamtsumme der Linienlängen für jedes Polygon eines Polygonlayers.
	Punkte in Polygonen	Zählt die Anzahl der Punkte eines Punktlayers, die innerhalb jedes Polygons eines Polygonlayers auftreten.
	Eindeutige Werte auflisten	Listet alle eindeutigen Werte in einem Eingabe Vektorlayerfeld auf.
	Grundstatistik	Berechnet Statistikwerte für eine Attributspalte (Mittelwert, Max, Min, Standardabweichung, Anzahl, Summe, Varianzkoeffizient).
	Nächster Nachbaranalyse	Berechnet Nächste Nachbar Statistik zur Bestimmung des Clustergrades eines Punktlayers.
	Mittlere Koordinate(n)	Berechnet entweder den normalen oder gewichteten Mittelwert eines Vektorlayers oder mehrerer Objekte mit eindeutiger Schlüsselspalte.
	Linien-schnittpunkte	Findet Überschneidungen zwischen Linien und gibt diese als Shapefile aus. Linienüberschneidungen mit einer Länge > 0 werden ignoriert.

Table Ftools 1: fTools Analysewerkzeuge

## 20.7.2 Forschungswerkzeuge








Icon	Werkzeug	Zweck
	Zufällige Auswahl	Wählt zufällig n Objekte oder n Prozent der Objekte aus.
	Zufällige Auswahl in Untermengen	Wählt zufällig n Objekte aus einer Untermenge mit eindeutiger Schlüsselspalte.
	Zufällige Punkte	Erstellt pseudo-zufällige Punkte im Bereich eines Eingabelayers.
	Regelmäßige Punkte	Erstellt ein regelmäßiges Punktegitter für ein bestimmtes Gebiet und exportiert diese als Shape.
	Vektorraster	Erstellt ein Linien- oder Polyongitter mit definierten Abständen.
	Nach Position selektieren	Wähle Objekte nach relativer Position zu einem anderen Layer für eine neue Auswahl, oder addiere oder subtrahiere aus aktueller Auswahl.
	Polygon aus Layergrenzen	Erstelle ein rechteckiges Polygon auf Basis der Ausdehnung eines Raster- oder Vektorlayers.

Table Ftools 2: fTools Forschungswerkzeuge



### 20.7.3 Geoverarbeitungswerkzeuge










Icon	Werkzeug	Zweck
	Konvexe Hülle	Erstelle kleinste konvexe Hülle für einen Layer oder eine Schlüsselspalte.
	Puffer	Erstelle einen Puffer um Objekte mit definierter Distanz oder auf Basis einer Schlüsselspalte.
	Schnittmengen	Verschneide Layer, so dass die Ausgabe Flächen enthält, an denen beide Layer überlagern.
	Vereinigungen	Verschneide Layer, so dass die Ausgabe überlagernde und nicht-überlagernde Flächen enthält.
	Symmetrische Differenz	Verschneide Layer, so dass die Ausgabe Flächen enthält, an denen beide Layer nicht überlagern.
	Clip	Verschneide Layer, so dass die Ausgabe nur die Flächen enthält, die mit dem clip Layer überlagern.
	Differenz	Verschneide Layer, so dass die Ausgabe nur die Flächen enthält, die nicht mit dem clip Layer überlagern.
	Auflösen	Verschmelze Objekte auf Basis einer Schlüsselspalte. Objekte mit gleichen Werten werden zusammengefügt.
	Splitterpolygone beseitigen	Führt ausgewählte Objekte mit dem benachbarten Polygon der größten Fläche oder größten gemeinsamen Grenze zusammen.

Table Ftools 3: fTools Geoverarbeitungswerkzeuge

## 20.7.4 Geometriewerkzeuge













Icon	Werkzeug	Zweck
	Geometriegültigkeit überprüfen	Überprüft Polygone nach Überschneidungen, geschlossenen Löchern und überprüft die Knotenreihenreihenfolge.
	Geometriespalte exportieren/hinzufügen	Geometrieinformationen für Punkte (XCOORD, YCOORD), Linie (Länge) oder Polygon (Fläche, Durchmesser) hinzufügen.
	Polygonschwerpunkt	Berechne den wahren Schwerpunkt (Zentroid) von Polygonen eines Vektorlayer.
	Delaunay Triangulation	Berechnet und gibt die Delaunay Triangulation für einen Eingabe Punktvektorlayer aus (als Polygone).
	Voronoipolygone	Berechnen von Voronoi Polygonen auf Basis eines Vektor-Punktlayers.
	Geometrie vereinfachen	Generalisiere Linien oder Polygone mit modifiziertem Douglas-Peucker Algorithmus.
	Geometrien verdichten	Linien oder Polgone durch Hinzufügen von Stützpunkten verdichten.
	Mehr- zu einteilig	Konvertiere mehrteilige Objekte zu vielen einteiligen Objekten. Erstellt einfache Polygone und Linien.
	Ein- zu mehrteilig	Verbinde viele Objekte zu einem mehrteiligen Objekt auf Basis einer Schlüsselspalte.
	Polygon zu Linie	Konvertiert Polygone zu Linien, mehrteilige Polygone zu vielen einteiligen Linien.
	Linien zu Polygonen	Konvertiert Linien zu Polygonen, viele einteilige Linien zu mehrteiligen Polygonen.
	Knoten extrahieren	Extrahiere Knotenpunkte aus Linien und Polygonen und gebe sie als Punkte aus.

Table Ftools 4: fTools Geometriewerkzeuge

**Bemerkung:** Das *Geometrie vereinfachen* Werkzeug kann auch verwendet werden um doppelte Stützpunkte in Linien und Polygonen zu löschen. Setzen Sie dazu einfach die *Vereinfachungstoleranz* auf den Wert 0.

## 20.7.5 Datenmanagementwerkzeuge






Icon	Werkzeug	Zweck
	Aktuelle Projektion definieren	Definiere ein KBS für ein Shapefile, für das kein KBS definiert ist.
	Attribute nach Position zusammenfügen	Füge zusätzliche Attribute zu einem Layer auf Basis der räumlichen Lage hinzu. Attribute eines Layers werden an die Attributtabelle eines anderen Layer angehängt und als Shape exportiert.
	Vektorlayer teilen	Teile Vektorlayer in einzelne Layer auf Basis einer Schlüsselspalte.
	Shapedateien zusammenführen	Füge mehrere Shapedateien aus einem Ordner zu einer neuen Shapedatei des Typs Punkt, Linie oder Polygon zusammen.
	Räumlichen Index erzeugen	Erzeugt einen räumlichen Index für OGR-Datenformate.

Table Ftools 5: fTools Datenmanagementwerkzeuge

## 20.8 GDALTools Plugin

### 20.8.1 Was sind die GDALTools?

Die GDALTools stellen eine grafische Benutzeroberfläche bereit, über die die verschiedenen Werkzeuge der Geospatial Data Abstraction Library, <http://gdal.osgeo.org>, angesprochen werden können. Dabei handelt es sich um Raster Management Tools, z.B. für die Abfrage, Umprojizierung, Transformierung oder Verschneidung von Rasterlayern in unterschiedlichen Formaten. Außerdem stehen Werkzeuge zur Verfügung, um Konturen als (Vektor) linien zu extrahieren, eine Schummerungskarte aus Höhendaten zu erzeugen oder ein VRT (Virtual Raster Tile in XML format) aus einer oder mehreren Rasterkarten zu erzeugen. Diese Werkzeuge können benutzt werden, wenn das Plugin installiert und aktiviert wurde.

#### Die GDAL Bibliothek

Die GDAL Bibliothek besteht aus einer Reihe von Kommandozeilen-Tools, jedes mit zahlreichen Optionen. Anwender, die sich mit der Kommandozeile auskennen, werden diese Art der Anwendung sicher bevorzugen, um so den vollen Funktionsumfang nutzen zu können. Das GDAL Tools Plugin ermöglicht hingegen einen einfachen Zugang zu den Funktionen, und stellt daher auch nur die am häufigsten benutzten Optionen zur Verfügung.

### 20.8.2 Liste der GDALTools

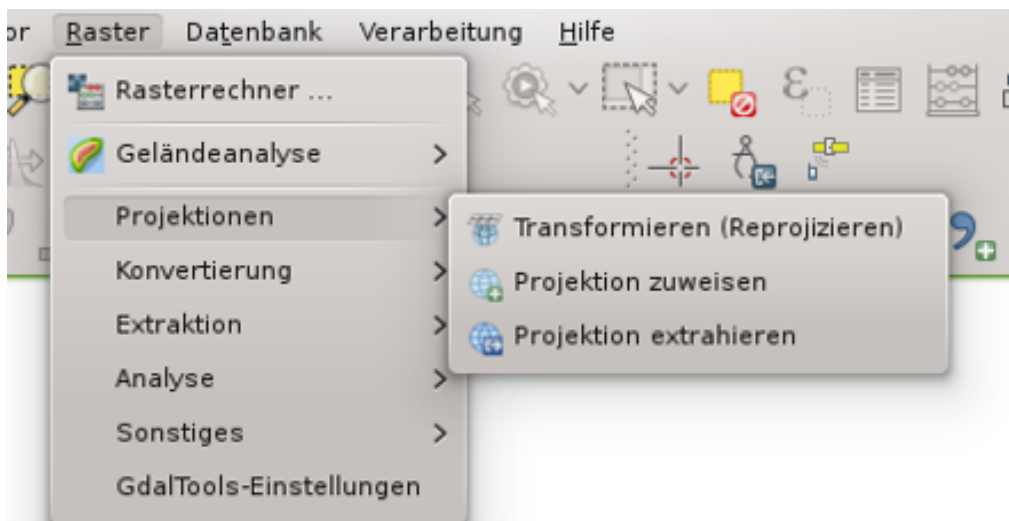










Abbildung 20.15: Die *GDALTools* Menüliste



## Projektionen

 <p>Transformieren (Reprojizieren)</p>	<p>Das Werkzeug kann mosaikieren, umprojizieren und konvertieren. Es kann in alle unterstützten Projektion projizieren und kann dabei sogar mit dem Bild bereitgestellte GCPs nutzen, wenn der Rasterlayer noch im "Rohzustand" ist. Weitere Informationen finden sich unter <a href="http://www.gdal.org/gdalwarp.html">http://www.gdal.org/gdalwarp.html</a> .</p>
 <p>Projektion zuzuweisen</p>	<p>Das Werkzeug ermöglicht es, Rasterlayern, die bereits georeferenziert sind, eine Projektionsinformation zuzuweisen oder bestehende Projektionsinformationen zu verändern. Dabei wird Einzeldatei- und Stapelverarbeitung unterstützt. Weitere Informationen finden sich unter <a href="http://www.gdal.org/gdalwarp.html">http://www.gdal.org/gdalwarp.html</a></p>
 <p>Projektion extrahieren</p>	<p>Dieses Werkzeug hilft Ihnen Projektionsinformationen aus einer Eingabedatei zu extrahieren. Wenn Sie Projektionsinformationen aus einem ganzen Verzeichnis extrahieren wollen, können Sie den Stapelverarbeitungsmodus verwenden. Es erstellt sowohl .prj als auch .wld Dateien.</p>







## Konvertierung

 <p>Rastern (Vektor nach Raster)</p>	<p>Dieses Programm brennt Vektorgeometrien (Punkte, Linien und Polygone) in Rasterkanäl(e) eines Rasterbildes. Die Vektoren werden aus OGR-unterstützten Vektorformaten gelesen. Beachten Sie, dass Vektordaten das gleiche Koordinatensystem wie die Rasterdaten haben müssen; Spontan-Reprojektion wird nicht bereitgestellt. Weitere Informationen können Sie unter <a href="http://www.gdal.org/gdal_rasterize.html">http://www.gdal.org/gdal_rasterize.html</a> nachschlagen.</p>
 <p>Polygonisieren (Raster nach Vektor)</p>	<p>Dieses Werkzeug erstellt Vektorpolygone für alle Flächen eines Rasterlayers, in denen die Pixelwerte gleich sind. Jedes Polygon erhält ein Attribut, das den Pixelwert wiedergibt. Das Werkzeug erstellt die Ausgabevektor Datenquelle wenn diese nicht bereits besteht, standardmäßig als ESRI Shapedateiformat. Weitere Informationen finden sich unter <a href="http://www.gdal.org/gdal_polygonize.html">http://www.gdal.org/gdal_polygonize.html</a>.</p>
 <p>Übersetzen (Format konvertieren)</p>	<p>Das Werkzeug kann Rasterlayer zwischen verschiedenen Formaten konvertieren. Dabei können zusätzlich Prozesse wie Ausschnitte bilden, Resampling oder auch Reskalierung angewendet werden. Weitere Informationen finden sich unter <a href="http://www.gdal.org/gdal_translate.html">http://www.gdal.org/gdal_translate.html</a></p>
 <p>RGB nach PCT</p>	<p>Dieses Werkzeug errechnet eine optimale Pseudo-Farbtabelle für einen RGB-Rasterlayer mittels eines 'median cut' Algorithmus, der auf einem downgesampten RGB-Histogramm basiert. Darauf basierend wird das Bild in ein Pseudo-Farb-Raster konvertiert. Dazu wird Floyd-Steinberg dithering (error diffusion) verwendet, um die Qualität der Ausgabe zu maximieren. Weitere Informationen finden sich unter <a href="http://www.gdal.org/rgb2pct.html">http://www.gdal.org/rgb2pct.html</a>.</p>
 <p>PCT nach RGB</p>	<p>Dieses Werkzeug konvertiert die Pseudo-Farbtabelle eines Kanals in einen RGB-Rasterlayer eines festgelegten Formats. Weitere Informationen finden sich unter <a href="http://www.gdal.org/pct2rgb.html">http://www.gdal.org/pct2rgb.html</a>.</p>






## Extraktion

 <p><i>Kontur</i></p>	<p>Dieses Werkzeug erstellt einen Vektorlayer mit den Konturlinien eines Höhenmodells (DGM). Unter <a href="http://www.gdal.org/gdal_contour.html">http://www.gdal.org/gdal_contour.html</a> finden sich weitere Informationen.</p>
 <p><i>Clipper</i></p>	<p>Dieses Werkzeug ermöglicht das Ausschneiden eines Rasterlayers (extrahieren eines Subsets) auf Basis einer definierten Ausdehnung oder eines anderen Layers/Maske. Weitere Informationen finden sich unter <a href="http://www.gdal.org/gdal_translate.html">http://www.gdal.org/gdal_translate.html</a>.</p>

## Analyse

 <p><i>Sieben</i></p>	<p>Das Werkzeug löscht Rasterflächen, die kleiner als eine festgelegter Schwellwert (in Pixeln) sind und ersetzt diese Fläche durch Pixelwerte der größten Nachbarfläche. Das Eingangsraaster kann dabei überschrieben werden, oder das Ergebnis wird in einen neuen Rasterlayer abgespeichert. Weitere Informationen finden sich unter <a href="http://www.gdal.org/gdal_sieve.html">http://www.gdal.org/gdal_sieve.html</a>.</p>
 <p><i>Fast Schwarz</i></p>	<p>Dieses Werkzeug scanned einen Rasterlayer und versucht alle Pixel, die fast schwarz (oder fast weiss) entlang von Rändern sind, in exakt schwarze (bzw. weisse) Pixel umzuwandeln. Dies wird oft verwendet, um verlusthaft komprimierte Luftbilder zu "reparieren", so dass Farbpixel bei der Mosaikierung als transparent behandelt werden können. Siehe auch <a href="http://www.gdal.org/nearblack.html">http://www.gdal.org/nearblack.html</a>.</p>
 <p><i>Leerwert füllen</i></p>	<p>Dieses Werkzeug füllt selektierte Bereiche eines Rasterlayers (normalerweise Nullwerte) durch Interpolation vorhandener Werte entlang der Ränder auf. Siehe auch <a href="http://www.gdal.org/gdal_fillnodata.html">http://www.gdal.org/gdal_fillnodata.html</a>.</p>
 <p><i>Nachbarschaft (Rasterabstand)</i></p>	<p>Werkzeug erstellt einen Raster-Abstandslayer, in dem der Abstand vom Zentrum jedes Pixels zum Zentrum eines Nachbarpixels gespeichert wird, das als Zielpixel festgelegt ist. Zielpixel sind jene Pixel im Ausgangslayer, bei denen der Pixelwert des Nachbarpixels sich in der angegebenen Liste der Zielpixelwerte befindet. Weitere Informationen unter <a href="http://www.gdal.org/gdal_proximity.html">http://www.gdal.org/gdal_proximity.html</a>.</p>
 <p><i>Gitter (Interpolation)</i></p>	<p>Dieses Werkzeug erstellt ein regelmäßiges Gitter (als Raster) aus den verstreuten Daten, die aus einer OGR-Vektordatenquelle gelesen werden. Die Eingabedaten werden dabei interpoliert, um die Gitterknoten mit Werten zu füllen. Dabei kann zwischen verschiedenen Interpolationsmethoden gewählt werden. Siehe auch <a href="http://www.gdal.org/gdal_grid.html">http://www.gdal.org/gdal_grid.html</a>.</p>
 <p><i>DHM (Geländemodelle)</i></p>	<p>Werkzeuge zum Visualisieren und Analysieren von Digitalen Höhenmodellen. Es erstellt Schummerung, Neigung, Perspektive, Farbreief, Oberflächenrauigkeit, topographische Position und Rauigkeit. Weitere Informationen unter <a href="http://www.gdal.org/gdaldem.html">http://www.gdal.org/gdaldem.html</a>.</p>

## Sonstiges

 <p><i>Virtuelles Raster erzeugen (Katalog)</i></p>	<p>Dieses Werkzeug erstellt ein VRT (Virtueller Datensatz), das ein Mosaik der GDAL Eingaberaster darstellt. Siehe auch <a href="http://www.gdal.org/gdalbuildvrt.html">http://www.gdal.org/gdalbuildvrt.html</a>.</p>
 <p><i>Verschmelzen</i></p>	<p>Dieses Werkzeug mosaikiert einen Satz von Bilddaten. Alle Layer müssen in derselben Projektion vorliegen und die gleiche Anzahl von Kanälen aufweisen. Sie können aber überlappen und eine unterschiedliche Auflösung aufweisen. Bei Überlappung wird der letzte Layer über die anderen gelegt. Siehe auch <a href="http://www.gdal.org/gdal_merge.html">http://www.gdal.org/gdal_merge.html</a>.</p>
 <p><i>Information</i></p>	<p>Das Werkzeug listet zahlreiche Informationen über einen Rasterlayer, dessen Format von GDAL gelesen werden kann. Unter <a href="http://www.gdal.org/gdalinfo.html">http://www.gdal.org/gdalinfo.html</a> finden Sie weitere Informationen.</p>
 <p><i>Übersichten erzeugen (Pyramiden)</i></p>	<p>Das gdaladdo Werkzeug wird benutzt, um Übersichten zu erstellen oder zu aktualisieren. Dabei werden 'one over several downsampling' Methoden angewendet. Siehe auch <a href="http://www.gdal.org/gdaladdo.html">http://www.gdal.org/gdaladdo.html</a>.</p>
 <p><i>Kachelindex</i></p>	<p>Dieses Werkzeug erstellt eine Shapedatei mit einem Eintrag für jede Eingaberasterdatei, einem Attribut, das den Dateinamen enthält sowie einer Polygeometrie, die das Raster umrandet. Siehe auch <a href="http://www.gdal.org/gdaltindex.html">http://www.gdal.org/gdaltindex.html</a>.</p>

## GDALTools-Einstellungen

Verwenden Sie diesen Dialog um Ihre GDAL Variablen einzubetten.

## 20.9 Georeferenzier Plugin

Das Plugin Georeferenzierung erlaubt die Erstellung von Worldfiles für existierende Rasterlayer und das Entzerren von Rasterlayern in ein neues GeoTiff. Es ermöglicht damit das Georeferenzieren von Rasterdaten in geografische und projizierte Koordinatensysteme oder die Transformation des Rasters in ein neues Koordinatensystem. Der Ansatz besteht darin, Bezugspunkte auf der Rasterkarte zu finden, denen eindeutige Koordinaten zugewiesen werden können.

### Funktionalitäten




















Icon	Funktion	Icon	Funktion
	Raster öffnen		Georeferenzierung beginnen
	GDAL Skript erzeugen		GCP Punkte laden
	GCP Punkte speichern als		Transformationseinstellungen
	Punkt hinzufügen		Punkt löschen
	GCP-Punkt verschieben		Verschieben
	Hineinzoomen		Herauszoomen
	Auf den Layer zoomen		Zoom zurück
	Zoom vor		Georeferenzierung mit QGIS verbinden
	QGIS mit Georeferenzierung verbinden		Volle Histogrammstreckung
	Lokale Histogrammstreckung		

Table Georeferencer 1: Georeferenzierfunktionen

### 20.9.1 Wie benutzt man den Georeferenzierer

Es gibt zwei Möglichkeiten, um X und Y Koordinaten (DMS (dd mm ss.ss), DD (dd.dd) einer Rohkarte oder die Koordinaten (mmmm.mm) ) einer projizierten Karte an ausgewählten Punkten eines Bildes zu georeferenzieren:


- Das Raster verfügt manchmal über Kreuze mit Koordinaten, die auf das Bild “geschrieben” sind. In diesem Fall können Sie die Koordinaten manuell eintragen.
- Bereits georeferenzierte Layer verwenden. Dies können entweder Vektor- oder Rasterdaten sein, die die gleichen Objekte/Features, die Sie auf dem Bild haben, das Sie georeferenzieren wollen, haben mit der Projektion, die Sie für Ihr Bild brauchen. In diesem Fall können Sie die Koordinaten eingeben, indem Sie auf den Referenzdatensatz, der im QGIS Kartenfenster geladen ist, klicken.

Die allgemeine Vorgehensweise besteht normalerweise darin, dass man eine Reihe von Punkten auf dem zu georeferenzierenden Bild auswählt, diesen die entsprechenden Koordinaten der Zielprojektion zuweist und dann eine passende Transformationsmethode auswählt. Entsprechend der Eingabeparameter erstellt das Plugin dann entweder einen Worldfile für das Bild oder erzeugt eine entzerrte Version des Bildes als GeoTiff. Allgemein gilt, je mehr Punkte gesetzt werden, desto besser ist das Resultat.

Der erste Schritt ist es, QGIS zu starten, das Georeferenzierung Plugin zu laden (siehe *Der Erweiterungen Dialog*) und auf *Raster* → *Georeferenzierung*, was in der QGIS Menüleiste erscheint, zu klicken. Der Georeferenzierung Plugin Dialog erscheint wie gezeigt in [figure\\_georeferencer\\_1](#).

In diesem Beispiel soll ein Worldfile für eine topografische Karte aus der Gegend Süd-Dakotas erstellt werden, welche zu dem GRASS Spearfish-Datensatz passt. 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.osgeo.org/sampled/spearfish\\_toposheet.tar.gz](http://grass.osgeo.org/sampled/spearfish_toposheet.tar.gz).

#### Eingabe von Bezugspunkten (Ground Control Points (GCP))

1. Um die Georeferenzierung eines unreferenzierten Rasters zu starten müssen wir es zuerst mit dem  Knopf laden. Das Raster erscheint im Hauptarbeitsbereich des Dialogs. Nachdem das Raster geladen wurde können wir anfangen Referenzpunkte einzugeben.
2. Fügen Sie Punkte der Hauptarbeitsfläche hinzu und geben Sie ihre Koordinaten ein (siehe Abbildung [figure\\_georeferencer\\_2](#)). Für diesen Ablauf haben Sie drei Optionen:
  - Sie klicken auf einen Punkt in der Rasterkarte und geben die X- und Y-Koordinaten ein.

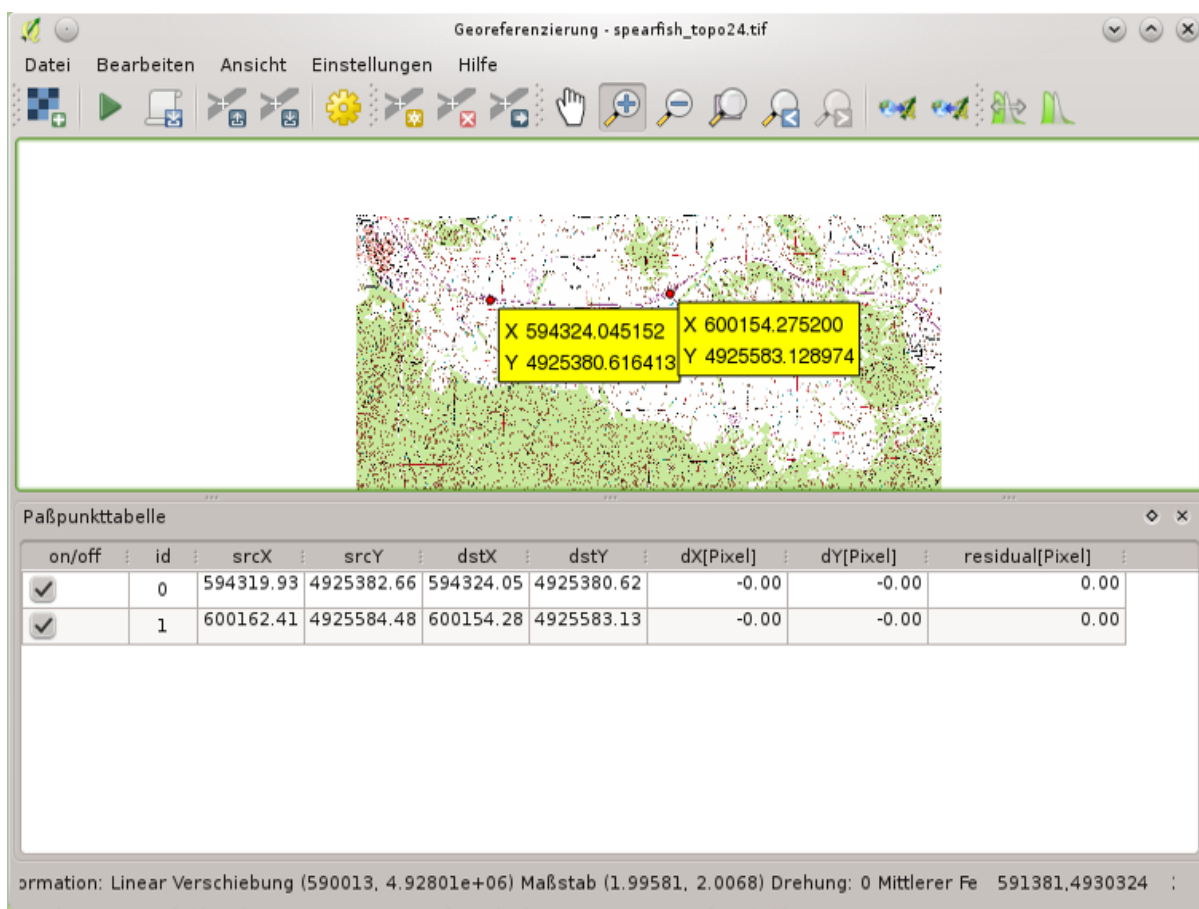




Abbildung 20.16: Georeferenzierung Plugin Dialog 



- Sie klicken auf einen Punkt in der Rasterkarte und wählen den Knopf  aus Karte, um die X- und Y-Koordinaten mit Hilfe einer georeferenzierten, in QGIS geladenen Karte hinzuzufügen.
  - Mit dem  Knopf können Sie die GCPs in beiden Fenstern verschieben, wenn Sie am falschen Platz sind.
3. Geben Sie weitere Bezugspunkte an. Sie sollten mindestens 4 Punkte festlegen, und je mehr Punkte Sie gut verteilt angeben, desto besser wird normalerweise das Ergebnis. Es gibt zusätzliche Werkzeuge im Plugin Dialog um in der Arbeitsumgebung zu zoomen und zu verschieben um einen relevanten Satz von GCP Punkten ausfindig zu machen.

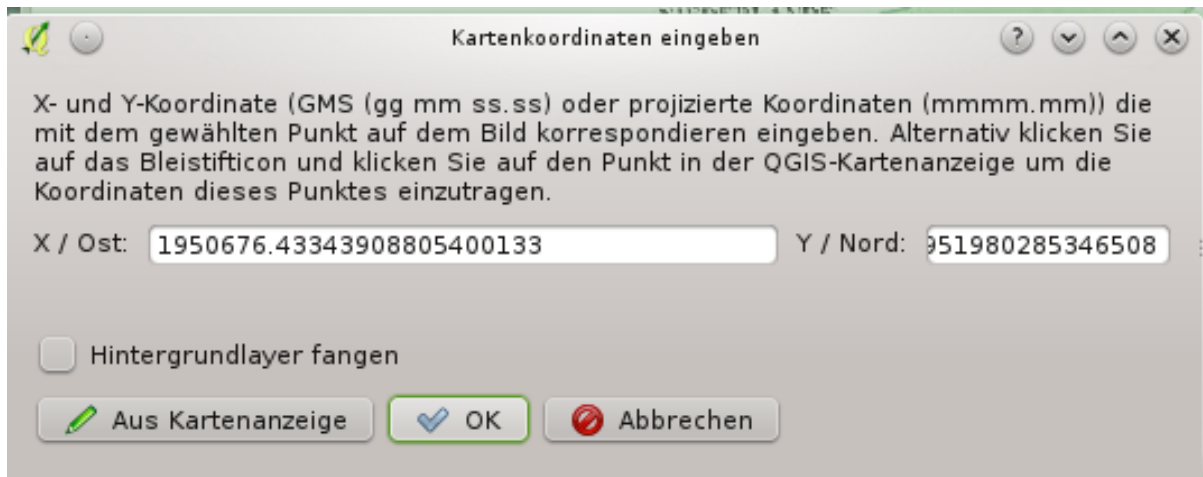




Abbildung 20.17: Dem Rasterbild Punkte hinzufügen 

Die Punkte die der Karte hinzugefügt werden werden in einer getrennten Textdatei gespeichert ([filename].points), in der Regel zusammen mit dem Rasterbild. Dies ermöglicht es uns das Georeferenzierung Plugin wieder zu einem späteren Zeitpunkt zu öffnen und neue Punkte hinzuzufügen oder bestehende zu löschen um das Ergebnis zu optimieren. Die Punktdatei enthält Wert vom Format: mapX, mapY, pixelX, pixelY. Sie können die  Passpunkte laden und  Passpunkte speichern als Knöpfe zum verwalten der Dateien verwenden.

## Festlegen der Transformationseinstellungen

Nachdem Sie in dem Bild eine ausreichende Anzahl an Punkten gesetzt haben, gilt es nun, die Transformationseinstellungen für die Georeferenzierung zu definieren.

### Auswahl des Transformationstyps

Abhängig davon, wieviele Bezugspunkte Sie gesetzt haben, stehen unterschiedliche Transformationstypen zur Verfügung. Der zu wählende Transformationstyp ist außerdem vom Typ und der Qualität der Eingangsdaten, sowie der Anzahl geometrischer Störungen, die in dem Ergebnis auftreten können, abhängig.

Derzeit stehen die folgenden *Transformationstypen* zur Verfügung:

- Der **Linear** Algorithmus wird verwendet eine Worlddatei zu erstellen und unterscheidet sich von der anderen Algorithmen, da er das Raster nicht wirklich transformiert. Dieser Algorithmus wird wahrscheinlich nicht ausreichen wenn Sie mit gescanntem Material zu tun haben.
- Die **\*\*Helmert\*** Transformation führt einfache Skalierungs- und Rotationstransformationen durch.
- Die **Polynomial** Algorithmen 1–3 sind unter den am meisten verwendeten Algorithmen, die eingeführt wurden, um Quell- und Zielkontrollpunkten zu entsprechen. Der am weitesten verbreitete polynomische Algorithmus ist die Polynomische Transformation zweiter Ordnung, die eine gewisse Krümmung ermöglicht.

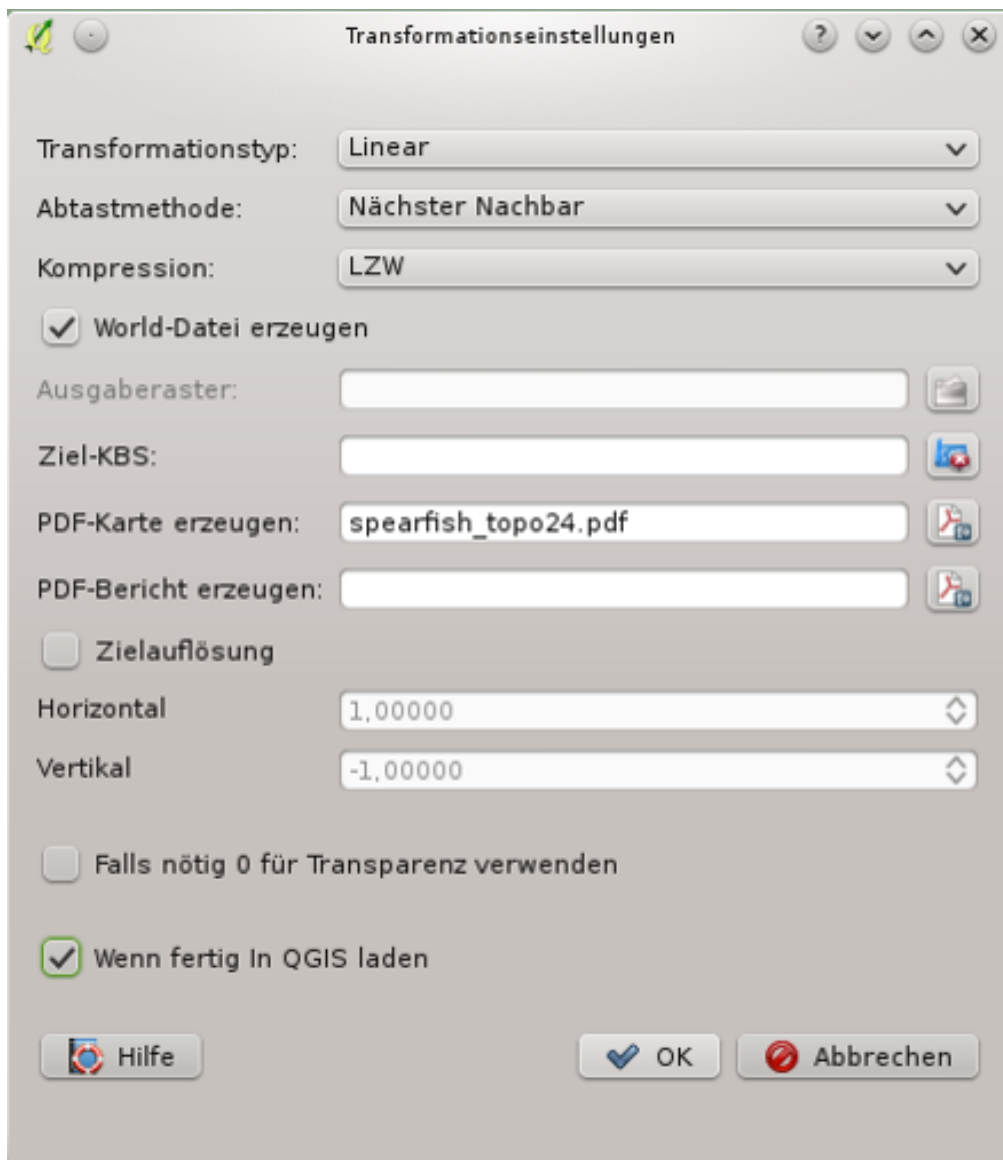


Abbildung 20.18: Definieren der Georeferenzierung Transformationseinstellungen 

Die Polynomische Transformation erster Ordnung (Affintransformation) bewahrt nur die Kollinearität und erlaubt nur die Skalierung, Übersetzung und Drehung.

- Der **Thin Plate Spline** (TPS) Algorithmus ist eine modernere Georeferenzierungsmethode, die lokale Deformationen in die Daten einführen kann. Dieser Algorithmus ist nützlich wenn Originalbilder mit sehr schlechter Qualität georeferenziert werden.
- Die *Projektiv\** Transformation ist eine lineare Drehung und Übersetzung von Koordinaten.

### Festlegen der Resampling Methode

Die verwendete Resampling Methode wird wahrscheinlich von den Eingabedaten und dem Ziel der Übung abhängig sein. Wenn die Bildstatistik nicht verändert werden soll, wählen Sie wahrscheinlich die Nächster Nachbar Methode, wo hingegen die Kubische Methode ein eher weicheres Ergebnis ergibt.

Es ist möglich zwischen 5 verschiedenen Resamplingmethoden auszuwählen:

1. Nächster Nachbar
2. Linear
3. Kubisch
4. Kubisches Spline
5. Lanczos

### Festlegen der Transformationseinstellungen


Es gibt mehrere Optionen, die für die Ausgabe der Georeferenzierung festgelegt werden müssen.

- Das  *World-Datei erzeugen* Kontrollkästchen steht nur zur Verfügung wenn Sie sich entschließen den linearen Transformationstyp zu verwenden, das dies heisst dass das Rasterbild tatsächlich nicht umgewandelt wird. In diesem Fall ist das *Ausgaberraster* Feld nicht aktiviert, da nur eine neue World-Datei erstellt wird.
- Für alle anderen Transformationstypen müssen Sie ein *Ausgaberraster* angeben. Als Standard wird eine Datei mit dem Namen ([filename]\_modified) in demselben Ordner, indem sich auch die Originaldatei befindet geschrieben.
- Als nächsten Schritt müssen Sie das *Ziel-KBS* (Koordinatenbezugssystem) für die georeferenzierte Rasterdatei definieren (siehe *Arbeiten mit Projektionen*).
- Wenn Sie mögen können Sie eine **PDF-Karte erzeugen** und auch einen **PDF-Bericht erzeugen**. Der Bericht enthält Informationen über die verwendeten Parameter, ein Bild mit den Residuen und eine Liste mit allen GCPs und ihrer RMS Fehler.
- Darüberhinaus können Sie das  *Zielauflösung* Kontrollkästchen aktivieren und die Pixelauflösung des Ausgaberrasters definieren. Die voreingestellte horizontale und vertikale Auflösung ist 1.
- *Falls nötig 0 für Transparenz verwenden* kann aktiviert werden wenn Pixel mit dem Wert 0 transparent dargestellt werden sollen. In unserer topografischen Karte wären alle weißen Bereiche transparent.
- Schließlich lädt  *Wenn fertig in QGIS laden* das Ausgaberraster automatisch in das QGIS Kartenfenster wenn die Transformation abgeschlossen ist.


### Rastereigenschaften anzeigen und anpassen

Wenn Sie auf den Knopf *Raster properties* im Menü *Einstellungen* klicken, öffnet sich ein Dialog, um die Rastereigenschaften des zu referenzierenden Layers zu verändern.

## Den Georeferenzierer konfigurieren




- Sie können definieren, ob Sie GCP Koordinaten und/oder IDs anzeigen wollen.
- Als Resteinheiten können Pixel und Karteneinheiten ausgewählt werden.
- Für den PDF-Bericht kann ein linker und rechter Rand definiert werden und Sie können auch das Papierformat für die PDF-Karte festlegen.
- Schließlich können Sie  *Georeferenzierungsfenster docken* aktivieren.

## Starten der Georeferenzierung


Nachdem alle GCPs gesetzt worden sind und alle Transformationseinstellungen definiert worden sind drücken Sie einfach den  *Georeferenzierung beginnen* Knopf um eine neue georeferenzierte Rasterdatei zu erstellen.

## 20.10 Interpolationsplugin

Die Interpolationserweiterung kann verwendet werden um eine TIN oder IDW Interpolation aus einem Punktvektorlayer zu erzeugen. Es ist sehr einfach zu bedienen und stellt eine intuitive Benutzeroberfläche zum Erstellen von interpolierten Rasterlayern bereit (siehe [Figure\\_interpolation\\_1](#)). Die Erweiterung benötigt die folgenden Parameter, die vor dem Ausführen angegeben werden müssen:

- **Eingabevektor:** Wählen Sie einen in QGIS geladenen Vektor Punktlayer(s). Wenn mehrere Layer angegeben werden, werden die Daten aller Layer für die Interpolation verwendet. Beachten Sie auch, dass es möglich ist, Linien und Polygone als Randbedingungen für die Triangulation zu verwenden, indem Sie in der Kombobox  *:guilabel‘Typ‘* des geladenen Layers entweder “Punkte”, “Strukturlinien” oder “Bruchkanten” auswählen.
- **Interpolationsattribut:** Wählen Sie die Attributspalte für die Interpolation oder aktivieren Sie  *Z-Koordinate für Interpolation verwenden* um die im Layer gespeicherten Z-Werte zu verwenden.
- **Interpolationsmethode:** Wählen Sie die Interpolationsmethode. Dies kann entweder ‘Unregelmäßiges Dreiecksnetz (TIN)’ oder ‘Inverse Distanzgewichtung (IDW)’ sein.
- **Spalten-/Zeilenanzahl:** Geben Sie die Anzahl von Zeilen und Spalten für die Ausgaberrasterdatei an.
- **Ausgabedatei:** Legen Sie einen Namen für den Ausgabebayer fest.
-  *Ergebnis zum Projekt hinzufügen* um das Ergebnis in die Kartenansicht zu laden.

### 20.10.1 Das Plugin anwenden

1. Starten Sie QGIS und laden Sie einen Punktvektorlayer (z.B. `elevp.csv`).
2. Laden Sie die Interpolationserweiterung in den Plugin Manager (siehe *Der Erweiterungen Dialog*) und klicken Sie auf *Raster* → *Interpolation* →  *Interpolation*, was in der QGIS Menüleiste erscheint. Der Interpolationserweiterungsdialoگ erscheint wie in [Figure\\_interpolation\\_1](#) gezeigt.
3. Wählen Sie den Layer ‘elevp’ als Eingabevektorlayer und Spalte ‘ELEV’ als Interpolationsattribut.
4. Wählen Sie eine Interpolationsmethode (z.B. ‘Unregelmäßiges Dreiecksnetz (TIN)’), und geben Sie eine Zellgröße von 5000 genauso wie den Ausgabedateinamen (z.B. `elevation_tin`) an.
5. Klicken Sie auf [OK].

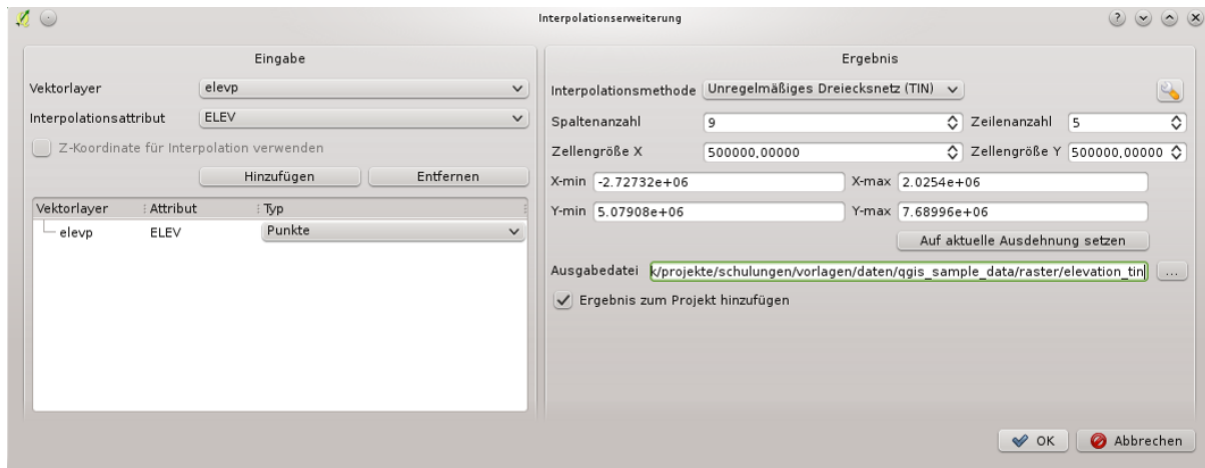





Abbildung 20.19: Interpolationserweiterung 

## 20.11 Offline-Bearbeitung Plugin


Bei der Datenerfassung ist es eine alltägliche Situation, um mit einem Laptop oder Smartphone im Gelände offline zu arbeiten. Nach der Rückkehr müssen die Änderungen wieder mit der Master-Datenquelle (z.B. einer PostGIS Datenbank) synchronisiert werden. Wenn mehrere Personen gleichzeitig mit denselben Datenbeständen arbeiten, ist es meist schwierig, die Änderungen von Hand zu verschmelzen, selbst wenn unterschiedliche Objekte verändert wurden.

Das  Offline-Bearbeitung Plugin automatisiert die Synchronisation durch Kopieren des Inhalts einer Datenquelle (in der Regel PostGIS oder WFS-T) zu einer SpatialLite Datenbank mit dedizierten Tabellen. Nachdem man wieder mit dem Netzwerk verbunden ist, können die Offline-Änderungen wieder an die Master-Datenquelle zurückgespielt werden.

### 20.11.1 Verwendung der Erweiterung

- Öffnen Sie einige Vektorlayer (z.B. von einer PostGIS- oder WFS-T-Datenquelle).
- Speichern Sie es als Projekt.
- Gehen Sie zu *Datenbank* → *Offline-Bearbeitung* →  *Zu Offline-Projekt konvertieren* und wählen Sie die zu speichernden Layer. Der Inhalt der Layer wird in SpatialLite gespeichert.
- Editieren Sie die Layers offline.
- Nachdem Sie sich wieder verbunden haben laden Sie die Änderungen mit *Datenbank* → *Offline-Bearbeitung* →  *Synchronisieren* hoch.

## 20.12 Oracle-Spatial-GeoRaster Plugin

Oracle Datenbanken mit Oracle Spatial Erweiterung ermöglichen es, Rasterlayer als SDO\_GEORASTER Objekte zu speichern. In QGIS existiert das  Oracle-Spatial-GeoRaster Plugin. Es basiert auf der GDAL Bibliothek und setzt voraus, dass eine entsprechende Oracle Datenbank auf ihrem Rechner läuft. Obwohl Oracle keine freie Software ist, stellen Sie ihre Software für Entwickler und zu Testzwecken kostenlos zur Verfügung. Ein einfaches Beispiel, wie man über GDAL ein Raster in ein GeoRaster laden kann sieht folgendermaßen aus:

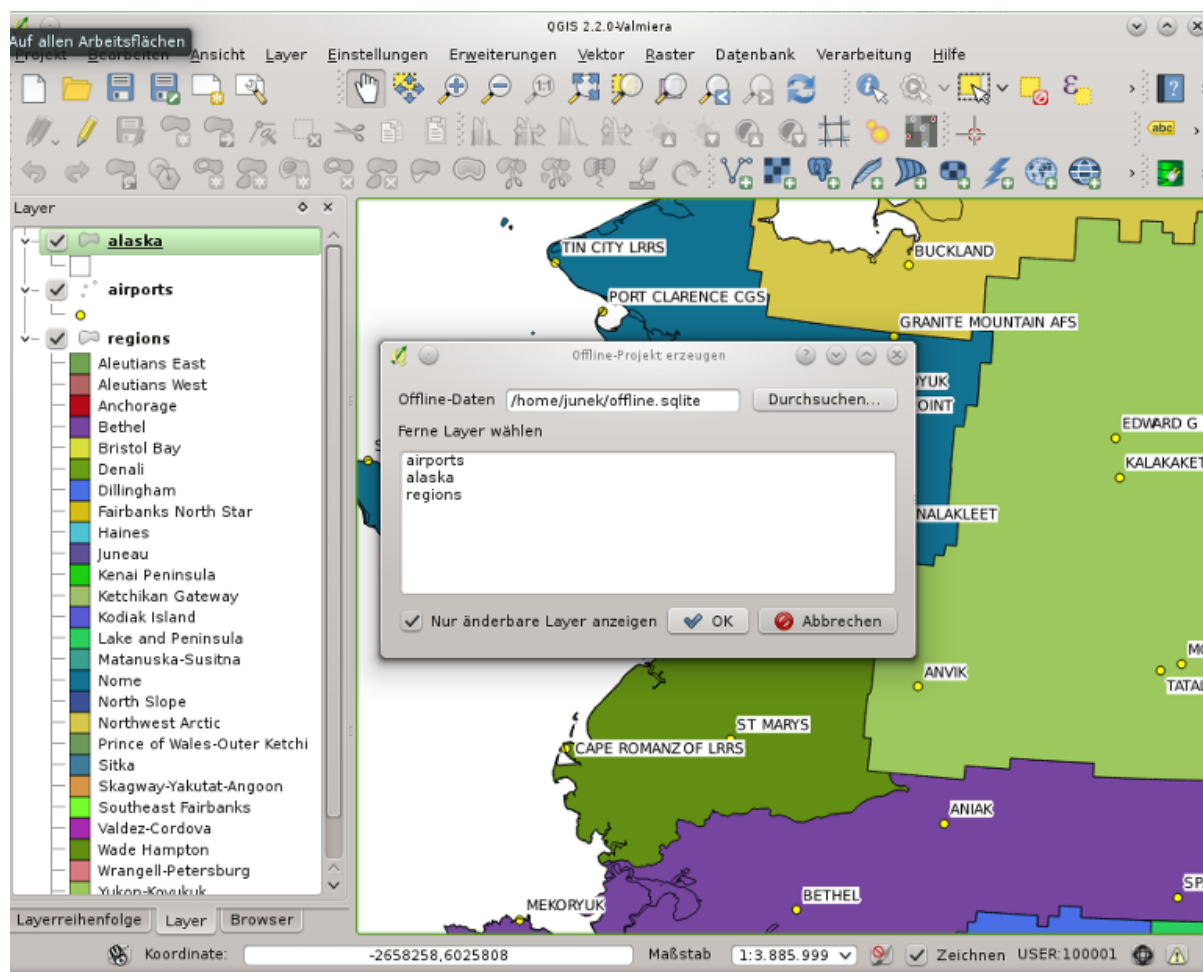



Abbildung 20.20: Ein Offline-Projekt aus PostGIS- oder WFS-Layern erstellen

```
$ gdal_translate -of georaster input_file.tif geor:scott/tiger@orcl
```

Das Raster wird in diesem Beispiel in die Standard GDAL\_IMPORT Tabelle als Spalte mit dem Namen RASTER geladen.

### 20.12.1 Mit der Datenbank verbinden

Als erstes muss das Oracle GeoRaster Plugin mit dem Plugin Manager geladen werden (siehe Kapitel *Der Erweiterungen Dialog*). Wenn Sie zum ersten Mal ein GeoRaster in QGIS laden wollen, müssen Sie zuvor eine Verbindung zu der Oracle Datenbank erstellen, in der sich die Daten befinden. Hierzu klicken Sie auf das  Icon in der Werkzeugleiste. In dem Dialog klicken Sie auf **[Neu]** und geben dann die notwendigen Verbindungsparameter ein (siehe Abbildung *Figure\_oracle\_raster\_1*):

- **Name:** Geben Sie einen Namen für die Datenbankverbindung ein.
- **Datenbankinstanz:** Geben Sie den Namen der Datenbank mit der Sie sich verbinden werden ein.
- **Benutzername:** Geben Sie Ihren eigenen Benutzernamen, den Sie verwenden werden um sich mit der Datenbank zu verbinden, an.
- **Passwort:** Vergeben Sie das Passwort das Ihrem Benutzernamen zugeordnet ist und das erforderlich ist um die Datenbank anzubinden.



Abbildung 20.21: Oracle-Verbindung herstellen Dialog

Zurück im Hauptfenster des Oracle Spatial GeoRaster Plugins (siehe Abbildung *Figure\_oracle\_raster\_2*), wählen Sie die Dropdown Liste, um die neue Verbindung auszuwählen und klicken dann auf **[Verbinden]**, um die Verbindung herzustellen. Sie können die Verbindung auch nochmals **[Bearbeiten]** und Veränderungen vornehmen oder mit dem Knopf **[Löschen]** die Verbindung aus der Dropdown Liste entfernen.

### 20.12.2 Ein GeoRaster auswählen

Nachdem eine Verbindung eingerichtet wurde, zeigt das Unterdaten Fenster die Namen aller Tabellen die GeoRaster-Spalten in der Datenbank enthalten und das Format von GDAL Unterdatennamen haben.

Wählen Sie einen der 'subdatasets' und klicken dann auf **[Wählen]**, um den Tabellennamen auszuwählen. Daraufhin erscheint eine weitere Liste mit den GeoRaster Spalten, die sich in der Tabelle befinden. Dies ist normalerweise eine kurze Liste, da es eher selten vorkommt, dass mehr als ein oder zwei GeoRaster Spalten in einer Tabelle abgelegt sind.

Klicken Sie auf einen der aufgelisteten Unterdatensätze und klicken Sie dann auf **[OK]** um eine der Tabellen-/Spaltenkombinationen auszuwählen. Der Dialog zeigt nun alle Zeilen die GeoRaster-Objekte enthalten. Beachten Sie, dass die Unterdaten-Liste jetzt die Raster Data Table und Raster Id pairs zeigt.

Der Auswahleintrag kann zu jeder Zeit bearbeitet werden um direkt zu einem bekannten GeoRaster zu gehen oder zurück zum Anfang zu gehen und einen anderen Tabellennamen auszuwählen.

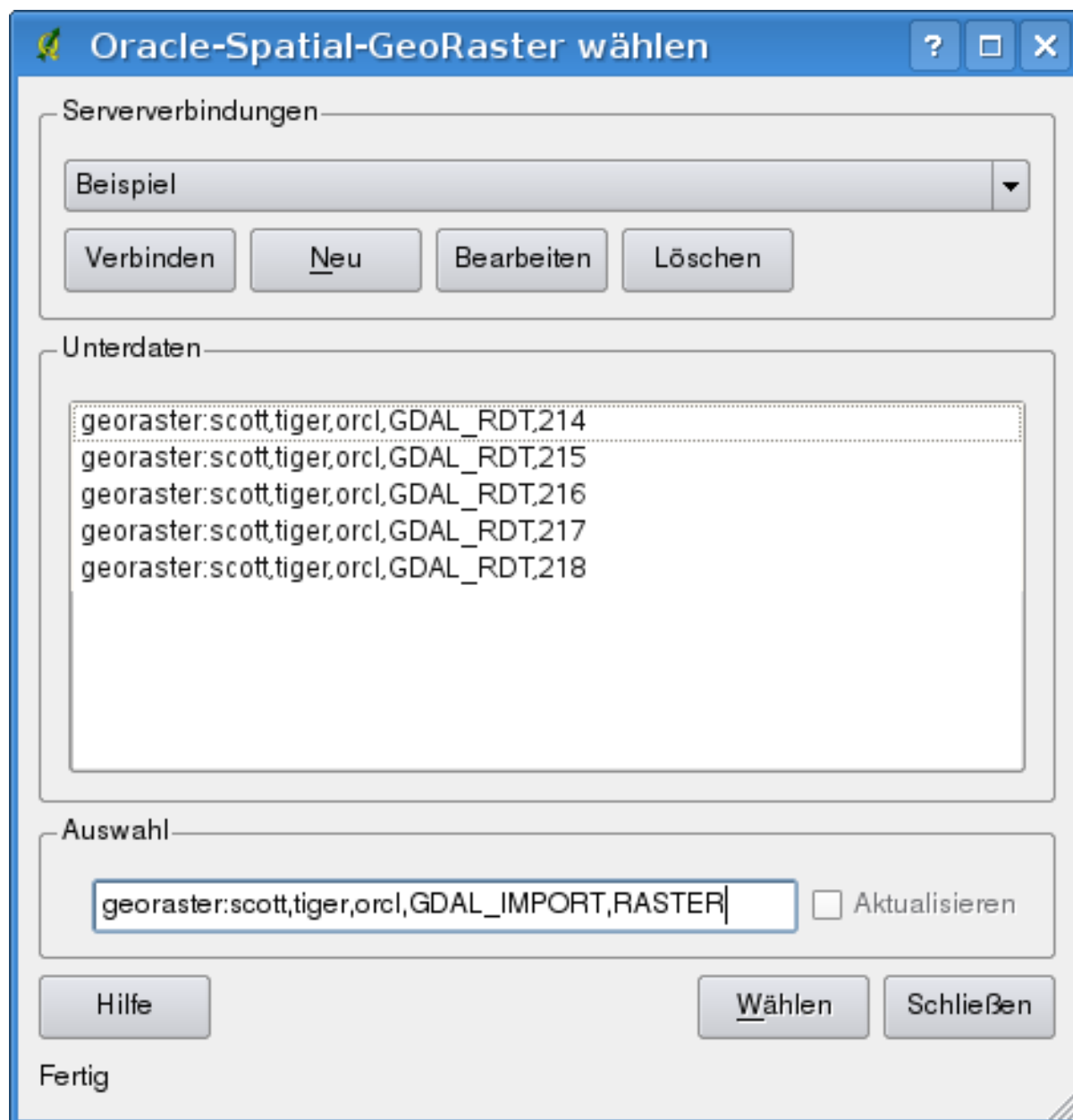


Abbildung 20.22: Oracle-Spatial-GeoRaster wählen Dialog

Der Auswahl Dateneintrag kann auch dazu verwendet werden eine WHERE Klausel am Ende des Identifikationsstrings (z.B. `geor:scott/tiger@orcl,gdal_import,raster,geoid=`) einzugeben. Siehe [http://www.gdal.org/frmt\\_georaster.html](http://www.gdal.org/frmt_georaster.html) für weitere Informationen.

### 20.12.3 Ein GeoRaster laden

Schließlich wird das Rasterbild durch Auswählen eines GeoRasters aus der Liste der Raster Data Tables und Raster Ids in QGIS geladen.



Der *Oracle-Spatial-GeoRaster wählen* Dialog kann jetzt geschlossen werden und wenn er das nächste Mal geöffnet wird, wird die gleiche Verbindung beibehalten und wird die gleiche vorherige Liste von Unterdatensätzen gezeigt, was es sehr einfach macht ein anderes Bild aus dem gleichen Kontext zu öffnen.

**Bemerkung:** GeoRaster, die mit Pyramiden abgelegt sind, werden in QGIS wesentlich schneller visualisiert. Die Pyramiden müssen aber im Vorfeld und außerhalb von QGIS mit Oracle PL/SQL oder gdaladdo erstellt werden.

Das folgende ist ein Beispiel, bei dem gdaladdo verwendet wird:

```
gdaladdo georaster:scott/tiger@orcl,georaster\_table,georaster,georid=6 -r
nearest 2 4 6 8 16 32
```

Beispiel zum Erstellen von Pyramiden mit PL/SQL:

```
$ sqlplus scott/tiger
SQL> DECLARE
  gr sdo_georaster;
BEGIN
  SELECT image INTO gr FROM cities WHERE id = 1 FOR UPDATE;
  sdo_geor.generatePyramid(gr, 'rLevel=5, resampling=NN');
  UPDATE cities SET image = gr WHERE id = 1;
  COMMIT;
END;
```

## 20.13 Rastergeländeanalyse-Erweiterung



Die Rastergeländeanalyse-Erweiterung kann verwendet werden um Neigung, Perspektive, Schummerung, Rauigkeitsindex und Relief von Digitalen Geländemodellen (DGM) zu berechnen. Es ist sehr einfach anzuwenden und stellt eine intuitive Benutzeroberfläche zum Erstellen neuer Rasterlayer bereit (siehe [Figure\\_raster\\_terrain\\_1](#)).

Description of the analysis:

- **Neigung:** Berechnet den Neigungswinkel für jede Zelle in Grad (beruht auf Ableitung erster Ordnung).
- **Perspektive:** Berechnung der Exposition. Beginnend mit 0 für Nord und dann in Grad gegen den Uhrzeigersinn.
- **Schummerung:** Erstellen einer Schummerungskarte auf Basis von Licht und Schatten, um ein 3D-ähnliches Erscheinungsbild zu schaffen.
- **Rauhigkeitsindex:** Eine quantitative Bestimmung der Heterogenität eines Geländes beschrieben durch Riley et al. (1999). Es wird für jeden Ort mit einem 3x3 Pixel Fenster berechnet.
- **Relief:** Erstellen einer Shaded Relief Karte auf Basis eines DGM. Integriert ist eine Methode, um auf Basis der Häufigkeitsverteilung der Höhenwerte Farbwerte zuzuweisen.

### 20.13.1 Das Plugin anwenden

1. Starten Sie QGIS und laden Sie den `gtopo30` Rasterlayer aus der GRASS Beispiel-Location.
2. Laden Sie die Rastergeländeanalyse-Erweiterung im Plugin Manager (siehe Abschnitt *Der Erweiterungen Dialog*).
3. Wählen Sie eine Analysemethode (z.B.: *Raster* → *Geländeanalyse* → *Neigung*). Der *Neigung* Dialog erscheint wie in [Figure\\_raster\\_terrain\\_1](#) gezeigt.
4. Geben Sie eine Ausgabedatei mit Pfad und Dateiformat an.
5. Klicken Sie **[OK]**.

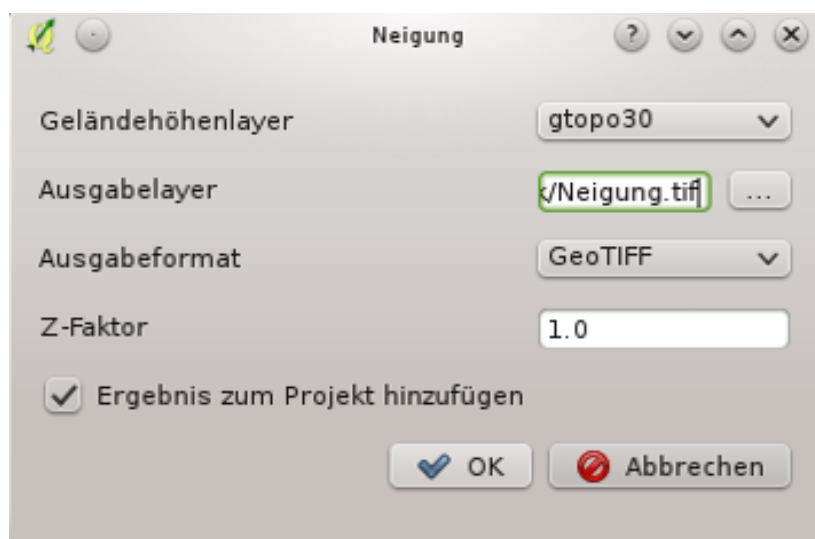



Abbildung 20.23: Rastergeländeanalyse-Erweiterung (Neigungsberechnung)

## 20.14 Heatmap-Erweiterung

Die *Heatmap*-Erweiterung verwendet Kernel Density Estimation um ein Dichte- (Heatmap) Raster eines Eingabe Punktvektorlayers zu erstellen. Die Dichte wird auf Grundlage der Anzahl von Punkten an einem Ort berechnet, wobei eine größere Anzahl von geclusterten Punkten höhere Werte zum Ergebnis haben. Heatmaps ermöglichen die einfache Identifikation von "Hotspots" und Punktclustern.

### 20.14.1 Das Heatmap Plugin starten


Zuerst muss diese Kernerweiterung anhand des Erweiterungsmanagers aktiviert werden (siehe *Der Erweiterungen Dialog*). Nach der Aktivierung kann das Heatmap Icon  in der Rasterwerkzeuggeste sowie im *Raster* → *Heatmap* → Menü gefunden werden.

Wählen Sie das Menü *Einstellungen* → *Werkzeugkästen* → *Raster* um die Rasterwerkzeuggeste anzuzeigen falls diese nicht sichtbar ist.


### 20.14.2 Das Heatmap Plugin verwenden

Das Klicken auf das  *Heatmap* Werkzeug öffnet den Heatmap-Erweiterung Dialog (siehe [figure\\_heatmap\\_2](#)).

Der Dialog hat folgende Optionen:

- **Eingabepunktlayer:** Listet alle Vektorpunktlayer im aktuelle Projekt auf und wird dazu benutzt den Layer, der analysiert werden soll, auszuwählen.
- **Ausgaberraster:** Ermöglicht es Ihnen den  Knopf zum Auswählen des Ordners und Dateinamens für das Ausgaberraster, das die Heatmap-Erweiterung erzeugt, zu verwenden. Eine Dateierweiterung ist nicht erforderlich.
- **Ausgabeformat:** Wählt das Ausgabeformat aus. Wenn auch alle von GDAL unterstützten Formate ausgewählt werden können ist GeoTIFF in den meisten Fällen das am besten geeignete Format.
- **Radius:** Wird verwendet um den Heatmap Suchradius (oder Kernbandbreite) in Metern oder Karteneinheit- en anzugeben. Der Radius gibt den Abstand um einen Punkt ab dem der Einfluss des Punktes spürbar wird

an. Größere Werte haben eine stärkere Glättung zur Folge, kleinere Werte können aber feinere Details und eine Abwechslung in der Punktdichte zeigen.

Wenn das  *Erweitert* Kontrollkästchen aktiviert ist, stehen zusätzliche Optionen zur Verfügung:

- **Zeilen und Spalten:** Werden dafür verwendet die Abmessungen des Ausgaberasters zu verändern. Diese Werte sind auch mit den **Zellengröße X** und **Zellengröße Y** Werten verbunden. Das Erhöhen der Anzahl Zeilen oder Spalten verringert die Zellgröße und vergrößert die Dateigröße der Ausgabe-datei. Die Werte in den Zeilen und Spalten sind ebenfalls verbunden, also verdoppelt das Verdoppeln der Anzahl von Zeilen automatisch die Anzahl von Spalten und die Zellgrößen werden ebenfalls halbiert. Das geografische Gebiet des Ausgaberasters bleibt das Gleiche!
- **Zellengröße X und Zellengröße Y:** Kontrollieren Sie die geografische Größe jedes Pixels im Ausgaberaster. Das Verändern dieser Wert verändert ebenfalls die Anzahl von Zeilen und Spalten im Ausgaberaster.
- **Kernform:** Die Kernform kontrolliert die Rate mit der der Einfluss auf einen Punkt abnimmt während der Abstand zum Punkt steigt. Verschiedene Kernel verfallen in verschiedenen Raten, also legt ein Triweight-Kernel größeres Gewicht auf Objekte, die näher am Punkt liegen, als der Epanechnikov Kernel. Folglich hat Triweight "schärfere" Hotspots zum Ergebnis und Epanechnikov "weichere" Hotspots. In QGIS steht eine Reihe von Standard-Kernel-Funktionen zur Verfügung, wie beschrieben und illustriert auf [Wikipedia](#).
- **Verfallrate:** Kann bei Dreieck Kernformen verwendet werden um weiter zu kontrollieren wie der Heat eines Objekts mit der Entfernung zum Objekt sinkt.
  - Ein Wert von 0 (=Minimum) zeigt an, dass das Heat im Zentrum des vorgegebenen Radius konzentriert wird und an der Kante vollständig gelöscht wird.
  - Ein Wert von 0.5 gibt an dass Pixel am Rand von Radius den halben heat wie Pixel im Zentrum des Suchradius erhalten.
  - Ein Wert von 1 heisst das der heat gleichmäßig über den gesamten Suchradiuskreis verteilt wird. (Dies entspricht dem 'Uniform' Kernel.)
  - Ein Wert größer als 1 gibt an, dass das Heat entgegen der Kante des Suchradius größer ist als im Zentrum.

Der Eingabepunktlayer kann ebenfalls Attributfelder, die das Heatmap beeinflussen, beinhalten:




- **Feld für Radius:** Setzt den Suchradius für jedes Objekt aus einem Attributfeld im Eingabelayer.
- **Feld für Gewichtung:** Ermöglicht es Eingabeobjekte anhand eines Attributfeldes zu gewichten. Dies kann verwendet werden um den Einfluss, den bestimmte Objekte auf die Ergebniskarte haben, zu erhöhen.

Wenn ein Ausgaberaster-Dateiname angegeben ist, kann der [OK] Knopf dazu verwendet werden, das Heatmap zu erstellen.

### 20.14.3 Tutorial: Erstellen eines Heatmap

Für das folgende Beispiel werden wir den `airports` Punktvektorlayer aus dem QGIS Beispieldatensatz (siehe *Beispieldaten*) verwenden. Einen anderes exzellentes QGIS Tutorial über das Erstellen von Heatmaps finden Sie unter <http://qgis.spatialthoughts.com>.

In *Figure\_Heatmap\_1* werden die airports von Alaska gezeigt.

1. Das Klicken auf das  *Heatmap* Werkzeug öffnet den Heatmap-Erweiterung Dialog (siehe *Figure\_heatmap\_2*).
2. Wählen Sie im *Eingabepunktlayer*  Feld die `airports` aus der Liste von Punktlayers, die ins aktuelle Projekt geladen sind, aus.
3. Geben Sie einen Ausgabedateinamen an indem Sie den  Knopf neben dem *Ausgaberaster* Feld klicken. Geben Sie den Dateinamen `heatmap_airports` ein (es ist keine Dateierweiterung erforderlich).
4. Lassen Sie das *Ausgabeformat* als voreingestelltes Format "GeoTIFF".

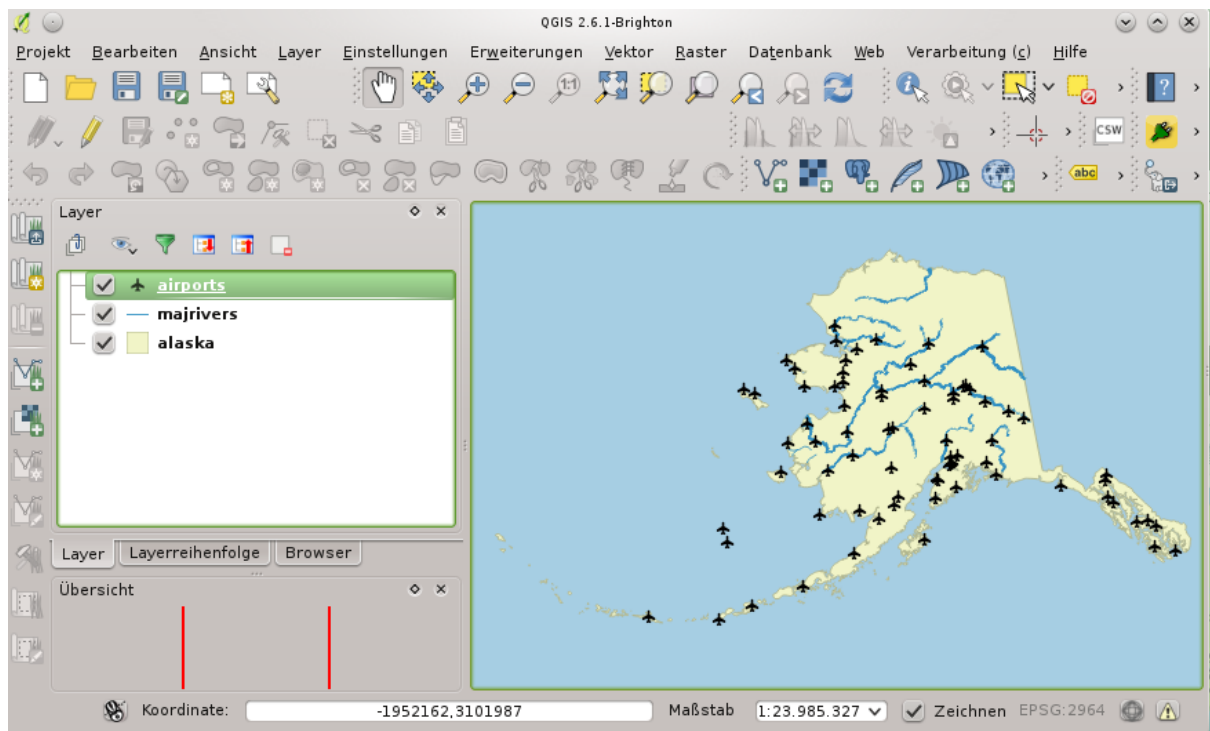




Abbildung 20.24: Aiports von Alaska 🐧

5. Ändern Sie den *Radius* auf 1000000 Meter.
6. Klicken Sie auf **[OK]** um das airports Heatmap zu erstellen und zu laden (siehe [Figure\\_Heatmap\\_3](#)).

QGIS erstellt das Heatmap und fügt die Ergebnisse Ihrem Kartenfenster hinzu. Standardmäßig wird das Heatmap in Graustufen schattiert, wobei hellere Bereiche höhere Konzentrationen von airports zeigen. Das Heatmap kann jetzt in QGIS dargestellt werden um sein Aussehen zu verbessern.

1. Öffnen Sie den Eigenschaftendialog des `heatmap_airports` Layers (wählen Sie den Layer `heatmap_airports`, öffnen Sie das Kontextmenü mit der rechten Maustaste und wählen Sie *Eigenschaften*).
2. Wählen Sie den Reiter *Stil*.
3. Ändern Sie die *Darstellungsart*  in 'Einkanalpseudofarbe'.
4. Wählen Sie eine geeignete *Farbbildung* , beispielsweise YlOrRed.
5. Klicken Sie den **[Laden]** Knopf um die Minimum- und Maximumwerte aus dem Raster zu holen und klicken Sie dann auf den **[Klassifizieren]** Knopf.
6. Klicken Sie **[OK]** um den Layer zu updaten.

Das Endergebnis sehen Sie in [Figure\\_Heatmap\\_4](#).

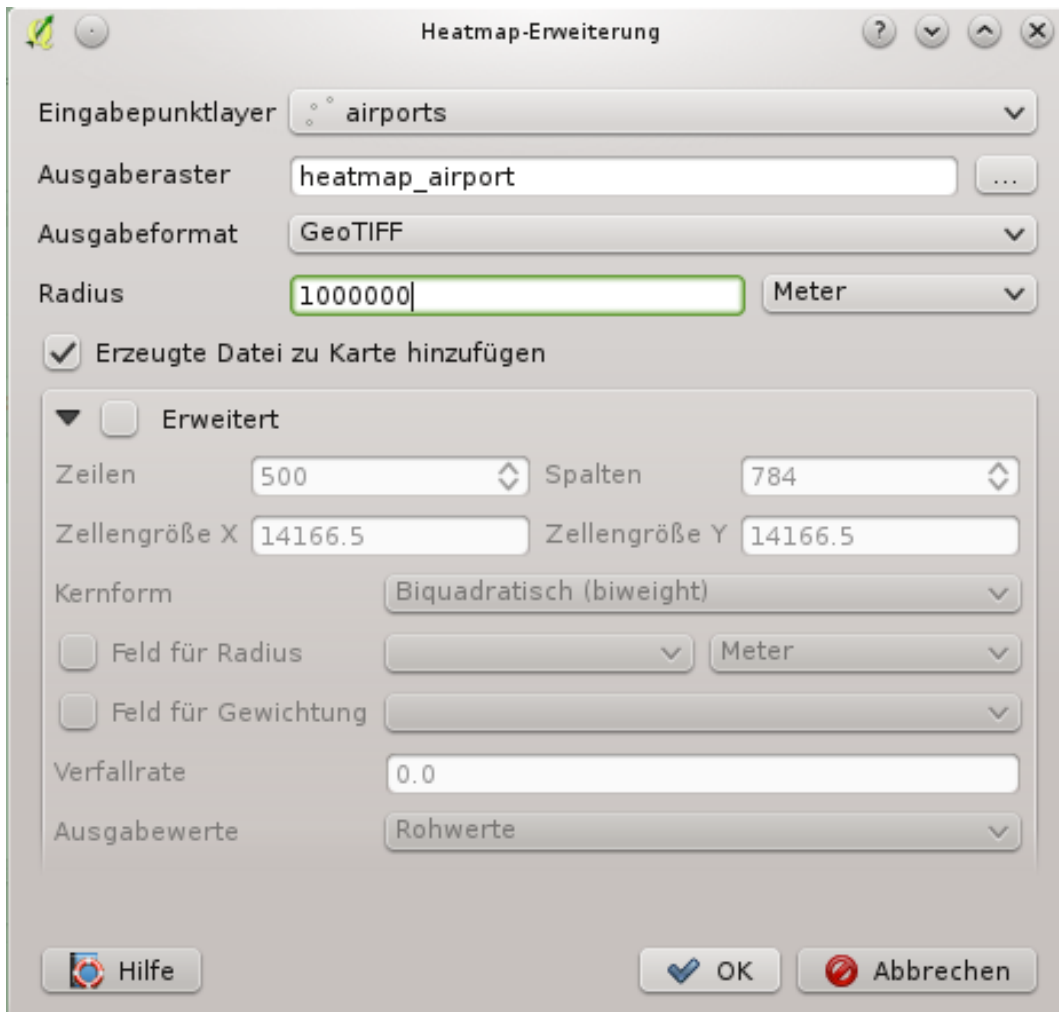


Abbildung 20.25: Der Heatmap-Erweiterung Dialog 

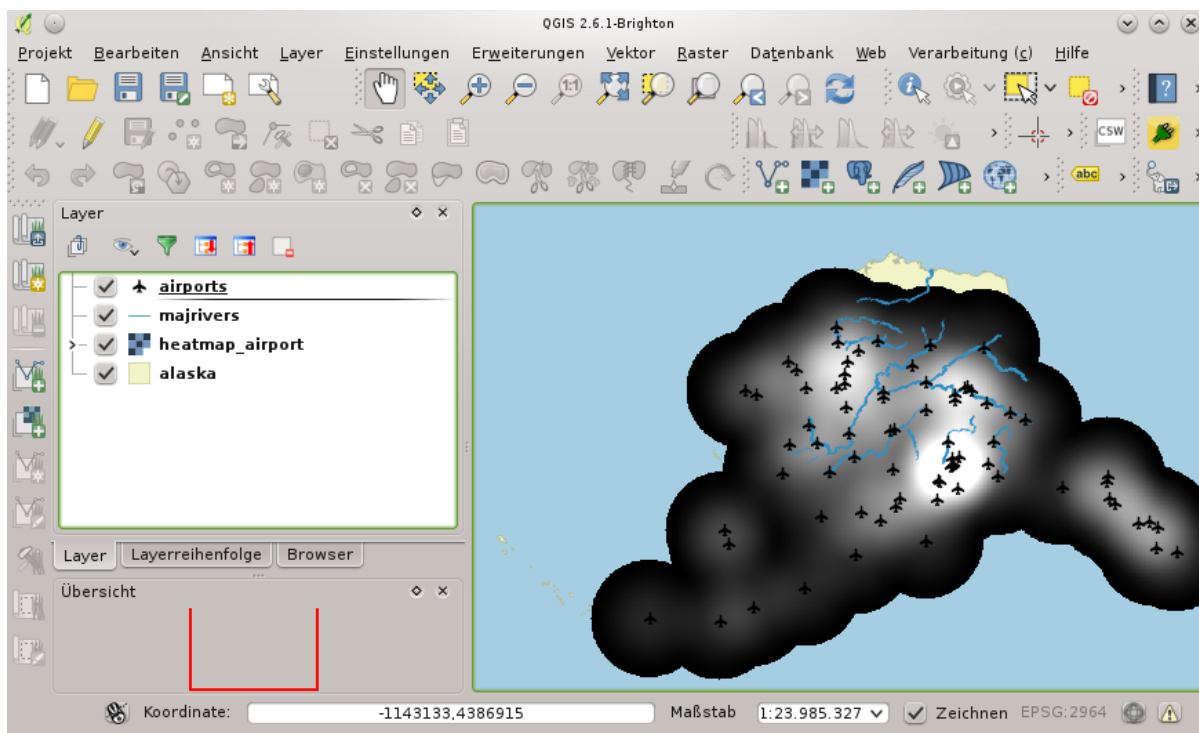


Abbildung 20.26: Nach dem Laden sieht das Heatmap wie eine graue Oberfläche aus 🐧

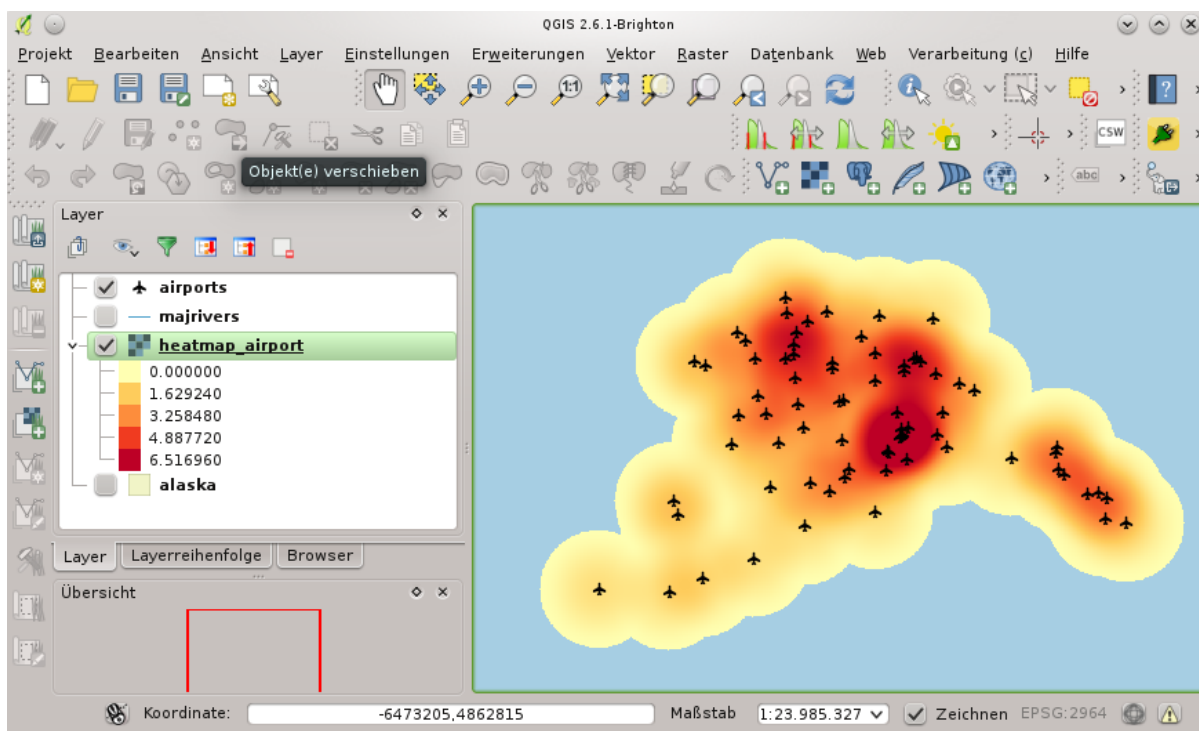
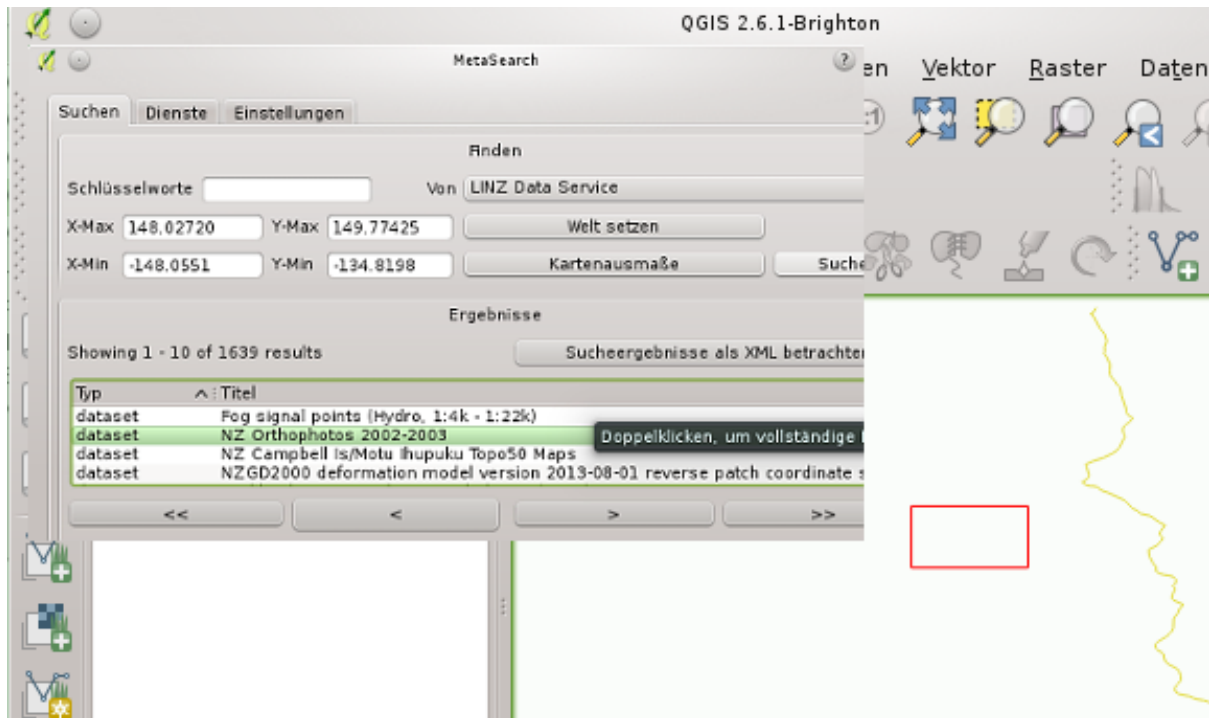


Abbildung 20.27: Heatmap der airports von Alaska mit geeignetem Stil 🐧

## 20.15 MetaSearch Katalog Client



### 20.15.1 Einleitung

MetaSearch ist eine QGIS Erweiterung um mit Metadatenkatalogservices zu interagieren und unterstützt den OGC Katalogservice für das Web (CSW) Standard.

MetaSearch stellt eine einfache und intuitiv zu benutzende Benutzerfreundliche Bedienoberfläche um Metadatenkataloge innerhalb von QGIS zu durchsuchen zur Verfügung.

### 20.15.2 Installation

MetaSearch ist standardmäßig in QGIS 2.0 und höher enthalten. Alle Abhängigkeiten sind in MetaSearch enthalten.

Metasearch wird über den QGIS Erweiterungsmanager oder manuell von <http://plugins.qgis.org/plugins/MetaSearch> installiert.

### 20.15.3 Arbeiten mit Metadatenkatalogen in QGIS

#### CSW (Katalog Services für das Web)

CSW (Catalogue Service for the Web) ist eine OGC (Open Geospatial Consortium) Spezifikation die allgemeine Schnittstellen definiert um Metadaten über Daten, Services und andere potentielle Ressourcen zu entdecken, zu durchsuchen und abzufragen.

#### Start

Um Metasearch zu starten klicke das MetaSearch Icon oder wähle Web / MetaSearch / MetaSearch mittels des QGIS Hauptmenüs. Der MetaSearch Dialog taucht auf. Das Hauptmenü besteht aus zwei TABS: 'Services' und 'Suche'.



## Verwaltung von Katalogservices



Der 'Services' TAB erlaubt die Verwaltung aller verfügbaren Katalogservices. MetaSearch stellt eine Standardliste von Katalogservices zur Verfügung die man über den 'Füge Standard Services hinzu' Button erreicht.

Für alle aufgeführten Catalogue Service Einträge klicken Sie die Drop-down-Auswahl-Box.

Um einen Katalog Service hinzuzufügen klicken sie auf 'Neu' und geben sie einen Namen und die Adresse des Service ein. Beachten Sie das nur die BasisURL notwendig ist (nicht die Vollständige GetCapabilities URL). Ein Klick auf OK fügt den Service hinzu.

Um einen existierenden Katalogservice zu editieren wählen sie den betreffenden Eintrag aus und klicken die 'Edit' Schaltfläche. Ändern Sie Namen und/oder URL und klicken auf OK.

Um einen Katalogservice zu löschen wählen Sie den Eintrag den Sie löschen wollen und klicken auf 'Löschen'. Eine Sicherheitsabfrage die Sie bestätigen müssen wird erscheinen.

MetaSearch erlaubt Verbindungen in XML Dateien zu speichern. Dies ist nützlich um Einstellungen mit anderen Programmen zu teilen. Unterhalb findet sich ein Beispiel des XML Dateiformats.

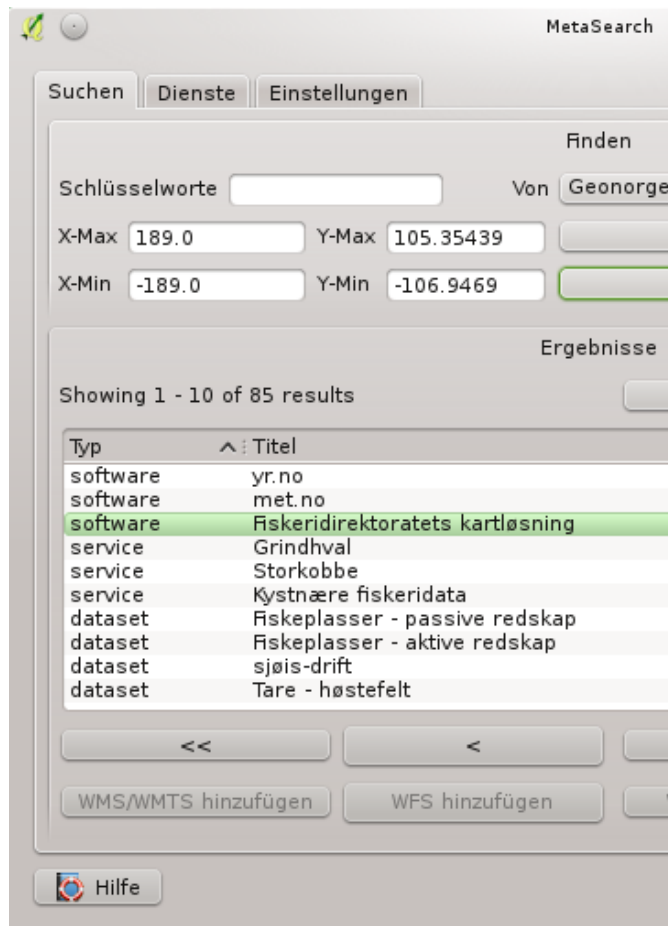
```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
  <csw name="Data.gov CSW" url="http://catalog.data.gov/csw-all"/>
  <csw name="Geonorge - National CSW service for Norway" url="http://www.geonorge.no/geonetwork/srv/eng/csw"/>
  <csw name="Geoportale Nazionale - Servizio di ricerca Italiano" url="http://www.pcn.minambiente.it/geonetwork/srv/eng/csw"/>
  <csw name="LINZ Data Service" url="http://data.linz.govt.nz/feeds/csw"/>
  <csw name="Nationaal Georegister (Nederland)" url="http://www.nationaalgeoregister.nl/geonetwork/srv/eng/csw"/>
  <csw name="RNDT - Repertorio Nazionale dei Dati Territoriali - Servizio di ricerca" url="http://www.rndt.it/geonetwork/srv/eng/csw"/>
  <csw name="UK Location Catalogue Publishing Service" url="http://csw.data.gov.uk/geonetwork/srv/eng/csw"/>
  <csw name="UNEP/GRID-Geneva Metadata Catalog" url="http://metadata.grid.unep.ch:8080/geonetwork/srv/eng/csw"/>
</qgsCSWConnections>
```



Um eine Liste von Einträgen zu laden klicken Sie den ‘Laden’ Knopf. Ein neues Fenster erscheint; klicken Sie den ‘Durchsuchen’ Knopf und navigieren Sie zu der XML Datei von Einträgen die Sie laden wollen und klicken Sie ‘Öffnen’. Die Liste von Einträgen wird angezeigt. Wählen Sie die Einträge die Sie von der Liste hinzufügen wollen und klicken Sie ‘Laden’.

The ‘Service info’ button displays information about the selected Catalogue Service such as service identification, service provider and contact information. If you would like to view the raw XML response, click the ‘GetCapabilities response’ button. A separate window will open displaying Capabilities XML.

### Catalogue Services suchen



Der ‘Suchen’ Reiter ermöglicht es dem Benutzer Catalogue Services nach Daten und Diensten abzufragen, zahlreiche Suchparameter einzusetzen und Ergebnisse anzuschauen.

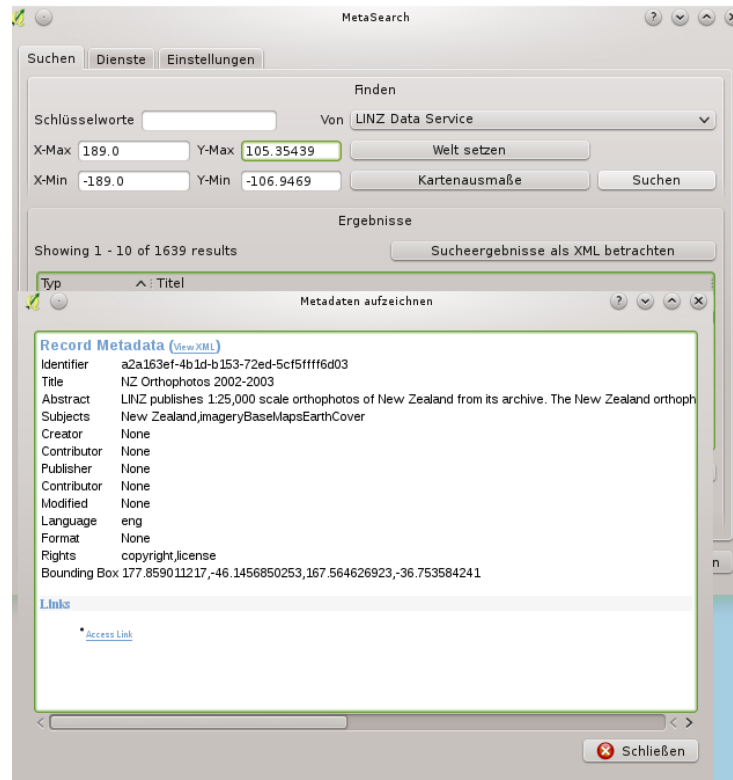
Die folgenden Suchparameter stehen zur Verfügung:

- **Schlüsselworte:** freie Textsuchschlüsselwörter
- **Von:** der Catalogue Service mit dem die Abfrage durchgeführt werden soll
- **Begrenzungsbox:** das räumliche Interessengebiet das gefiltert werden soll. Die voreingestellte Begrenzungsbox ist die Kartenansicht. Klicken Sie ‘Welt setzen’ um eine globale Suche durchzuführen oder geben Sie benutzerdefinierte Werte nach Wunsch ein
- **Ergebnisse:** die Anzahl von Einträgen die bei der Suche wiedergegeben werden. Voreingestellt sind 10 Einträge

Das Klicken des ‘Suche’ Knopfes sucht den ausgewählten Metadata Catalogue. Suchergebnisse werden in einer Liste dargestellt und können mit einem Klick auf den Spaltennamen sortiert werden. Sie können durch Suchergebnisse navigieren mit den Richtungsknöpfen unter den Suchergebnissen. Das Klicken des ‘Suchergebnisse als XML betrachten’ Knopfes öffnet ein Fenster mit der Dienstanwort im rohen XML Format.

Das Klicken eines Ergebnisses zeigt eine Zusammenfassung im 'Zusammenfassung' Fenster und stellt die folgenden Optionen zur Verfügung:

- wenn der Metadata Eintrag eine damit verbundene Begrenzungsbox hat wird ein Footprint auf der Karte angezeigt
- das Doppelklicken des Eintrages stellt die Metadaten des Eintrags mit allen damit verbundenen Zugangslinks dar. Das Klicken des Links öffnet den Link im Webbrowser des Anwenders
- wenn der Eintrag ein OGC Webservice ist (WMS/WMTS, WFS, WCS) werden die entsprechenden 'WMS/WMTS/WFS/WCS hinzufügen' Knöpfe für den Anwender aktiviert um diese QGIS hinzuzufügen. Der OWS wird dann zur entsprechenden QGIS Verbindungsliste hinzugefügt und der entsprechende WMS/WMTS/WFS/WCS Verbindungsdialog erscheint dann



## Einstellungen

Sie können mit den folgenden Einstellungen eine Feineinstellung von MetaSearch durchführen:

- **Ergebnisseiten:** wenn nach Metadata Catalogues gesucht wird die Anzahl von Ergebnissen die pro Seite gezeigt werden
- **Timeout:** wenn nach Metadata Catalogues gesucht wird die Anzahl von Sekunden während der Verbindungsversuche blockiert werden. Vorgabewert ist 10

## 20.16 Straßengraph Plugin

Das Straßengraph-Erweiterung ist ein C++ Plugin, mit dem man die kürzeste Verbindung zwischen zwei Punkten entlang eines Polyline Vektorlayers berechnen kann und den Pfad über dem Straßennetz plottet.

Hauptfunktionen:

- Berechnet den Pfad genauso wie die Länge und Reisezeit.

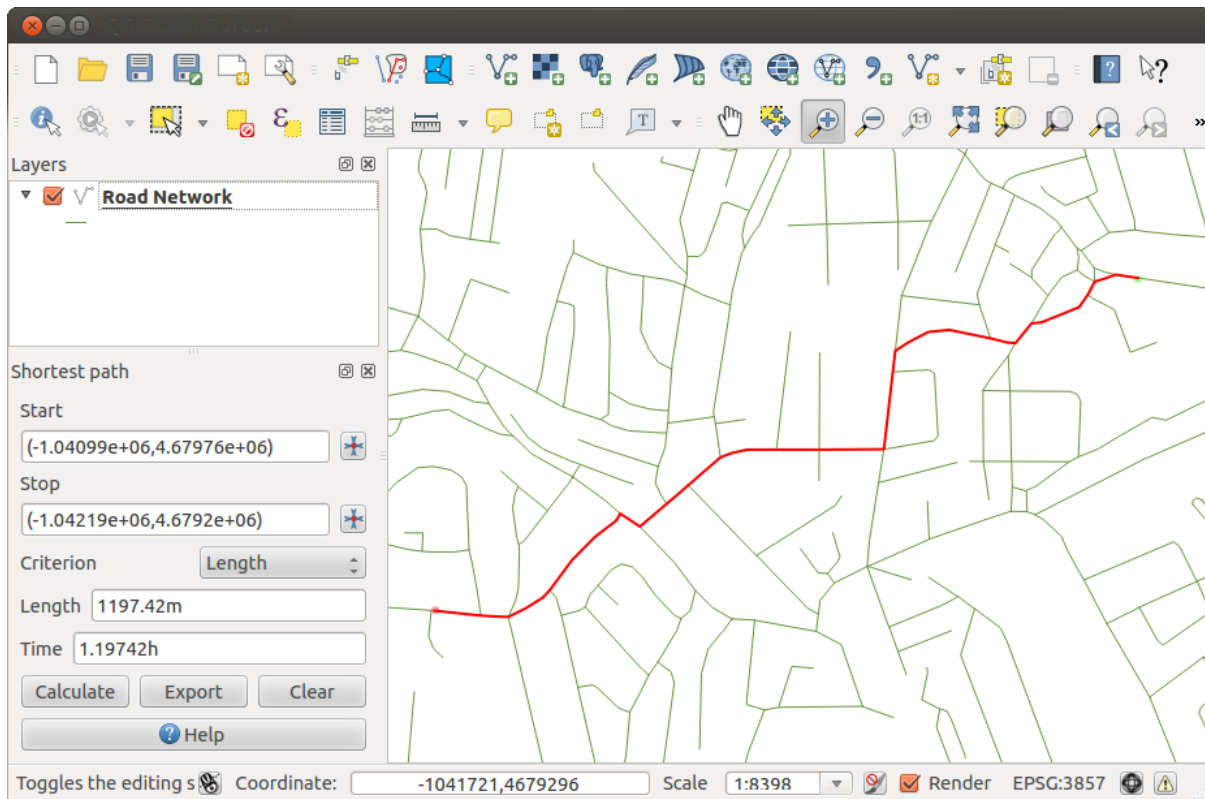


Abbildung 20.28: Straßengraph-Erweiterung 🐧

- Optimiert anhand der Länge oder der Reisezeit.
- Exportiert dem Pfad in einen Vektorlayer.
- Hebt Straßenrichtungen hervor (dies ist langsam und wird hauptsächlich für Debug-Zwecke und für das Testen von Einstellungen verwendet).

Als Netzwerk kann jeder Polyline Vektorlayer verwendet werden, der in einem von QGIS unterstützten Format gespeichert ist. Zwei Linien mit einem gemeinsamen Punkt werden dabei als verknüpft angesehen. Wichtig ist, dass das Layer-KBS als Projekt-KBS gesetzt werden muss. Dies ist wichtig, da Neuberechnungen von Koordinaten ansonsten zu Fehlern führen können, selbst eine Fangtoleranz eingestellt ist.

In der Attributtabelle des Layers können die folgenden Felder benutzt werden:

- Geschwindigkeit auf der Straße Abschnitt (numerisches Feld).
- Richtung (jeder Typ, der in einen String umgewandelt werden kann). Vorwärts und Rückwärts Richtungen beziehen sich auf eine Einbahnstraße, beide Richtungen zeigen eine zweisepurige Straße an.

Wenn einige Zeilen keine Werte haben, werden die Defaultwerte verwendet. Sie können bei Bedarf geändert werden, gemeinsam mit ein paar weiteren Einstellungsmöglichkeiten im Erweiterungseinstellungen Dialog .

### 20.16.1 Verwendung der Erweiterung

Nach der Aktivierung des Plugins sehen Sie ein zusätzliches Bedienfeld auf der linken Seite des QGIS Hauptfensters. Geben Sie jetzt einige Parameter im *Straßengraphen-Erweiterungseinstellungen* Dialog aus dem *Vektor* → *Straßengraph* Menü ein (siehe [figure\\_road\\_graph\\_2](#)).

Nachdem Sie die *Zeiteinheit*, *Distanzeinheit* und *Topologietoleranz* gesetzt haben können Sie den Vektorlayer im *Verkehrslayer* Reiter auswählen. Hier können Sie auch das *Richtungsfeld* und das *Geschwindigkeitsfeld* wählen. Im Reiter *Voreinstellungen* können Sie die *Richtung* für die Berechnung setzen.

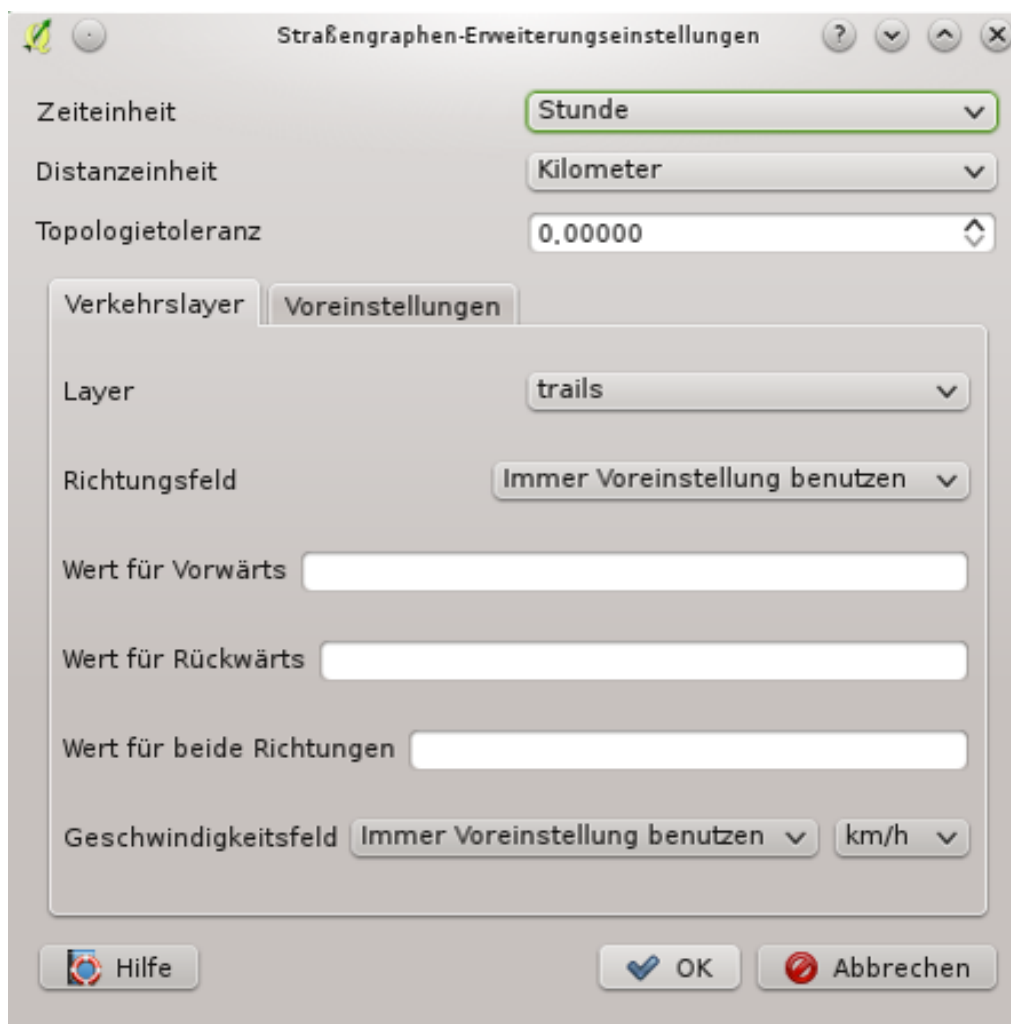




Abbildung 20.29: Straßengraph-Erweiterung Einstellungen 

Wählen Sie schließlich im *Kürzester Weg* Bedienfeld einen Start und einen Stopp Punkt im Straßennetzwerk Layer und klicken Sie auf [**Berechnen**].

## 20.17 Räumliche Abfrage Plugin


Das  Die Räumliche Abfrageerweiterung ermöglicht es Ihnen eine räumliche Abfrage (z.B. Objekte auswählen) in einem Ziellayer mit Bezug auf einen anderen Layer zu erstellen. Die Funktionalität basiert auf der GEOS Bibliothek und hängt vom ausgewählten Quellobjekt Layer ab.

Mögliche Operatoren sind:




- Enthält
- Gleich
- Überlappt
- Kreuzt
- Überschneidet
- Ist ausserhalb
- Berührt
- Innerhalb

### 20.17.1 Verwendung der Erweiterung

Als Beispiel sollen die regions Alaskas gefunden werden, die airports enthalten. Folgende Schritte sind notwendig:

1. Starten Sie QGIS und laden Sie die Vektorlayer `regions.shp` und `airports.shp`.
2. Laden Sie die Räumliche Abfrageerweiterung in den Plugin Manager (siehe *Der Erweiterungen Dialog*) und klicken Sie auf das  Räumliche Abfrage Icon, das in der QGIS Werkzeugleiste erscheint. Der Erweiterungsdialog erscheint.
3. Wählen Sie den Layer `regions` als Quelllayer und `airports` als Referenzobjektlayer.
4. Wählen Sie 'Innerhalb' als Operator und klicken Sie [**Anwenden**].

Jetzt erhalten Sie eine Liste von Objekt IDs aus der Abfrage und Sie haben mehrere Optionen, wie in `figure_spatial_query_1` gezeigt.

- Klicken Sie auf  Layer mit Liste von Elementen erzeugen
- Wählen Sie eine ID aus der Liste und klicken Sie auf  Layer mit gewählten erzeugen
- Wählen Sie 'Aus aktueller Auswahl entfernen' im Feld *Das Ergebnis speichern in* .
- Zusätzlich können Sie  *Zum Element zoomen* oder ein  *Protokoll anzeigen*.

## 20.18 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,

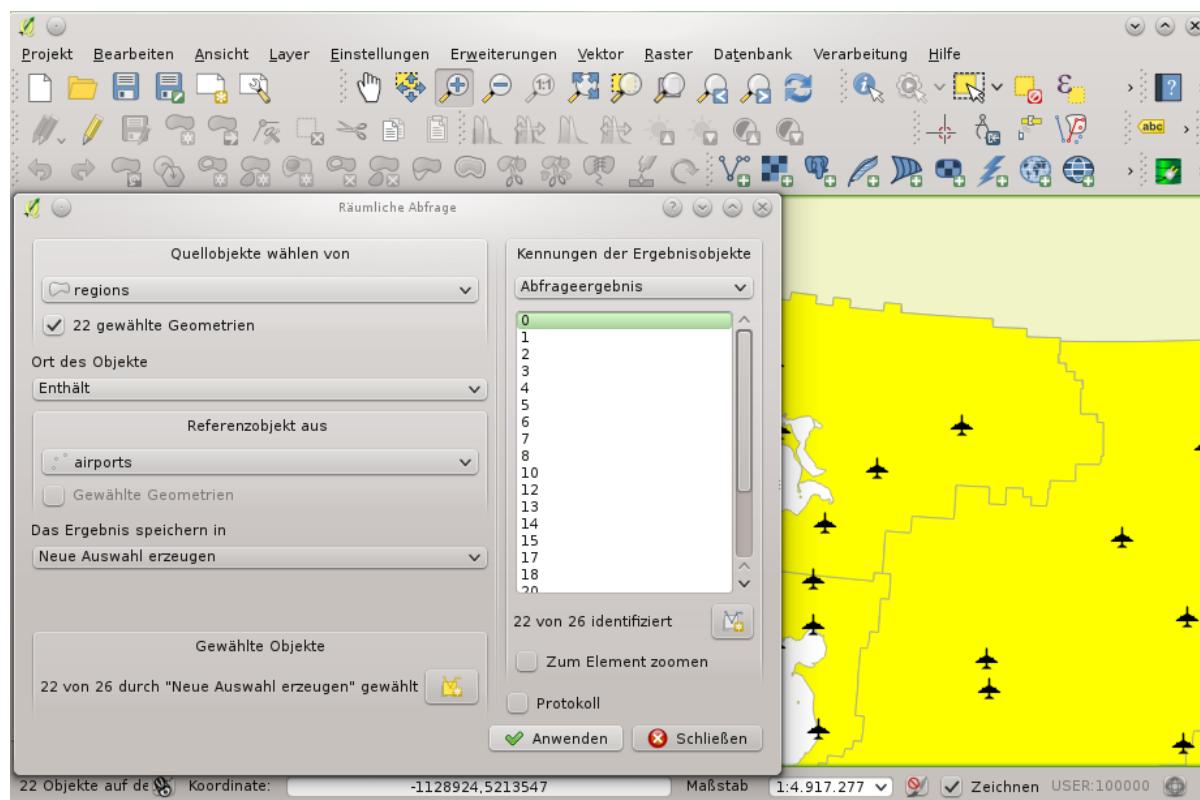



Abbildung 20.30: Räumliche Abfrage Analyse - regions enthalten airports 🐧

öffnen Sie den Plugin Manager aus dem *Erweiterungen* Menü, wählen das Plugin SPIT im *Installiert* Menü aus, aktivieren Sie das Kontrollkästchen neben  *SPIT* und drücken auf **[OK]**.

Um eine Shapedatei zu importieren verwenden Sie *Datenbank* → *Spit* → *Shapedateien nach PostgreSQL importieren* aus der Menüleiste um den *SPIT - Shapedatei nach PostGIS-Importwerkzeug* Dialog zu öffnen. Wählen Sie die PostGIS Datenbank, mit der Sie sich verbinden wollen, aus und klicken Sie auf **[Verbinden]**. Wenn Sie wollen können Sie einige Importoptionen definieren oder verändern. Jetzt können Sie eine oder mehrere Dateien zur Warteschlange hinzufügen indem Sie auf den **[Hinzufügen]** Knopf klicken. Um die Dateien zu prozessieren klicken Sie auf den **[OK]** Knopf. Der Fortschritt des Imports genauso wie Fehler/Warnungen werden angezeigt sowie eine Shapedatei prozessiert ist. .

## 20.19 SQL-Anywhere Plugin

SQL-Anywhere ist ein proprietäres relationales Datenbank-Managementsystem (RDBMS) von Sybase. SQL-Anywhere umfasst Unterstützung für räumliche Daten einschließlich OGC, Shape-Dateien etc. sowie eingebaute Funktionen für den Export nach KML, GML und ins SVG-Format.

Das  SQL-Anywhere Plugin erlaubt die Anbindung an räumliche SQL Anywhere Datenbanken. Der *SQL-Anywhere-Layer hinzufügen* Dialog ist in der Funktionalität ähnlich den Dialogen für das Laden von PostGIS oder SpatialLite Layern. .

## 20.20 Topologieprüfung Erweiterung

Topologie beschreibt die Beziehungen zwischen Punkten, Linien und Polygonen, die die Objekte eines Geografischen Gebiets repräsentieren. Mit dem Topologie-Prüfung Plugin können Sie Ihre Vektordateien anschauen und



Abbildung 20.31: Verwenden Sie die SPIT-Erweiterung um Shapedateien nach PostGIS zu importieren 🐧

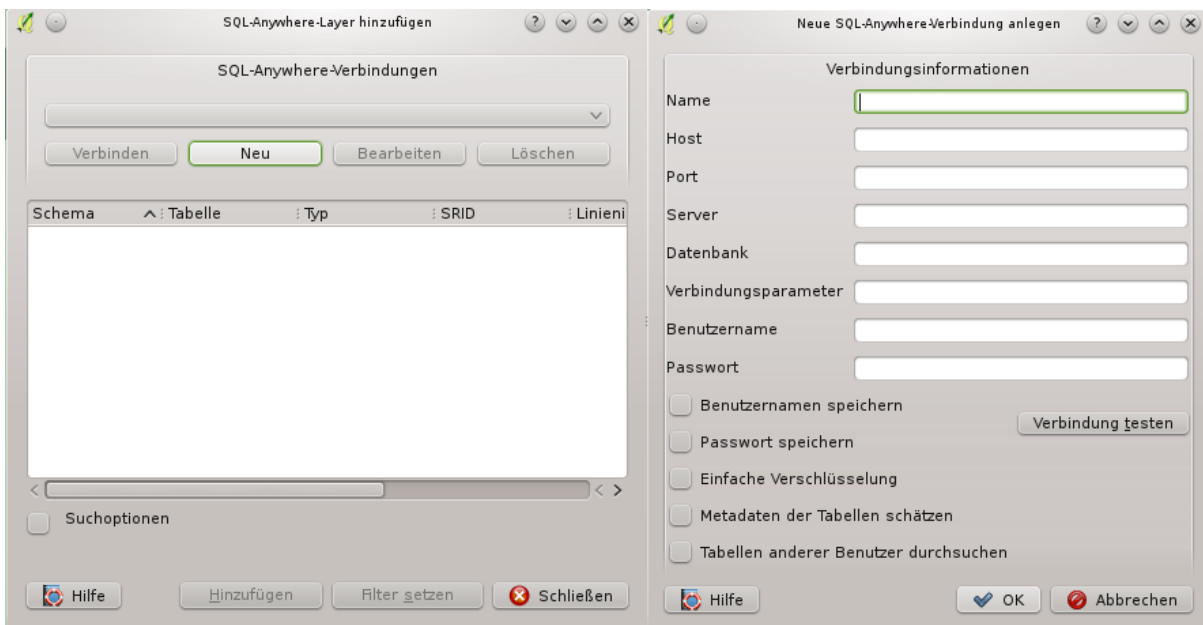


Abbildung 20.32: Der SQL-Anywhere Dialog (KDE) 🐧

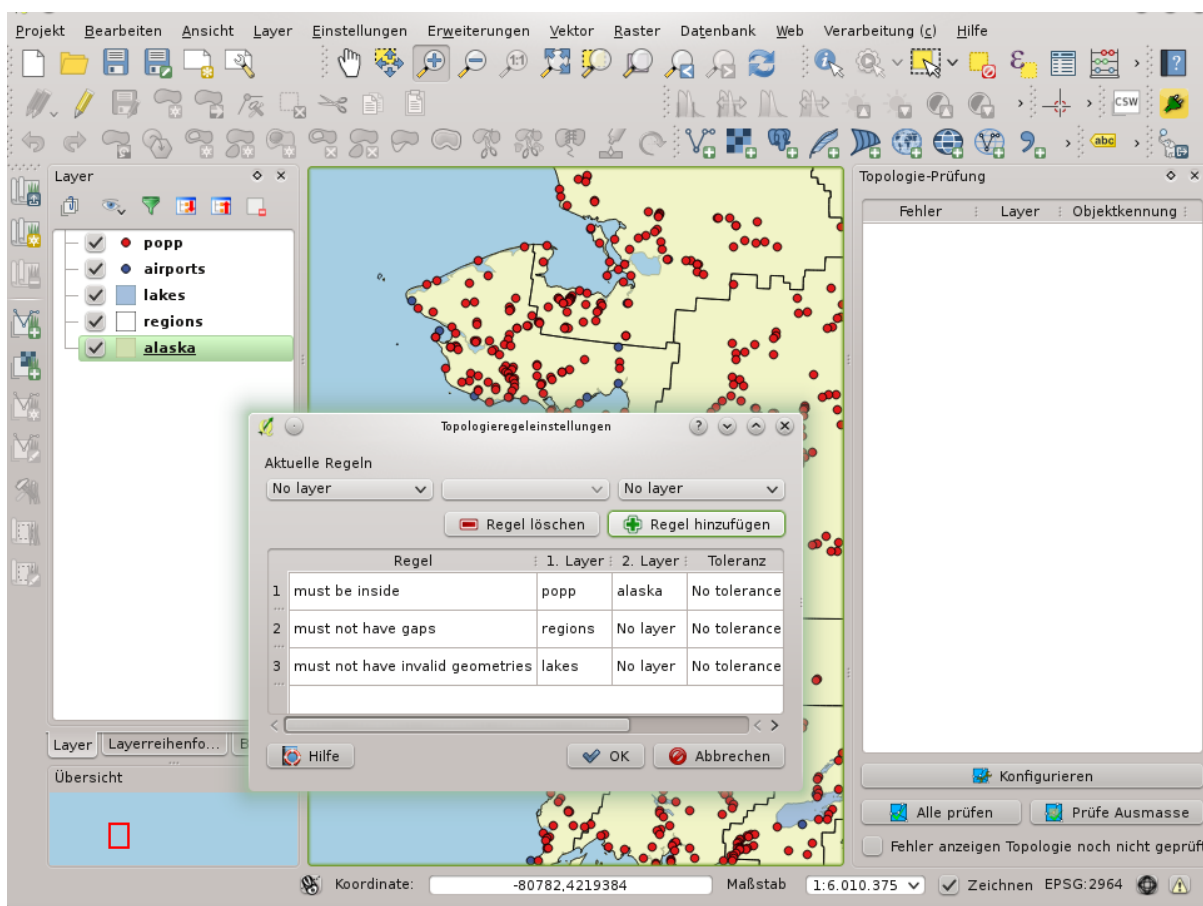


Abbildung 20.33: Das Topologie-Prüfung Plugin



die Topologie mit mehreren Topologieregeln überprüfen. Diese Regeln überprüfen mit räumlichen Beziehungen ob sich Ihre Objekte 'Gleichen', 'Enthalten', 'Abdecken', 'Abgedeckt werden von', 'Kreuzen', 'Disjunkt' sind, 'Überschneiden', 'Überlappen', 'Berühren' oder 'Innerhalb' voneinander sind. Es hängt von Ihren individuellen Fragestellungen ab welche Topologieregel Sie auf Ihre Vektordaten anwenden (z.B. werden Sie normalerweise keine Overshoots in Linienlayer akzeptieren, aber wenn Sie Sackgassen darstellen werden Sie sie nicht aus Ihrem Vektorlayer entfernen).

QGIS besitzt eine eingebaute Bearbeitungsfunktion, die toll für das Erstellen neuer Objekte ohne Fehler ist. Aber Datenfehler und benutzergemachte Fehler sind schwierig zu finden. Diese Erweiterung hilft Ihnen solche Fehler anhand einer Liste von Regeln zu finden.

Es ist sehr einfach Topologieregeln mit dem Topologie-Prüfung Plugin zu erstellen.

Für **Punktlayer** stehen die folgenden Regeln zu Verfügung:

- **Must be covered by:** Hier können Sie einen Vektorlayer aus Ihrem Projekt auswählen. Punkte die nicht vom vorgegebenen Vektorlayer abgedeckt sind erscheinen im 'Fehler' Feld.
- **Must be covered by endpoints of:** Hier können Sie einen Linienlayer aus Ihrem Projekt auswählen.
- **Must be inside:** Hier können Sie einen Polygonlayer aus Ihrem Projekt auswählen. Die Punkte müssen innerhalb eines Polygons sein. Andernfalls schreibt QGIS einen 'Fehler' für den Punkt.
- **Must not have duplicates:** Wann immer ein Punkt zweifach oder mehr repräsentiert wird erscheint dies im 'Fehler' Feld.
- **Must not have invalid geometries:** Überprüft ob die Geometrien gültig sind.
- **Must not have multi-part-geometries:** Alle Multi-Part Punkte werden in das 'Fehler' Feld geschrieben.

Für **Linienlayer** stehen die folgenden Regeln zur Verfügung:


- **End points must be covered by:** Hier können Sie einen Punktlayer aus Ihrem Projekt auswählen.
- **Must not have dangles:** Dies zeigt die Overshoots in Ihrem Linienlayer.
- **Must not have duplicates:** Wann immer ein Objekt zweimal oder mehr repräsentiert wird erscheint dies im 'Fehler' Feld.
- **Must not have invalid geometries:** Überprüft ob die Geometrien gültig sind.
- **Must not have multi-part geometries:** Manchmal ist eine Geometrie eigentlich eine Sammlung von einfachen (single-part) Geometrien. Solch eine Geometrie wird Multi-Part Geometrie genannt. Wenn es nur einen Typ von einfacher Geometrie enthält nennen wir dies Multi-Point, Multi-Linestring oder Multi-Polygon. Alle Multi-Part Linien werden in das 'Fehler' Feld geschrieben.
- **Must not have pseudos:** Ein Liniengeometrie-Endpunkt sollte mit den Endpunkten von zwei anderen Geometrien verbunden sein. Wenn der Endpunkt nur mit einem anderen Endpunkt verbunden ist, wird der Endpunkt ein Pseudonode genannt.

Für **Polygonlayer** stehen die folgenden Regeln zu Verfügung:

- **Must contain:** Polygonlayer müssen mindestens eine Punktgeometrie von einem zweiten Layer enthalten.
- **Must not have duplicates:** Polygonlayer aus dem gleichen Layer dürfen keine identischen Geometrien haben. Wann immer ein Polygonobjekt zweimal oder mehr repräsentiert wird, erscheint dies im 'Fehler' Feld.
- **Must not have gaps:** Aneinanderhängende Polygone dürfen keine Lücken zwischeneinander bilden. Als Beispiel können hier administrative Grenzen genannt werden (US Staaten Polygone haben keine Lücken zwischeneinander ...).
- **Must not have invalid geometries:** Überprüft ob die Geometrien gültig sind. Einige der Regeln, die eine gültige Geometrie definieren sind:
  - Polygonringe müssen geschlossen sein.
  - Ringe, die Lücken definieren, sollten innerhalb von Ringen sein, die äußere Grenzen definieren.
  - Ringe können sich nicht schneiden (sie dürfen sich weder berühren noch kreuzen).

- Ringe dürfen keine anderen Ringe berühren, es sei denn an einem Punkt.
- **Must not have multi-part geometries:** Manchmal ist eine Geometrie eigentlich eine Sammlung von einfachen (single-part) Geometrien. Solch eine Geometrie wird Multi-Part Geometrie genannt. Wenn es nur einen Typ von einfacher Geometrie enthält nennen wir dies Multi-Point, Multi-Linestring oder Multi-Polygon.
- **Must not overlap:** Aneinanderhängende Polygone sollten keine gemeinsame Fläche bilden.
- **Must not overlap with:** Aneinanderhängende Polygone aus einem Layer sollten keine gemeinsame Fläche mit Polygonen eines anderen Layers bilden.

## 20.21 Zonenstatistikerweiterung

Mit der  Zonenstatistikerweiterung können Sie das Ergebnis einer thematischen Klassifikation analysieren. Es ermöglicht, verschiedene, descriptive statistische Werte von Rasterpixeln innerhalb von Flächen eines Vektorlayers zu berechnen (see [figure\\_zonal\\_statistics](#)). Sie können Summe, Mittelwert und die Anzahl der Pixel innerhalb der einzelnen Flächen berechnen. Das Plugin erstellt für jeden zu berechnenden Wert eine Attributspalte, die man mit einem Präfix definieren kann.

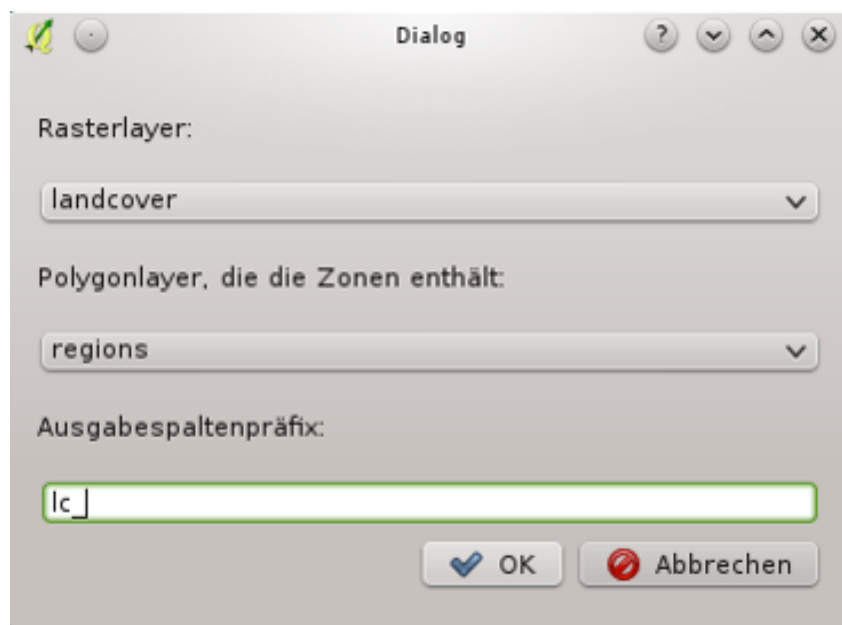


Abbildung 20.34: Zonenstatistik Dialog (KDE) 

---

## Hilfe und Support

---

### 21.1 Mailinglisten

QGIS wird immer weiterentwickelt und wird nicht immer so funktionieren wie Sie es erwarten. Der beste Weg Hilfe zu bekommen ist sich in die qgis-users Mailingliste einzutragen. Ihre Fragen werden ein breiteres Publikum erreichen und die Antworten werden anderen nützen.

#### 21.1.1 qgis-users

Diese Mailingliste wird dazu benutzt um QGIS allgemein zu diskutieren und auch für gezielte Fragen zur Installation und Benutzung. Sie können sich in die qgis-users Mailingliste eintragen indem Sie die folgende URL besuchen: <http://lists.osgeo.org/mailman/listinfo/qgis-user>

#### 21.1.2 fossgis-talk-liste

Für das deutschsprachige Publikum bietet der deutsche FOSSGIS e.V. die fossgis-talk-liste Mailingliste. Diese Mailingliste wird dazu benutzt um Open Source GIS im allgemeinen, darunter QGIS, zu diskutieren. Sie können sich in die fossgis-talk-liste Mailingliste eintragen wenn Sie die folgende URL wählen: <https://lists.fossGIS.de/mailman/listinfo/fossGIS-talk-liste>

#### 21.1.3 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.osgeo.org/mailman/listinfo/qgis-developer>

#### 21.1.4 qgis-commit

Jedes mal wenn ein Commit im QGIS Code Repository gemacht wurde wird eine Email an diese Liste gesendet. Wenn Sie über jede Änderung zum aktuellen Code Base aufgeklärt werden wollen, können Sie sich in diese Liste eintragen: <http://lists.osgeo.org/mailman/listinfo/qgis-commit>

#### 21.1.5 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.osgeo.org/mailman/listinfo/qgis-trac>

### 21.1.6 qgis-community-team

Diese Liste befasst sich mit Themen wie Dokumentation, Kontexthilfe, Benutzerhandbuch, Webseiten, Blog, Mailinglisten, Foren und Übersetzungen. Wenn Sie auch an dem Benutzerhandbuch arbeiten wollen ist diese Liste eine gute Möglichkeit um Fragen zu stellen. Sie können sich unter <http://lists.osgeo.org/mailman/listinfo/qgis-community-team> in die Liste eintragen.

### 21.1.7 qgis-release-team

Diese Liste befasst sich mit Themen wie der Veröffentlichung von QGIS , Binärpaketen für zahlreiche Betriebssysteme und der Ankündigung von neuen Erscheinungen an die ganze Welt. Sie können sich unter <http://lists.osgeo.org/mailman/listinfo/qgis-release-team> in die Liste eintragen.

### 21.1.8 qgis-tr

Diese Liste beschäftigt sich mit der Übersetzung der QGIS GUI und der Dokumentation. Sie können sich für diese Mailingliste eintragen unter der URL: <http://lists.osgeo.org/mailman/listinfo/qgis-tr>

### 21.1.9 qgis-edu

Diese Liste befasst sich mit der Umsetzung des QGIS Education Projektes. Wenn Sie mit an den QGIS Education Materialien arbeiten wollen können Sie hier mit Ihren Fragen anfangen. Sie können sich unter <http://lists.osgeo.org/mailman/listinfo/qgis-edu> in die Liste eintragen.

### 21.1.10 qgis-psc

Diese Liste wird dazu benutzt Themen zu diskutieren die den Lenkungsausschuß betreffen, also Fragen zum übergreifenden Management und zur Leitung. Sie können sich unter <http://lists.osgeo.org/mailman/listinfo/qgis-psc> in die Liste eintragen.

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.

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

Kommerzieller Support ist auch möglich. Schauen Sie dazu auf die Internetseite <http://www.qgis.org/de/kommerzieller-support.html>.

## 21.3 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://hub.qgis.org/projects/quantum-gis/issues> 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 `Feature` an.

Wenn Sie einen Fehler gefunden haben und selbst repariert haben können Sie diesen auch als Patch schicken. Dafür hat das schöne Redmine Ticketsystem unter <http://hub.qgis.org/wiki/quantum-gis/issues> wiederum einen Typ dafür. Kreuzen Sie das `Patch supplied` Kontrollkästchen an und hängen Sie Ihren Patch an bevor Sie Ihren Fehler einreichen. Einer der Entwickler wird sich den Patch anschauen und ihn auf QGIS anwenden. Bitte seien Sie nicht beunruhigt wenn Ihr Patch nicht gleich angewendet wird - die Entwickler sind dann mit anderen Aufgaben beschäftigt.

## 21.4 Blog

Die QGIS-Gemeinschaft stellt auch einen Weblog (BLOG) unter <http://planet.qgis.org/planet/> bereit, mit vielen interessanten Artikeln für Anwender und Entwickler zum Thema QGIS. Es handelt sich dabei um eine Sammlung verschiedener Blogs zum Thema QGIS. Sie sind natürlich herzlich eingeladen, ihren eigenen QGIS Blog zu integrieren !

## 21.5 Plugins

Die Internetseite <http://plugins.qgis.org> stellt das offizielle QGIS Plugin Webportal dar. Hier finden Sie eine Liste mit allen stabilen und experimentellen QGIS Plugins, die über das 'Official QGIS Plugin Repository' zur Verfügung stehen.

## 21.6 Wiki

Schließlich gibt es auch ein QGIS WIKI unter <http://hub.qgis.org/projects/quantum-gis/wiki> , 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!



## 22.1 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)

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 lizensieren - 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

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:

- (a) Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.
- (b) 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.
- (c) 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 ausdrückt, 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 auszugswweisen 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:
  - (a) 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,
  - (b) 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,
  - (c) 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 Verbreitungssystems 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 Urheberrechtinhaber, 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") 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.

### QGIS Qt Ausnahme für die GPL

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.

## 22.2 GNU Free Documentation License

Dies ist eine inoffizielle Übersetzung der GNU Free Documentation License (FDLv1.3) ins Deutsche. Sie wurde nicht von der Free Software Foundation veröffentlicht, und legt keine rechtsgültigen Bestimmungen zur Verteilung für Dokumentation fest, welche die GNU FDL verwendet - nur der englische Originaltext der GNU FDL gewährleistet dies. Dennoch hoffen wir, dass diese Übersetzung Deutsch sprechenden dazu verhilft, die GNU FDL besser zu verstehen

Copyright 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc

<http://fsf.org/>

Es ist jedermann gestattet, diese Lizenzurkunde zu vervielfältigen und unveränderte Kopien zu verbreiten; Änderungen sind jedoch nicht erlaubt.

### Einleitung

Der Zweck dieser Lizenz ist es, ein Handbuch, Fachbuch, oder ein anderes sachliches sowie nützliches Dokument 'frei' im Sinne von Freiheit anzufertigen: um jedermann die wirkliche Freiheit zuzusichern, es zu kopieren und neu zu verteilen, mit oder ohne Modifikation daran, entweder kommerziell oder nicht kommerziell. Zweitrangig erhält diese Lizenz für den Autor und Herausgeber die Möglichkeit aufrecht, Anerkennung für ihr Werk zu bekommen, während Sie nicht als verantwortlich betrachtet werden fürvorgenommene Modifikationen anderer.

Diese Lizenz ist eine Art 'Copyleft', was bedeutet, dass abgeleitete Werke des Dokuments ihrerseits im selben Sinne frei sein müssen. Sie ergänzt die GNU General Public License, welche eine Copyleft-Lizenz darstellt, die für freie Software vorgesehen ist.

Wir haben diese Lizenz zur Verwendung für Handbücher von freier Software konzipiert, da freie Software freie Dokumentation benötigt: Ein freies Programm sollte mit Handbüchern erhältlich sein, welche dieselben Freiheiten zur Verfügung stellen wie es die Software tut. Doch diese Lizenz ist nicht auf Software-Handbücher beschränkt; sie kann für jedes textliche Werk verwendet werden, unabhängig vom Thema oder ob es als ein gedrucktes Buch veröffentlicht wird. Wir empfehlen diese Lizenz vorwiegend für Werke, deren Einsatzzweck Anleitung oder Referenz ist.

### 1. Anwendbarkeit und Definitionen

Diese Lizenz trifft auf jedes Handbuch oder sonstiges Werk zu, in beliebiger Form, das einen vom Urheberrechtsinhaber untergebrachten Hinweis mit den Worten enthält, dass es unter den Bestimmungen dieser Lizenz verteilt werden kann. Solch ein Hinweis gewährt eine weltweite, vergütungsfreie Lizenz von unbefristeter Dauer, um dieses Werk unter den hier festgelegten Bedingungen zu verwenden. Das 'Dokument', nachstehend, bezieht sich auf jedes derartige Handbuch oder Werk. Jedes Mitglied der Öffentlichkeit ist ein Lizenznehmer und wird als 'Sie' angesprochen. Sie akzeptieren die Lizenz, wenn Sie die Werke kopieren, modifizieren oder verteilen, was gewissermaßen unter dem Urheberrechtsgesetz die Erlaubnis erfordert.

Eine **modifizierte Version** des Dokumentes bedeutet, dass jedes Werk das Dokument selbst oder einen Teil davon beinhaltet, entweder unverändert kopiert, oder mit Modifikationen und/oder übersetzt in eine andere Sprache.

Ein **untergeordneter Abschnitt** ist ein benannter Anhang oder ein Titelei-Abschnitt des Dokuments, der sich ausschließlich mit dem Verhältnis der Herausgeber oder Autoren des Dokuments zum Gesamthema des Dokuments befasst (oder damit in Verbindung stehende Bewandnisse), und nichts beinhaltet was direkt innerhalb des Gesamthemas fallen könnte. (Wenn das Dokument zu einem Fachbuch über Mathematik gehört, kann ein untergeordneter Abschnitt folglich nichts Mathematisches erläutern.) Die Beziehung könnte ein Anliegen mit historischer Verbindung zum Thema oder ähnlicher Angelegenheiten, oder bezüglich ihrer rechtlichen, kommerziellen, philosophischen, ethischen oder politischen Position sein.

Die ‘unveränderlichen Abschnitte’ sind bestimmte untergeordnete Abschnitte, deren Titel zum Beispiel in dem Hinweis, der besagt, dass das Dokument unter dieser Lizenz freigegeben ist, als jene unveränderlichen Abschnitte gekennzeichnet sind. Wenn ein Abschnitt nicht zur obigen Definition von untergeordnet passt, dann ist es nicht erlaubt ihn als unveränderlich zu kennzeichnen. Das Dokument kann null unveränderliche Abschnitte enthalten. Wenn das Dokument keine unveränderlichen Abschnitte kennzeichnet, dann gibt es keine.

Die ‘Umschlagtexte’ sind bestimmte kurze Textpassagen die als vordere Umschlagtexte oder hintere Umschlagtexte in dem Hinweis, der besagt, dass das Dokument unter dieser Lizenz freigegeben ist, verzeichnet sind. Ein vorderer Umschlagtext darf höchstens 5 Wörter lang sein, und ein hinterer Umschlagtext darf höchstens 25 Wörter lang sein.

Eine ‘transparente’ Kopie des Dokumentes bedeutet eine maschinenlesbare Kopie, in einem ansehnlichen Format, dessen Spezifikation für die Allgemeinheit verfügbar ist, welches geeignet ist das Dokument unkompliziert mit allgemeinen Texteditoren oder (für aus Pixeln bestehende Bilder) allgemeinen Malprogrammen oder (für Zeichnungen) irgendeinem weit verbreiteten Zeicheneditor zu überarbeiten, und das geeignet ist zur Eingabe in Textformatierer oder zur automatischen Übersetzung in eine Variante von geeigneten Formaten zur Eingabe in Textformatierer. Eine Kopie, erstellt in einem ansonsten transparenten Dateiformat, dessen Auszeichnung oder fehlende Auszeichnung derart ausgestaltet wurde, um nachträgliche Modifikation durch Leser zu behindern oder zu verhindern, ist nicht transparent. Ein Bildformat ist nicht transparent, wenn es für irgendeine beträchtliche Menge von Text verwendet wird. Eine Kopie, die nicht ‘transparent’ ist, wird ‘undurchlässig’ genannt.

Beispiele von geeigneten Formaten für transparente Kopien beinhalten einfachen ASCII ohne Auszeichnung, Textinfo Eingabeformat, LaTeX Eingabeformat, SGML oder XML unter Verwendung einer öffentlich zugänglichen DTD, und standardkonformes einfaches HTML, PostScript oder PDF, vorgesehen für humane Modifikation. Beispiele für transparente Bildformate beinhalten PNG, XCF und JPG. Undurchlässige Formate beinhalten proprietäre Formate die nur mit proprietären Textverarbeitungssystemen gelesen und bearbeitet werden können, SGML oder XML für welche die DTD und/oder Bearbeitungswerkzeuge nicht allgemein verfügbar sind, und das maschinengenerierte HTML, PostScript oder PDF, erzeugt mit irgendwelchen Textverarbeitungssystemen, nur für Ausgabezwecke.

Das ‘Titelblatt’ bedeutet, für ein gedrucktes Buch, das Titelblatt an sich zzgl. solcher nachfolgenden Seiten die notwendig sind, die Lesbarkeit des Materials beizubehalten, wie von dieser Lizenz erfordert, um im Titelblatt zu erscheinen. Für Werke in Formaten, welche kein Titelblatt als solches haben, bedeutet ‘Titelblatt’ der Text nahe dem bedeutendsten Auftreten des Titels dieses Werkes, ausgehend vom Anfang des Textkörpers.

Der ‘Herausgeber’ ist jede Person oder Instanz, welche Kopien des Dokuments an die Öffentlichkeit verteilt.

Ein Abschnitt **Mit dem Titel XYZ** bedeutet eine benannte Untereinheit des Dokuments, dessen Titel entweder genau XYZ ist, oder XYZ in runden Klammern, gefolgt von Text, welcher XYZ in eine andere Sprache übersetzt. (Hier steht XYZ für einen spezifischen Abschnittsnamen, weiter unten erwähnt, wie zum Beispiel ‘Danksagungen’, ‘Widmungen’, ‘Befürwortungen’ oder ‘Verlauf’.) Den ‘Titel’ eines solchen Abschnitts ‘zu erhalten’, wenn Sie das Dokument modifizieren, bedeutet, dass dieser ein Abschnitt ‘Mit dem Titel XYZ’ gemäß seiner Definition bleibt.

Das Dokument kann neben dem Hinweis, der festlegt, dass diese Lizenz auf das Dokument zutrifft, Garantie-Ausschlussklauseln beinhalten. Diese Garantie-Ausschlussklauseln werden als aufgenommen betrachtet, durch Verweis in dieser Lizenz, aber nur was die Ausschließung von Garantien betrifft: Jede andere Auswirkung, die diese Garantie-Ausschlussklauseln haben können, ist ungültig und hat keine Wirkung auf die Bedeutung dieser Lizenz.

## 2. Unveränderte Vervielfältigung

Sie dürfen das Dokument in beliebiger Form kopieren und verteilen, entweder kommerziell oder nicht kommerziell, vorausgesetzt, dass diese Lizenz, die Urheberrechtshinweise und der Lizenzhinweis mit den Worten, dass diese Lizenz auf das Dokument zutrifft, in allen Kopien wiedergegeben werden, und dass Sie keine anderen Bedingungen, welcher Art auch immer, zu jenen dieser Lizenz hinzufügen. Sie dürfen keine technischen Maßnahmen anwenden die das Lesen oder weitere Vervielfältigung von den Kopien, die Sie erstellen oder verteilen, behindern oder kontrollieren. Allerdings können Sie Bezahlung im Austausch für Kopien entgegennehmen. Wenn Sie eine ausreichend große Anzahl von Kopien verteilen, müssen Sie außerdem die Bedingungen in Abschnitt 3 beachten.

Sie dürfen außerdem Kopien verleihen, unter den selben, oben angegebenen Bedingungen, und Sie dürfen Kopien öffentlich auslegen.

### 3. Vervielfältigung grosser Mengen

Wenn Sie, mehr als 100 umfassend, gedruckte Kopien eines Dokuments veröffentlichen (oder Kopien in Medien, die üblicherweise gedruckte Umschläge haben) und der Lizenzhinweis des Dokuments Umschlagtexte erfordert, müssen Sie die Kopien in Umschläge beifügen, welche eindeutig und leserlich alle diese Umschlagtexte tragen: vordere Umschlagtexte auf vordere Umschläge, und hintere Umschlagtexte auf hintere Umschläge. Beide Umschläge müssen, ebenso eindeutig und leserlich, Sie, als den Herausgeber dieser Kopien, identifizieren. Der vordere Umschlag muss den vollständigen Titel, mit allen Wörtern des Titels, in gleicher Weise auffallend und leicht erkennbar darstellen. Sie dürfen darüber hinaus sonstiges Material auf die Umschläge anbringen. Vervielfältigung mit Änderungen, begrenzt auf die Umschläge, sofern sie den Titel des Dokuments erhalten und diese Bedingungen erfüllen, können ansonsten als unveränderte Vervielfältigung behandelt werden.

Wenn die erforderlichen Texte für beide Umschläge zu umfangreich sind, um lesbar zu passen, sollten Sie die ersten verzeichneten (so viele, um angemessen zu passen) auf den aktuellen Umschlag setzen, und den Rest auf den nachfolgenden Seiten fortsetzen.

Wenn Sie, mehr als 100 umfassend, undurchlässige Kopien des Dokuments veröffentlichen oder verteilen, müssen Sie entweder eine maschinenlesbare transparente Kopie zusammen mit jeder undurchlässigen Kopie einbeziehen, oder in bzw. mit jeder undurchlässigen Kopie eine Computer-Netzwerkadresse angeben, von welcher die allgemeine netzwerknutzende Öffentlichkeit, unter Verwendung von Netzwerkprotokollen öffentlicher Standards, Zugang hat, um eine vollständig transparente Kopie, frei von hinzugefügtem Material, herunterzuladen. Falls Sie die letzte Option verwenden, müssen Sie angemessen überlegte Schritte unternehmen, wenn Sie mit der Verteilung von undurchlässigen Kopien in großen Mengen beginnen, um sicherzustellen, dass diese transparente Kopie unter der angegebenen Adresse auf diese Weise zugänglich bleibt, bis mindestens ein Jahr nachdem Sie zum letzten Mal eine undurchlässige Kopie (unmittelbar oder durch Ihre Vertreter oder Einzelhändler) in dieser Ausgabe an die Öffentlichkeit verteilen.

Es ist erwünscht, aber nicht erforderlich, dass Sie in Kontakt mit den Autoren des Dokuments treten, kurz bevor Sie irgendeine große Anzahl von Kopien neu verteilen, um ihnen die Möglichkeit zu geben, Sie mit einer aktualisierten Version des Dokuments zu versorgen.

### 4. Modifikationen

Sie dürfen eine modifizierte Version des Dokuments unter den oben erwähnten Bedingungen der Abschnitte 2 und 3 kopieren und verteilen, vorausgesetzt, dass Sie die modifizierte Version unter genau dieser Lizenz freigeben, mit der modifizierten Version wird die Rolle des Dokuments besetzt, und somit der Lizenzierung von Verteilung und Modifikation der modifizierten Version, für jeden der eine Kopie davon besitzt. Zusätzlich müssen Sie diese Sachen in der modifizierten Version erledigen:

1. Verwenden Sie auf dem Titelblatt (und gegebenenfalls auf den Umschlägen) einen Titel der sich von dem des Dokuments unterscheidet, und von jenen der vorhergehenden Versionen (die, wenn es irgendwelche gab, in dem Verlaufs-Abschnitt des Dokuments verzeichnet sein sollten). Sie dürfen denselben Titel wie den einer vorhergehenden Version verwenden, wenn der ursprüngliche Herausgeber dieser Version die Erlaubnis gibt.
2. Verzeichnen Sie auf dem Titelblatt, als Autoren, eine oder mehrere Personen oder Organe, verantwortlich für die Autorschaft der Modifikationen in der modifizierten Version, zusammen mit mindestens fünf der Hauptautoren des Dokuments (alle seine Hauptautoren, wenn es weniger als fünf hat), es sei denn, dass sie Sie von dieser Anforderung befreien.
3. Geben Sie auf dem Titelblatt den Namen des Herausgebers der modifizierten Version als den des Herausgebers an.

4. Erhalten Sie alle Urheberrechtshinweise des Dokumentes.
5. Fügen Sie einen entsprechenden Urheberrechtshinweis für Ihre Modifikationen, angrenzend zu den anderen Urheberrechtshinweisen, hinzu.
6. Nehmen Sie, direkt nach den Urheberrechtshinweisen, einen Lizenzhinweis auf, der die öffentliche Erlaubnis gibt, die modifizierte Version unter den Bestimmungen dieser Lizenz zu verwenden, in der Form, wie weiter unten im Anhang gezeigt.
7. Erhalten Sie in diesem Lizenzhinweis die vollständigen Listen der unveränderlichen Abschnitte und erforderlichen Umschlagtexte, aufgeführt in dem Lizenzhinweis des Dokuments.
8. Nehmen Sie eine ungeänderte Kopie dieser Lizenz auf.
9. Erhalten Sie den Abschnitt mit dem Titel 'Verlauf', erhalten Sie seinen Titel und fügen Sie ihm einen Punkt hinzu, der mindestens den Titel, das Jahr, neue Autoren und Herausgeber der modifizierte Version angibt, wie auf dem Titelblatt aufgeführt. Wenn es keinen Abschnitt 'Verlauf' in dem Dokument gibt, erzeugen Sie einen, der den Titel, das Jahr, die Autoren und Herausgeber des Dokuments angibt, wie auf seinem Titelblatt aufgeführt, dann fügen Sie einen Punkt hinzu, der die modifizierte Version beschreibt, wie in dem vorhergehenden Satz angegeben.
10. Erhalten Sie gegebenenfalls die Netzwerkadresse, aufgeführt in dem Dokument, für den öffentlichen Zugang zu einer transparenten Kopie des Dokuments, und ebenso die in dem Dokument aufgeführten Netzwerkadressen, für vorhergehende Versionen, auf dem es beruht. Diese können in den Abschnitt 'Verlauf' untergebracht werden. Sie dürfen eine Netzwerkadresse für ein Werk auslassen, das mindestens vier Jahre vor dem Dokument selbst veröffentlicht wurde, oder der ursprüngliche Herausgeber der Version, auf die es sich bezieht, die Erlaubnis gibt.
11. Für jeden Abschnitt mit dem Titel 'Danksagungen' oder 'Widmungen', erhalten Sie den Titel des Abschnitts, und erhalten Sie in dem Abschnitt vollständig den wesentlichen Gehalt und Umfangston der, von jeglichen Mitwirkenden darin aufgeführten, Danksagungen und/oder Widmungen.
12. Erhalten Sie alle unveränderlichen Abschnitte des Dokuments, ungeändert in ihrem Text und ihren Titeln. Abschnittsnummern oder Entsprechendes werden nicht als Teil der Abschnittstitel betrachtet.
13. Löschen Sie jeden Abschnitt mit dem Titel 'Befürwortungen'. Solch ein Abschnitt darf nicht in die modifizierte Version aufgenommen werden.
14. Betiteln Sie keinen Titel eines vorhandenen Abschnitts neu, der mit 'Befürwortung' betitelt ist oder in Konflikt zum Titel irgendeines unveränderlichen Abschnitts steht.
15. Erhalten Sie jegliche Garantie-Ausschlussklauseln.

Wenn die modifizierte Version neue Titelei-Abschnitte oder Anhänge beinhaltet, die als untergeordnete Abschnitte qualifiziert sind und kein kopiertes Material aus dem Dokument enthalten, können Sie Ihrer Wahl nach einige oder alle diese Abschnitte als unveränderlich bestimmen. Um dies zu tun, fügen Sie ihre Titel zur Liste der unveränderlichen Abschnitte in den Lizenzhinweis der modifizierten Version hinzu. Diese Titel müssen sich von allen anderen Abschnittstitel unterscheiden.

Sie können einen Abschnitt mit dem Titel 'Befürwortungen' hinzufügen, vorausgesetzt, dass dieser nichts enthält als nur Befürwortungen Ihrer modifizierten Version durch verschiedene Parteien Beispiel Aussagen der Begutachtung oder dass der Text von einer Organisation als maßgebliche Definition eines Standards anerkannt worden ist.

Sie können eine Textpassage von bis zu fünf Wörtern als einen vorderen Umschlagtext, und eine Textpassage von bis zu 25 Wörtern als hinteren Umschlagtext in der modifizierten Version hinzufügen. Nur eine Textpassage des vorderen Umschlagtextes und eine des hinteren Umschlagtextes kann von (oder durch, von ihr, angefertigte Zusammenstellung) irgendeiner Person hinzugefügt werden. Wenn das Dokument bereits einen Umschlagtext für denselben Umschlag beinhaltet, zuvor von Ihnen hinzugefügt oder durch Zusammenstellung, angefertigt von derselben Person, in dessen Namen Sie handeln, können Sie keinen weiteren hinzufügen; aber Sie dürfen den alten ersetzen, mit ausdrücklicher Erlaubnis des vorherigen Herausgebers, welcher den alten hinzufügte.

Der/die Autor(en) und Herausgeber des Dokuments geben durch diese Lizenz für keine modifizierte Version die Erlaubnis ihre Namen für Werbung zu verwenden oder Befürwortung zu behaupten oder anzudeuten.

### 5. Dokumente kombinieren

Sie dürfen das Dokument mit anderen, unter dieser Lizenz freigegeben, Dokumenten kombinieren, unter den Bestimmungen, definiert in Abschnitt 4 für modifizierte Versionen, vorausgesetzt, dass Sie in die Kombination alle unveränderlichen Abschnitte aller Originaldokumente, nicht modifiziert, einbeziehen und sie alle als unveränderliche Abschnitte Ihres kombinierten Werkes in dessen Lizenzhinweis verzeichnen, und dass Sie alle ihre Garantie-Ausschlussklauseln erhalten.

Das kombinierte Werk muss nur eine Kopie dieser Lizenz enthalten, und mehrfach identische unveränderliche Abschnitte können gegen eine einzelne Kopie ausgetauscht werden. Wenn es mehrfach identische unveränderliche Abschnitte mit demselben Namen, aber unterschiedlichem Inhalt gibt, erzeugen Sie den Titel eines jeden solchen Abschnitts an seinem Ende eindeutig durch Hinzufügen, in runden Klammern, des Namens, falls bekannt, des ursprünglichen Autors oder Herausgebers dieses Abschnittes, oder andernfalls einer eindeutigen Nummer. Nehmen Sie dieselbe Anpassung bei den Abschnittstiteln in der Liste von unveränderlichen Abschnitten in dem Lizenzhinweis des kombinierten Werkes vor.

In der Kombination müssen Sie jegliche Abschnitte mit dem Titel 'Verlauf' in den verschiedenen Originaldokumenten kombinieren und einen Abschnitt mit dem Titel 'Verlauf' bilden; ebenso kombinieren Sie jegliche Abschnitte mit dem Titel 'Danksagungen', und jegliche Abschnitte mit dem Titel 'Widmungen'. Sie müssen alle Abschnitte mit dem Titel 'Befürwortungen' löschen.

## 6. Sammlungen von Dokumenten

Sie dürfen eine Sammlung erzeugen, bestehend aus dem Dokument und anderen Dokumenten, freigegeben unter dieser Lizenz, und die Einzel-Kopien dieser Lizenz in den verschiedenen Dokumenten gegen eine einzelne Kopie, die in die Sammlung aufgenommen wird, austauschen, vorausgesetzt, dass Sie die Regeln dieser Lizenz für unveränderte Vervielfältigung jedes Dokuments in jeder anderen Hinsicht befolgen.

Sie dürfen ein einzelnes Dokument aus solch einer Sammlung entnehmen, und es individuell unter dieser Lizenz verteilen, vorausgesetzt, dass Sie eine Kopie dieser Lizenz in das extrahierte Dokument einfügen, und diese Lizenz in jeder anderen Hinsicht, bezüglich der unveränderten Vervielfältigung dieses Dokuments, befolgen.

## 7. Gruppierung mit unabhängigen Werken

Eine Zusammentragung des Dokuments oder seiner Ableitungen mit anderen gesonderten und unabhängigen Dokumenten oder Werken, in oder zu einem Band auf einem Speicher- oder Verteilungsmedium, wird eine 'Gruppierung' genannt, wenn das Urheberrecht, das sich aus der Zusammentragung ergibt, nicht verwendet wird, um die gültigen Rechte der Benutzer dieser Zusammentragungen darüber hinaus, was die Einzel-Werke gestatten, zu beschränken. Wenn das Dokument in eine Gruppierung aufgenommen ist, trifft diese Lizenz nicht auf die anderen Werke in der Gruppierung zu, welche nicht selbst abgeleitete Werke des Dokuments sind.

Wenn die Anforderung für Umschlagtext von Abschnitt 3 auf diese Kopien des Dokuments anwendbar ist, wenn außerdem das Dokument weniger als eine Hälfte der gesamten Gruppierung darstellt, kann der Umschlagtext des Dokuments auf Umschläge, die das Dokument innerhalb der Gruppierung einklammern, untergebracht werden, oder der elektronischen Entsprechung von Umschlägen, wenn das Dokument in elektronischer Form vorliegt. Andernfalls müssen sie auf gedruckten Umschlägen vorkommen, welche die vollständige Gruppierung einklammern.

## 8. Übersetzung

Übersetzung wird als eine Art von Modifikation betrachtet, also dürfen Sie Übersetzungen unter den Bestimmungen von Abschnitt 4 verteilen. Das Austauschen unveränderlicher Abschnitte mit Übersetzungen erfordert besondere Erlaubnis von ihren Urheberrechtshabern, aber Sie können Übersetzungen von einigen oder allen unveränderlichen Abschnitten aufnehmen, zusätzlich zu den Originalversionen dieser unveränderlichen Abschnitte. Sie können eine Übersetzung dieser Lizenz aufnehmen, und alle Lizenzhinweise in dem Dokument, und jegliche Garantie-Ausschlussklauseln, vorausgesetzt, dass Sie außerdem die englische Originalversion dieser Lizenz und die Originalversionen jener Hinweise und Ausschlussklauseln aufnehmen. Im Falle eines Widerspruchs zwischen der Übersetzung und der Originalversion dieser Lizenz oder eines Hinweises oder einer Ausschlussklausel, wird sich die Originalversion durchsetzen.

Wenn ein Abschnitt in dem Dokument mit 'Danksagungen', 'Widmungen' oder 'Verlauf' betitelt ist, wird die Anforderung (Abschnitt 4), seinen Titel (Abschnitt 1) zu erhalten, normalerweise die Änderung des tatsächlichen Titels erfordern.

## 9. Schlussbestimmung

Sie dürfen das Dokument nicht kopieren, modifizieren, unterlizenzieren oder verteilen, außer, als ausdrücklich

unter dieser Lizenz zur Verfügung gestellt. Jeder andere Versuch es zu kopieren, zu modifizieren, zu unterlizenzieren oder zu verteilen ist ungültig, und wird automatisch Ihre Rechte unter dieser Lizenz aufkündigen.

Jedoch, wenn Sie jeglichen Verstoß gegen diese Lizenz beenden, dann ist ihre Lizenz eines bestimmten Urheberrechtsinhabers wieder (a) vorläufig, sofern und solange der Urheberrechtsinhaber nicht ihre Lizenz beendet, und (b) dauerhaft, wenn der Urheberrechtsinhaber es versäumt, Sie über den Verstoß zu benachrichtigen, mit angemessenen Mitteln, vorzeitig 60 Tage nach der Beendigung.

Darüber hinaus ist Ihre Lizenz eines bestimmten Urheberrechtsinhabers wieder dauerhaft, wenn der Urheberrechtsinhaber Sie über den Verstoß mit angemessenen Mitteln benachrichtigt, es das erste Mal ist, dass Sie eine Benachrichtigung über den Verstoß dieser Lizenz (für jedes Werk) vom Urheberrechtsinhaber erhalten, und Sie den Verstoß vorzeitig 30 Tage nach Erhalt der Benachrichtigung beseitigen.

Die Beendigung Ihrer Rechte unter diesem Abschnitt beendet nicht die Lizenzen der Parteien, welche Kopien oder Rechte von Ihnen unter dieser Lizenz erhalten haben. Wenn Ihre Rechte aufgekündigt und nicht wieder dauerhaft geworden sind, gibt Ihnen der Erhalt einer Kopie von Teilen oder Sämtlichem des gleichen Materials keine Rechte, diese zu verwenden.

### 10. Zukünftige Überarbeitungen dieser Lizenz

Die Free Software Foundation kann von Zeit zu Zeit neue, überarbeitete Versionen der GNU-Lizenz für freie Dokumentation veröffentlichen. Solche neuen Versionen werden im Geist zur gegenwärtigen Version gleich sein, aber können sich im Detail unterscheiden, um neue Probleme oder Anliegen anzusprechen. Sehen Sie <http://www.gnu.org/copyleft/>.

Jeder Version der Lizenz wird eine charakteristische Versionsnummer gegeben. Wenn das Dokument spezifiziert, dass eine besonders nummerierte Version dieser Lizenz ‘oder jede spätere Version’ auf sich zutrifft, haben Sie die Wahl den Bestimmungen und Bedingungen entweder dieser spezifizierten Version oder jeder späteren Version, die (nicht als Entwurf) durch die Free Software Foundation veröffentlicht worden ist, zu folgen. Wenn das Dokument spezifiziert, dass ein Bevollmächtigter entscheiden darf welche zukünftigen Versionen dieser Lizenz verwendet werden können, berechtigt Sie das dauerhaft, durch die öffentliche Aussage desjenigen Bevollmächtigten über die Akzeptanz einer Version, diese Version für das Dokument zu wählen.

### 11. Relizenzierung

“Massive Multiauthor Collaboration Site” (or “MMC Site”) meint jeglichen World Wide Web Server, welcher urheberrechtlichsfähige Werke veröffentlicht und ferner bedeutende Möglichkeiten für jeden bietet, solche Werke zu editieren. Ein öffentliches Wiki, welches jeder bearbeiten kann, ist ein Beispiel eines solchen Servers. Eine ‘Massives Multiautor Collaboration’ (oder ‘MMC’), enthalten im Standort, bedeutet jegliche Zusammenstellung urheberrechtlichsfähiger Werke, die folglich auf dem MMC-Standort veröffentlicht werden.

“CC-BY-SA” meint die Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Lizenz, veröffentlicht von der Creative Commons Corporation, einem gemeinnützigen Unternehmen mit Hauptsitz in San Francisco, Californien, sowie zukünftige Copyleft-Versionen dieser Lizenz, welche durch dieselbe Organisation veröffentlicht wird.

“Einbeziehen” bedeutet, ein Dokument zu veröffentlichen oder neu zu veröffentlichen, ganz oder teilweise, als Bestandteil eines anderen Dokuments.

Eine MMC ist “geeignet zur Relizenzierung”, wenn sie unter dieser Lizenz lizenziert wird, und wenn alle Werke, die zuerst unter dieser Lizenz irgendwo anders als auf der MMC veröffentlicht, und nachträglich ganz oder teilweise in der MMC einbezogen wurden, (1) keine Umschlagtexte oder unveränderliche Abschnitte hatten, und (2) folglich vor dem 1. November 2008 einbezogen wurden.

Betreiber eines MMC-Standortes kann eine MMC, enthalten im Standort unter CC-BY-SA des gleichen Standortes, jederzeit vor dem 1. August 2009 neu veröffentlichen, vorausgesetzt, die MMC ist für die Relizenzierung geeignet.

#### Anhang: Wie Sie diese Lizenz für Ihre Dokumente verwenden

Um diese Lizenz in einem Dokument, das Sie geschrieben haben, zu verwenden, nehmen Sie eine Kopie der Lizenz in das Dokument auf und setzen Sie die folgenden Urheberrechts- und Lizenzhinweise unmittelbar hinter das Titelblatt:

Copyright © JAHR IHR NAME. Es wird die Erlaubnis gewährt, dieses Dokument zu kopieren, zu verteilen und/oder zu modifizieren, unter den Bestimmungen der GNU Free Documentation License,



Version 1.3 oder jeder späteren Version, veröffentlicht von der Free Software Foundation; ohne unveränderliche Abschnitte, ohne vordere Umschlagtexte und ohne hintere Umschlagtexte. Eine Kopie der Lizenz wird im Abschnitt mit dem Titel 'GNU Free Documentation License' bereitgestellt.

Wenn Sie unveränderliche Abschnitte, vordere Umschlagtexte und hintere Umschlagtexte haben, ersetzen Sie die Zeile 'ohne ...Umschlagtexte.' durch diese:

mit den unveränderlichen Abschnitten AUFGELISTET NACH TITELN, mit den vorderen Umschlagtexten AUFGELISTET und mit den hinteren Umschlagtexten AUFGELISTET.

Wenn Sie unveränderliche Abschnitte ohne Umschlagtexte haben oder eine andere Kombination der drei, vermischen Sie jene zwei Alternativen zu einer übereinstimmenden Situation.

Wenn Ihr Dokument nicht-triviale Beispiele eines Programmcodes enthält, empfehlen wir, diese Beispiele entsprechend unter einer freien Software-Lizenz ihrer Wahl, wie der GNU General Public License freizugeben, um deren Verwendung in freier Software zu gestatten.

.



---

## Literatur und Internetreferenzen

---

GDAL-SOFTWARE-SUITE. Geospatial data abstraction library. <http://www.gdal.org>, 2013.

GRASS-PROJECT. Geographic resource analysis support system. <http://grass.osgeo.org> , 2013.

NETELER, M., AND MITASOVA, H. Open source gis: A grass gis approach, 2008.

OGR-SOFTWARE-SUITE. Geospatial data abstraction library. <http://www.gdal.org/ogr> , 2013.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.1.1) implementation specification. <http://portal.opengeospatial.org>, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.3.0) implementation specification. <http://portal.opengeospatial.org>, 2004.

POSTGIS-PROJECT. Räumliche Unterstützung für Postgresql. <http://postgis.refrains.net/> , 2013.



- 
- %%, 109
- Aktionen, 107
- apache, 171
- apache2, 171
- Arc/Info\_ASCII\_Grid, 147
- Arc/Info\_Binary\_Grid, 147
- ArcInfo\_Binary\_Coverage, 71
- Atlas\_Generation, 691
- Attribute von Objekten vereinen, 132
- Attribute\_Actions, 107
- Attribute\_Table, 680
- Attribute\_Table\_Selection, 134
- Attributtabelle, 134
- Ausdrücke, 115
- Ausgabe als Bild speichern, 20
- Avoid\_Intersections\_Of\_Polygons, 124
- Bearbeitungsstatus umschalten, 125
- Beispielaktionen, 109
- Beschriftung, 42
- Beziehungen, 138
- Browse\_Maps, 65
- Calculator\_Field, 143
- CAT, 161
- Categorized\_Renderer, 87
- CGI, 170
- Colliding\_labels, 97
- Color\_interpolation , 153
- color\_Ramp, 84
- colorBrewer, 84
- Comma Separated Values, 71
- Common\_Gateway\_Interface, 170
- Compose\_Maps, 653
- Composer\_Manager, 695
- Composer\_Template, 654
- Contrast\_enhancement , 151
- Coordinate\_Reference\_System, 165
- Create\_Maps, 653
- Create\_New\_Layers, 132
- CRS, 165
- CSV, 71, 127
- Current\_Edits, 125
- Custom\_color\_Ramp, 84
- Custom\_CRS, 62
- Darstellung anhalten, 35
- Darstellungsqualität, 35
- Darstellungsupdate während der Wiedergabe, 35
- Datum\_transformation, 64
- DB\_Manager, 78
- Debian\_Squeeze, 171
- default\_CRS, 59
- Derived\_Fields, 143
- Digitalisieren, 124
- Diskret, 153
- Displacement\_plugin, 91
- Dokumentation, 7
- Drucken
  - Export\_Map, 694
- Druckzusammenstellung Schnelldruck, 20
- eine Aktion zu definieren, 107
- eine Shapedatei zu laden, 68
- Elements\_Alignment, 689
- EPSG, 59
- Equal\_Interval, 89
- Erdas Imagine, 147
- Erweiterungen , 697
- ESRI, 67
- European\_Petroleum\_Search\_Group, 59
- Export\_as\_image, 694
- Export\_as\_PDF, 694
- Export\_as\_SVG, 694
- Fangen, 122
- Farabbildung, 153
- FastCGI, 170
- Field\_Calculator, 143
- Field\_Calculator\_Functions, 116
- Forschungswerkzeuge, 718
- fTools Funktionen, 717
- GDAL, 147
- Georeferenzierwerkzeuge, 724
- GeoTIFF, 147
- GiST (Generalized Search Tree) Index, 77
- Gitter
  - Grids
    - Map\_Grid, 663
-

- GML, 161
- Gradient\_color\_Ramp, 84
- Graduated\_Renderer, 89
- GRASS, 187, *siehe* Creating new vectors; editing; creating a new layer
  - attribute linkage, 192
  - attribute storage, 192
  - category settings, 194
  - digitizing tools, 193
  - display results, 201, 202
  - region, 195
  - region display, 195
  - region editing, 197
  - snapping tolerance, 194
  - symbolology settings, 195
  - table editing, 195
  - toolbox, 203
- GRASS toolbox, 197
  - Browser, 206
  - customize, 208
- GRASS Vektordatenmodell, 192
  
- Hauptfenster, 21
- Hilfe, 33
- Histogramm, 155
- HTML\_Frame, 686
  
- IGNF, 59
- Import\_Maps, 65
- Institut\_Geographique\_National\_de\_France, 59
- InteProxy, 168
- Inverted\_Polygon\_Renderer, 93
  
- Joinlayer, 112
  
- Kartenübersicht, 47
- KBS, 59
- Knoten, 126
- Koordinatenbezugssystem, 59
  
- Layeranzeige kontrollieren, 34
- Layersichtbarkeit, 29
- Layout\_Maps, 653
- Legende, 29
- Lesezeichen, 44
- Lizenzurkunde, 757
- loading\_raster, 147
  
- Maßstab, 34
- Maßstab berechnen, 32
- Maßstabsabhängige Layerdarstellung, 34
- Map\_Legend, 670
- Map\_Navigation, 123
- Map\_Template, 654
- MapInfo, 71
- Menüs, 22
- Merge\_Attributes\_of\_Selected\_Features, 132
- Merge\_Selected\_Features, 131
- Messen, 35
  - Flächen, 36
  - Linienlänge, 36
  - Winkel, 36
- Metadaten, 156
  - mit den Pfeiltasten verschieben, 32
- MSSQL Spatial, 78
- Multi\_Band\_Raster, 150
- multipolygon, 130
  
- Natural\_Breaks\_(Jenks), 89
- New\_GPX\_Layer, 132, 134
- New\_Shapefile\_Layer, 132
- New\_SpatialLite\_Layer, 132
- New\_Spatiallite\_Layer, 134
- Node\_Tool, 126
- Non\_Spatial\_Attribute\_Tables, 137
  
- Objekte abfragen, 37
- OGC, 161
- OGR, 67
- OGR Simple Feature Library, 67
- ogr2ogr, 76
- Open\_Geospatial\_Consortium, 161
- OpenStreetMap, 73
- Optionen in der Kommandozeile, 17
- Oracle Spatial, 79
- OSM, 73
  
- pgsql2shp, 76
- Picture\_database, 668
- Point\_Displacement\_Renderer, 91
- PostGIS, 73
- PostGIS räumlicher Index, 77
- PostgreSQL, 73
- Pretty\_Breaks, 89
- print\_composer
  - tools, 653
- Proj.4, 62
- Proj4, 61
- Proj4\_text, 61
- Projekte verschachteln, 44
- Projektionen, 59
- Proxy, 163
- proxy-server, 163
- Pyramiden, 155
  
- QGIS\_mapserver, 169
- QGIS\_Server, 170
- QSpatialite, 78
- Quantile, 89
- Query\_Builder, 141
  
- Räumliche Lesezeichen
  - siehe* Lesezeichen, 44
- Raster, 147
- Raster\_Calculator, 156
- Renderer\_Categorized, 87
- Renderer\_Graduated, 89
- Renderer\_Point\_Displacement, 91

- Renderer\_Single\_Symbol, 87
- Rendering\_Mode, 659
- Rendering\_Rule-based, 91
- Revert\_Layout\_Actions, 690
- Ring-Polygone, 130
- Rotate\_Point\_symbols, 132
- Rotated\_North\_Arrow, 668
- Rule-based\_Rendering, 91
  
- Scalebar
  - Map\_Scalebar, 675
- Search\_Radius, 123
- Secured\_OGC\_Authentication, 168
- Select\_using\_Query, 143
- SFS, 161
- Shapedatei, 67
- Shapefile\_to\_Postgis\_Import\_Tool, 747
- Shared\_Polygon\_Boundaries, 124
- shp2pgsql, 76
- Single\_Band\_Raster , 150
- Single\_Symbol\_Renderer, 87
- SLD, 170
- SLD/SE, 170
- Snapping\_On\_Intersections, 124
- Snapping\_Tolerance, 122
- Spatialite, 78
- Spatialite\_Manager, 78
- SPIT, 747
- Split\_Features, 131
- SQLite, 78
- SRS, 165
- ST\_Shift\_Longitude, 77
- Stützpunkt, 126
- Stützpunkte, 126
- Symbolisierung, 96
- Symbologie, 150
  
- Tastenkürzel, 33
- Three\_Band\_Color\_Raster, 150
- Tiger\_Format, 71
- Topological\_Editing, 124
- Transparenz, 154
  
- UK\_National\_Transfer\_Format, 71
- US\_Census\_Bureau, 71
  
- Vektorlayer, der über den Längengrad 180 hinausgeht, 77
- verknüpfen, 112
- Verschieben, 123
  
- WCS, 161, 169
- Web Coverage Service, 169
- Werkzeugleiste, 28
- Werkzeugleisten gestalten, 28
- WFS, 161, 169
- WFS-T, 169
- WFS\_Transactional, 169
- WKT, 59, 127
- WMS, 161
- WMS-C, 166
- WMS\_1.3.0, 169
- WMS\_client, 161
- WMS\_identify, 166
- WMS\_layer\_transparency, 165
- WMS\_metadata, 167
- WMS\_properties, 167
- WMS\_tiles, 166
- WMTS, 166
- WMTS\_client, 161
- Work\_with\_Attribute\_Table, 134
  
- Zoom\_In Zoom\_Out, 123
- Zoomen mit dem Mausrad, 31