QGIS User Guide

Release 2.0

QGIS Project

December 15, 2013

Contents

CHAPTER 1

Preamble

This document is the original user guide of the described software QGIS. The software and hardware described in this document are in most cases registered trademarks and are therefore subject to the legal requirements. QGIS is subject to the GNU General Public License. Find more information on the QGIS Homepage http://www.qgis.org.

The details, data, results etc. in this document have been written and verified to the best of knowledge and responsibility of the authors and editors. Nevertheless, mistakes concerning the content are possible.

Therefore, all data are not liable to any duties or guarantees. The authors, editors and publishers do not take any responsibility or liability for failures and their consequences. You are always welcome to indicate possible mistakes.

This document has been typeset with reStructuredText. It is available as reST source code via github and online as HTML and PDF via http://www.qgis.org/en/docs/. Translated versions of this document can be downloaded in several formats via the documentation area of the QGIS project as well. For more information about contributing to this document and about translating it, please visit: http://www.qgis.org/wiki/.

Links in this Document

This document contains internal and external links. Clicking on an internal link moves within the document, while clicking on an external link opens an internet address. In PDF form, internal and external links are shown in blue and are handled by the system browser. In HTML form, the browser displays and handles both identically.

User, Installation and Coding Guide Authors and Editors:

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 Nijjsen	Richard Duivenvoorde	Andreas Neumann
Astrid Emde	Yves Jacolin	Alexandre Neto	Andy Schmid	Hien Tran-Quang

Internet: http://www.qgis.org

License of this document

CHAPTER 2

Conventions

This section describes a collection of uniform styles throughout the manual. The conventions used in this manual are as follows:

2.1 GUI Conventions

The GUI convention styles are intended to mimic the appearance of the GUI. In general, the objective is to use the non-hover appearance, so a user can visually scan the GUI to find something that looks like the instruction in the manual.

- Menu Options: *Layer* → *Add a Raster Layer* or *Settings* → *Toolbars* → *Digitizing*
- Tool:  Add a Raster Layer
- Button : **[Save as Default]**
- Dialog Box Title: *Layer Properties*
- Tab: *General*
- Checkbox:  *Render*
- Radio Button:  *Postgis SRID*  *EPSG ID*
- Select a Number: 

- Select a String: 
- Browse for a File: 
- Select a Color: 

- Slider: 
- Input Text: 

A shadow indicates a clickable GUI component.

2.2 Text or Keyboard Conventions

The manual also includes styles related to text, keyboard commands and coding to indicate different entities, such as classes, or methods. They don't correspond to any actual appearance.

- Hyperlinks: http://qgis.org
- Keystroke Combinations: press `Ctrl+B`, meaning press and hold the Ctrl key and then press the B key.
- Name of a File: `lakes.shp`

- Name of a Class: **NewLayer**
- Method: *classFactory*
- Server: *myhost.de*
- User Text: `qgis --help`

Lines of code are indicated by a fixed-width font

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

2.3 Platform-specific instructions

GUI sequences and small amounts of text can be formatted inline: Click *File* *QGIS → Quit to close QGIS*.

This indicates that on Linux, Unix and Windows platforms, click the File menu option first, then Quit from the dropdown menu, while on Macintosh OSX platforms, click the QGIS menu option first, then Quit from the dropdown menu. Larger amounts of text may be formatted as a list:

- do this;
- do that;
- do something else.

or as paragraphs.

Do this and this and this. Then do this and this and this and this and this and this and this and this and this.

Do that. Then do that and that and that and that and that and that and that and that and that and that and that and that and that and that and that.

Screenshots that appear throughout the user guide have been created on different platforms; the platform is indicated by the platform-specific icon at the end of the figure caption.

CHAPTER 3

Foreword

Welcome to the wonderful world of Geographical Information Systems (GIS)!

QGIS is an Open Source Geographic Information System. The project was born in May of 2002 and was established as a project on SourceForge in June of the same year. We've worked hard to make GIS software (which is traditionally expensive proprietary software) a viable prospect for anyone with basic access to a Personal Computer. QGIS currently runs on most Unix platforms, Windows, and OS X. QGIS is developed using the Qt toolkit (http://qt.digia.com) and C++. This means that QGIS feels snappy to use and has a pleasing, easy-to-use graphical user interface (GUI).

QGIS aims to be an easy-to-use GIS, providing common functions and features. The initial goal was to provide a GIS data viewer. QGIS has reached the point in its evolution where it is being used by many for their daily GIS data viewing needs. QGIS supports a number of raster and vector data formats, with new format support easily added using the plugin architecture.

QGIS is released under the GNU General Public License (GPL). Developing QGIS under this license means that you can inspect and modify the source code, and guarantees that you, our happy user, will always have access to a GIS program that is free of cost and can be freely modified. You should have received a full copy of the license with your copy of QGIS, and you also can find it in Appendix *GNU General Public License*.

Tip: Up-to-date Documentation

The latest version of this document can always be found in the documentation area of the QGIS website at http://www.qgis.org/en/docs/

CHAPTER 4

Features

QGIS offers many common GIS functionalities provided by core features and plugins. As a short summary they are presented in six categories to gain a first insight.

4.1 View data

You can view and overlay vector and raster data in different formats and projections without conversion to an internal or common format. Supported formats include:

- Spatially-enabled tables and views using PostGIS, SpatiaLite and MSSQL Spatial, Oracle Spatial, vector formats supported by the installed OGR library, including ESRI shapefiles, MapInfo, SDTS, GML and many more, see section *Working with Vector Data*.
- Raster and imagery formats supported by the installed GDAL (Geospatial Data Abstraction Library) library, such as GeoTiff, Erdas Img., ArcInfo Ascii Grid, JPEG, PNG and many more, see section *Working with Raster Data*.
- QGIS processing framework to call hundreds of native and third party algorithms from QGIS, see section Processing *Introduction*.
- GRASS raster and vector data from GRASS databases (location/mapset), see section *GRASS GIS Integration*.
- Online spatial data served as OGC Web Services, such as (WMS, WMTS, WCS, WFS, WFS-T, ...), see section *Working with OGC Data*.
- OpenStreetMap data, see section *plugins_osm*.

4.2 Explore data and compose maps

You can compose maps and interactively explore spatial data with a friendly GUI. The many helpful tools available in the GUI include e.g.:

- QGIS browser
- On-the-fly reprojection
- DB Manager
- Map composer
- Overview panel
- Spatial bookmarks
- Annotation tools
- Identify/select features

- Edit/view/search attributes
- Feature labeling also data defined
- Change vector and raster symbology also data defined
- Add a graticule layers to create an atlas map composition
- Decorate your map with a north arrow scale bar and copyright label
- Save and restore projects

4.3 Create, edit, manage and export data

You can create, edit, manage and export vector and raster layers in several formats. QGIS offers e.g. the following:

- Digitizing tools for OGR supported formats and GRASS vector layer
- Create and edit shapefiles and GRASS vector layers
- Geocode images with the Georeferencer plugin
- GPS tools to import and export GPX format, and convert other GPS formats to GPX or down/upload directly to a GPS unit (on Linux, usb: has been addedto list of GPS devices)
- Visualize and edit OpenStreetMap data
- Create spatial database tables from shapefiles with DB Manager plugin
- Improved handling of spatial database tables
- Manage vector attribute tables
- Save screenshots as georeferenced images

4.4 Analyse data

You can perform spatial data analysis on spatial databases and other OGR supported formats. QGIS currently offers vector analysis, sampling, geoprocessing, geometry and database management tools. You can also use the integrated GRASS tools, which include the complete GRASS functionality of more than 400 modules (See Section *GRASS GIS Integration*). Or you work with the Processing Plugin, which provides powerful geospatial analysis framework to call native and third party algorithms from QGIS, such as GDAL, SAGA, GRASS, fTools and more (see section *Introduction*).

4.5 Publish maps on the Internet

QGIS can be used as a WMS, WMTS, WMS-C or WFS and WFS-T client, and as WMS or WFS server (see section *Working with OGC Data*). Additionally you can export data publish them on the Internet using a webserver with UMN MapServer or GeoServer installed.

4.6 Extend QGIS functionality through plugins

QGIS can be adapted to your special needs with the extensible plugin architecture. QGIS provides libraries that can be used to create plugins. You can even create new applications with C++ or Python!

4.6.1 Core Plugins

1. Coordinate Capture (Capture mouse coordinates in different CRS)
2. DB Manager (Exchange, edit and view layers and tables; execute SQL queries)
3. Diagram Overlay (Placing diagrams on vector layer)
4. Dxf2Shp Converter (Convert DXF to Shape)
5. eVIS (Event Visualization Tool)
6. fTools (Tools for vector data analysis and management)
7. GDALTools (Integrate GDAL Tools into QGIS)
8. Georeferencer GDAL (Adding projection information to raster using GDAL)
9. GPS Tools (Loading and importing GPS data)
10. GRASS (GRASS GIS integration)
11. Heatmap (Generating raster heatmaps from point data)
12. Interpolation plugin (interpolate based on vertices of a vector layer)
13. Mapserver Export (Export QGIS project file to a MapServer map file)
14. Offline Editing (Allow offline editing and synchronizing with database)
15. Open Layers plugin (OpenStreetMap, Google Maps, Bing Maps layers and more)
16. Oracle Spatial GeoRaster
17. Processing (formerly SEXTANTE)
18. Raster terrain analysis (Raster based terrain analysis)
19. Road graph plugin (Shortest Path network analysis)
20. Spatial Query Plugin
21. SPIT (Import Shapefile to PostgreSQL/PostGIS)
22. SQL Anywhere Plugin (Store vector layers within a SQL Anywhere database)
23. Topology Checker (Finding topological errors in vector layers)
24. Zonal statistics plugin (Calculate count, sum, mean of raster for each polygon of a vector layer)

4.6.2 External Python Plugins

QGIS offers a growing number of external python plugins that are provided by the community. These plugins reside in the official plugins repository, and can be easily installed using the Python Plugin Installer (See Section *Loading an external QGIS Plugin*).

4.7 Python Console

For scripting, it is possible to take advantage of an integrated Python console. It can be opened from menu: *Plugins → Python Console*. The console opens as a non-modal utility window. For interaction with the QGIS environment, there is the `qgis.utils.iface` variable, which is an instance of `QgsInterface`. This interface allows access to the map canvas, menus, toolbars and other parts of the QGIS application.

For further information about working with the Python Console and Programming Py|qg| plugins and applications, please refer to http://www.qgis.org/html/en/docs/pyqgis_developer_cookbook/index.html.

What's new in QGIS 2.0

Please note that this is a release in our 'cutting edge' release series. As such it contains new features and extends the programmatic interface over QGIS 1.8.0. We recommend that you use this version over previous releases.

This release includes hundreds of bug fixes and many new features and enhancements that will be described in this manual. Also compare with the visual changelog at http://changelog.linfiniti.com/version/1/

5.1 User Interface

- **New icon theme**: We have updated our icon theme to use the 'GIS' theme introducing an improved level of consistency and professionalism to the QGIS user interface.
- **Side tabs, collapsable groups**: We have standardised the layout of tabs and introduced collapsible group boxes into many of our dialogs to make navigating the various options more easy, and to make better use of screen real estate.
- **Soft notifications**: In many cases we want to tell you something, but we don't want to stop your work or get in your way. With the new notification system QGIS can let you know about important information via a message bar (colour depends on the importance of the message) that appears at the top of the map canvas but doesn't force you to deal with it if you are busy doing something else. Programmers can create these notification (e.g. from a plugin) to using our python API.
- **Application custom font and Qt stylesheet**: The system font used for the application's user interface can now be set. Any C++ or Python plugin that is a child of the QGIS GUI application or has copied/applied the application's GUI stylesheet can inherit its styling, which is useful for GUI fixes across platforms and when using custom QGIS Qt widgets, like QgsCollapsibleGroupBox.
- **Live color chooser dialogs and buttons**: Every color chooser button throughout the interface has been updated to give visual feedback on whether the current color has a transparent, or 'alpha,' component. The color chooser opened by the new color buttons will now always be the default for the operating system. If the user has Use live-updating color chooser dialogs checked under *Options -> General -> Application* , any change in the color chooser will immediately be shown in the color button and for any item currently being edited, where applicable.
- **SVG Annotations**: With QGIS 2.0 you can now add SVG annotations to your map - either pinned to a specific place or in a relative position over the map canvas.

5.2 Data Provider

- **Oracle Spatial support**: QGIS 2.0 now includes Oracle Spatial support.
- **Web Coverage Service provider added**: QGIS now provides native support for Web Coverage Service layers - the process for adding WCS is similar to adding a WMS layer or WFS layer.

- **Raster Data Provider overhaul**: The raster data provider system has been completely overhauled. One of the best new features stemming from this work is the ability to *Layer -> Save As...* to save any raster layer as a new layer. In the process you can clip, resample, and reproject the layer to a new Coordinate Reference System. You can also save a raster layer as a rendered image so if you for example have single band raster that you have applied a colour palette to, you can save the rendered layer out to a georeferenced RGB layer.
- **Raster 2% cumulative cut by default**: Many raster imagery products have a large number of outliers which result in images having a washed out appearance. QGIS 2.0 intoduces much more fine grained control over the rendering behaviour of rasters, including using a 2% - 98% percent cumulative cut by default when determining the colour space for the image.
- **WMS identify format**: It is now possible to select the format of the identify tool result for WMS layers if multiple known formats are supported by the server. The supported formats are HTML, feature (GML) and plain text. If the feature (GML) format is selected, the result is in the same form as for vector layers, the geometry may be highlighted and the feature including attributes and geometry may be copied to clipboard and pasted to another layer.
- **WMTS Support**: The WMS client in QGIS now supports WMTS (Web Mapping Tile Service) including selection of sub-datasets such as time slices. When adding a WMS layer from a compliant server, you will be prompted to select the time slice to display.

5.3 Symbology

- **Data defined properties**: With the new data defined properties, it is possible to control symbol type, size, color, rotation, and many other properties through feature attributes.
- **Improved symbol layer management**: The new symbol layer overview uses a clear, tree-structured layout which allows for easy and fast access to all symbol layers.
- **Support for transparency in colour definitions**: In most places where you select colours, QGIS now allows you to specify the alpha channel (which determins how transparent the colour should be). This allows you to create great looking maps and to hide data easily that you don't want users to see.
- **Color Control for Raster Layers**: QGIS 2.0 allows you to precisely control exactly how you'd like raster layers to appear. You now have complete control over the brightness, contrast and saturation of raster layers. There's even options to allow display of rasters in grayscale or by colorising with a specified color.
- **Copy symbology between layers**: Its now super easy to copy symbology from one layer to another layer. If you are working with several similar layer, you can simply right-click on one layer, choose Copy Style from the context menu and then right-click on another layer and choose Paste-Style.
- **Save styles in your database**: If you are using a database vector data store, you can now store the layer style definitions directly in the database. This makes it easy to share styled layers in an enterprise or multi-user environment.
- **Colour ramp support**: Colour ramps are now available in many places in QGIS symbology settings and QGIS ships with a rich, extensible set of colour ramps. You can also design your own and many cpt-city themes are included in QGIS now 'out of the box'. Color ramps even have full support for transparency!
- **Set custom default styles for all layer types**: Now QGIS lets you control how new layers will be drawn when they do not have an existing .qml style defined. You can also set the default transparency level for new layers and whether symbols should have random colours assigned to them.

5.4 Map Composer

- **HTML Map Items**: You can now place html elements onto your map.
- **Auto snap lines**: Having nicely align map items is critical to making nice printed maps. Auto snapping lines have been added to allow for easy composer object alignment by simply dragging an object close to another.

- **Manual Snap Lines**: Sometimes you need to align objects a curtain distance on the composer. With the new manual snapping lines you are able to add manual snap lines which allow for better align objects using a common alignment. Simply drag from the top or side ruler to add new guide line.
- **Map series generation**: Ever needed to generate a map series? Of course you have. The composer now includes built in map series generation using the atlas feature. Coverage layers can be points, lines, polygons, and the current feature attribute data is available in labels for on the fly value replacement.
- **Multipage support**: A single composer window can now contain more than one page.
- **Expressions in composer labels**: The composer label item in 1.8 was quite limited and only allowed a single token $CURRENT_DATE to be used. In 2.0 full expression support has been added too greater power and control of the final labels.
- **Automatic overview support in map frame**: Need to show the current area of the main map frame in a smaller overview window. Now you can. The map frame now contains the ability to show the extents of other and will update when moved. Using this with the atlas generation feature now core in the composer allows for some slick map generation. Overview frame style uses the same styling as a normal map polygon object so your creativity is never restricted.
- **Layer blending**: Layer blending makes it possible to combine layers in new and exciting ways. While in older versions, all you could do was to make the layer transparent, you can now choose between much more advanced options such as "multiply", "darken only", and many more. Blending can be used in the normal map view as well as in print composer. For a short tutorial on how to use blending in print composer to make the most out of background images, see "Vintage map design using QGIS".
- **HTML Label support**: HTML support has been added map composer label item to give you even more control over your final maps. HTML labels support full css styles sheets, html, and even javascript if you are that way inclined.
- **Multicolumn composer legend**: The composer legend now supports multiple columns. Splitting of a single layer with many classes into multiple columns is optional. Single symbol layers are now added by default as single line item. Three different styles may be assigned to layer/group title: Group, Subgroup or Hidden. Title styles allow arbitrary visual grouping of items. For example, a single symbol layer may be displayed as single line item or with layer title (like in 1.8), symbols from multiple following layers may be grouped into a single group (hiding titles) etc. Feature counts may be added to labels.
- **Updates to map composer management**: The following improvements have been made to map composer management:
  - Composer name can now be defined upon creation, optionally choosing to start from other composer names
  - Composers can now be duplicated
  - New from Template and from Specific (in Composer Manager) creates a composer from a template located anywhere on the filesystem
  - Parent project can now be saved directly from the composer work space
  - All composer management actions now accessible directly from the composer work space

5.5 Labeling

- **New labeling system**: The labeling system has been totally overhauled - it now includes many new features such as drop shadows, 'highway shields', many more data bound options, and various performance enhancements. We are slowly doing away with the 'old labels' system, although you will still find that functionality available for this release, you should expect that it will disappear in a follow up release.
- **Expression based label properties**: The full power of normal label and rule expressions can now be used for label properties. Nearly every property can be defined with an expression or field value giving you more control over the label result. Expressions can refer to a field (e.g. set the font size to the value of the field 'font') or can include more complex logic.

- **Older labeling engine deprecated**: Use of the older labeling engine available in QGIS <= 1.8 is now discouraged (i.e. deprecated), but has not been removed. This is to allow users to migrate existing projects from the old to new labeling engine. The following guidelines for working with the older engine in QGIS 2.0 apply:
  - Deprecated labeling tab is removed from vector layer properties dialog for new projects or older opened projects that don't use that labeling engine.
  - Deprecated tab remains active for older opened projects, if any layer uses them, and does not go away even if saving the project with no layers having the older labeling engine enabled.
  - Deprecated labeling tab can be enabled/disabled for the current project, via Python console commands. Please note: There is a very high likelihood the deprecated labelling engine will be completely removed prior to the next stable release of QGIS. Please migrate older projects.

5.6 Programmability

- **New Python Console**: The new Python console gives you even more power. Now the with auto complete support, syntax highlighting, adjustable font settings. The side code editor allows for easier entry of larger blocks of code with the ability to open and run any Python file in the QGIS session.
- **Even more expression functions**: With the expression engine being used more and more though out QGIS to allow for things like expression based labels and symbol, many more functions have been added to the expression builder and are all accessible through the expression builder. All functions include comprehensive help and usage guides for ease of use.
- **Custom expression functions**: If the expression engine doesn't have the function that you need. Not to worry. New functions can be added via a plugin using a simple Python API.
- **New cleaner Python API**: The Python API has been revamped to allow for a more cleaner, more pythonic, programming experience. The QGIS 2.0 API uses SIP V2 which removes the messy toString(), toInt() logic that was needed when working with values. Types are now converted into native Python types making for a much nicer API. Attributes access is now done on the feature itself using a simple key lookup, no more index lookup and attribute maps.
- **Code compatibility with version 1.x releases**: As this is a major release, it is not completely API compatible with previous 1.x releases. In most cases porting your code should be fairly straightforward - you can use this guide to get started. Please use the developer mailing list if you need further help.
- **Python project macros**: A Python module, saved into a project.qgs file, can be loaded and have specific functions run on the following project events: openProject(), saveProject() and closeProject(). Whether the macros are run can be configured in the application options.

5.7 Analysis tools

- **Processing Commander**: For quick access to geoprocessing functionality, just launch the processing commander (Ctrl + Alt + M) and start typing the name of the tool you are looking for. Commander will show you the available options and launch them for you. No more searching through menus to find tools. They are now right at your fingertips.
- **Heatmap Plugin Improvements**: The heatmap plugin has seen numerous improvements and optimisations, resulting in much faster creation of heatmaps. Additionally, you now have the choice of which kernel function is used to create the heatmap.
- **Processing Support**: The SEXTANTE project has been ported to and incorporated into QGIS as core functionality. SEXTANTE has been renamed to 'Processing' and introduces a new menu in QGIS from where you can access a rich toolbox of spatial analysis tools. The processing toolbox has incredibly rich functionality - with a python programming API allowing you to easily add new tools, and hooks to provide access to analysis capabilities of many popular open source tools such as GRASS, OTB, SAGA etc.

- **Processing Modeller**: One of the great features of the new processing framework is the ability to combine the tools graphically. Using the Processing Modeller, you can build up complex analysis from a series of small single purpose modules. You can save these models and then use them as building blocks in even more complex models. Awesome power integrated right into QGIS and very easy to use!

5.8 Plugins

- **Revamped plugin manager**: In QGIS 1.x managing plugins was somewhat confusing with two interfaces - one for managing already installed plugins and one for fetching python plugins from an only plugin repository. In QGIS 2.0 we introduce a new, unified, plugin manager which provides a one stop shop for downloading, enabling/disabling and generally managing you plugins. Oh, and the user interface is gorgeous too with side tabs and easy to recognise icons!
- **Application and Project Options**: Define default startup project and project templates. With QGIS 2.0 you can specify what QGIS should do when it starts: New Project (legacy behaviour, starts with a blank project), Most recent (when you start QGIS it will load the last project you worked on), Specific (always load a specific project when QGIS starts). You can use the project template directory to specify where your template projects should be stored. Any project that you store in that directory will be available for use as a template when invoking the *Project* → *New* from template menu.
- **System environment variables**: Current system environment variables can now be viewed and many configured within the application Options dialog. Useful for platforms, such as Mac, where a GUI application does not necessarily inherit the user's shell environment. Also useful for setting/viewing environment variables for the external tool sets controlled by the processing toolbox, e.g. SAGA, GRASS; and, for turning on debugging output for specific sections of the source code.
- **User-defined zoom scales**: A listing of zoom scales can now be configured for the application and optionally overridden per project. The list will show up in the Scale popup combo box in the main window status bar, allowing for quick access to known scales for efficiently viewing and working with the current data sources. Defined scales can be exported to an XML file that can be imported into other projects or another QGIS application.

5.9 General

- **Quantum GIS is now known only as '|qg|'**: The 'Quantum' in 'Quantum GIS' never had any particular significance and the duality of referring to our project as both Quantum GIS and QGIS caused some confusion. We are streamlining our project and as part of that process we are officially dropping the use of the word Quantum - henceforth we will be known only as QGIS (spelled with all letters in upper case). We will be updating all our code and publicity material to reflect this.

5.10 Layer Legend

- **Legend visual feedback and options**
  - Total count for features in layer, as well as per symbol
  - Vector layers in edit mode now have a red pencil to indicate uncommitted (unsaved) edits
  - Active layer is now underlined, to indicate it in multi-layer selections or when there is no selection
  - Clicking in non-list-item whitespace now clears the selection
  - Right-clicks are now treated as left-clicks prior to showing the contextual menu, allowing for one click instead of two
  - Groups and layers can optionally be in a bold font style

  - Raster layer generated preview icons can now be turned off, for projects where such rendering may be slow

- **Duplicate existing map layer**: Duplicate selected vector and raster layers in the map layer legend. Similar to importing the same data source again, as a separate layer, then copy/pasting style and symbology attributes.
- **Multi-layer toggle editing commands**: User can now select multiple layers in legend and, if any of those are vector layers in edit mode, choose to save, rollback, or cancel current uncommitted edits. User can also choose to apply those actions across all layers, regardless of selection.

5.11 Browser

- **Improvements to in-app browser panel**:
  - Directories can be filtered by wildcard or regex expressions
  - New Project home (parent directory of current project)
  - View Properties of the selected directory in a dialog
  - Choose which directories to Fast scan
  - Choose to Add a directory directly to Favourites via filesystem browse dialog
  - New /Volumes on Mac (hidden directory for access to external drives)
  - New OWS group (collation of available map server connections)
  - Open a second browser (*View -> Panels -> Browser (2)*) for drag-n-drop interactions between browser panels
  - Icons now sorted by item group type (filesystem, databases, map servers)
  - Layer Properties now have better visual layout

CHAPTER 6

Getting Started

This chapter gives a quick overview of installing QGIS, some sample data from the QGIS web page and running a first and simple session visualizing raster and vector layers.

6.1 Installation

Installation of QGIS is very simple. Standard installer packages are available for MS Windows and Mac OS X. For many flavors of GNU/Linux binary packages (rpm and deb) or software repositories to add to your installation manager are provided. Get the latest information on binary packages at the QGIS website at http://download.qgis.org.

6.1.1 Installation from source

If you need to build QGIS from source, please refer to the installation instructions. They are distributed with the QGIS source code in a file called 'INSTALL'. You can also find it online at http://htmlpreview.github.io/?https://raw.github.com/qgis/QGIS/master/doc/INSTALL.html

6.1.2 Installation on external media

QGIS allows to define a `--configpath` option that overrides the default path (e.g. ~/.qgis2 under Linux) for user configuration and forces QSettings to use this directory, too. This allows users to e.g. carry a QGIS installation on a flash drive together with all plugins and settings. Also compare with section *System Menu*.

6.2 Sample Data

The user guide contains examples based on the QGIS sample dataset.

The Windows installer has an option to download the QGIS sample dataset. If checked, the data will be downloaded to your `My Documents` folder and placed in a folder called `GIS Database`. You may use Windows Explorer to move this folder to any convenient location. If you did not select the checkbox to install the sample dataset during the initial QGIS installation, you can either

- use GIS data that you already have;
- download sample data from at http://download.osgeo.org/qgis/data/qgis_sample_data.zip; or
- uninstall QGIS and reinstall with the data download option checked, only if the above solutions are unsuccessful.

For GNU/Linux and Mac OSX there are not yet dataset installation packages available as rpm, deb or dmg. To use the sample dataset download the file `qgis_sample_data` as ZIP archive from http://download.osgeo.org/qgis/data/qgis_sample_data.zip and unzip the archive on your system. The Alaska dataset includes all GIS data that are used as examples and screenshots in the user guide, and also includes a small GRASS database. The projection for the QGIS sample dataset is Alaska Albers Equal Area with unit feet. The EPSG code is 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]]
```

If you intend to use QGIS as graphical frontend for GRASS, you can find a selection of sample locations (e.g. Spearfish or South Dakota) at the official GRASS GIS website http://grass.osgeo.org/download/sample-data/.

6.3 Sample Session

Now that you have QGIS installed and a sample dataset available, we would like to demonstrate a short and simple QGIS sample session. We will visualize a raster and a vector layer. We will use the landcover raster layer `qgis_sample_data/raster/landcover.img` and the lakes vector layer `qgis_sample_data/gml/lakes.gml`.

6.3.1 Start QGIS

- Start QGIS by typing: "QGIS" at a command prompt, or if using precompiled binary, using the Applications menu.
- Start QGIS using the Start menu or desktop shortcut, or double click on a QGIS project file.
- Double click the icon in your Applications folder.

6.3.2 Load raster and vector layers from the sample dataset

1. Click on the Load Raster icon.
2. Browse to the folder `qgis_sample_data/raster/`, select the ERDAS Img file `landcover.img` and click **[Open]**.
3. If the file is not listed, check if the Filetype combobox at the bottom of the dialog is set on the right type, in this case "Erdas Imagine Images (`*.img`, `*.IMG`)".
4. Now click on the Load Vector icon.
5. *File* should be selected as Source Type in the new *Add Vector Layer* dialog. Now click **[Browse]** to select the vector layer.

6. Browse to the folder `qgis_sample_data/gml/`, select "GML" from the filetype combobox, then select the GML file `lakes.gml` and click **[Open]**, then in Add Vector dialog click **[OK]**.
7. Zoom in a bit to your favorite area with some lakes.
8. Double click the `lakes` layer in the map legend to open the *Properties* dialog.
9. Click on the *Style* menu and select a blue as fill color.
10. Click on the *Labels* menu and check the *Label this layer with* checkbox to enable labeling and choose "NAMES" field as field containing labels.
11. To improve readability of labels, you can add a white buffer around them, by clicking "Buffer" in the list on the left, checking *Draw text buffer* and choosing 3 as buffer size.
12. Click **[Apply]**, check if the result looks good and finally click **[OK]**.

You can see how easy it is to visualize raster and vector layers in QGIS. Let's move on to the sections that follow to learn more about the available functionality, features and settings and how to use them.

6.4 Starting and Stopping QGIS

In Section *Sample Session* you already learned how to start QGIS. We will repeat this here and you will see that QGIS also provides further command line options.

- Assuming that QGIS is installed in the PATH, you can start QGIS by typing: `qgis` at a command prompt or by double clicking on the QGIS application link (or shortcut) on the desktop or in the application menu.
- Start QGIS using the Start menu or desktop shortcut, or double click on a QGIS project file.
- Double click the icon in your Applications folder. If you need to start QGIS in a shell, run /path-to-installation-executable/Contents/MacOS/Qgis.

To stop QGIS, click the menu options *File* *QGIS* → *Quit*, or use the shortcut `Ctrl+Q`.

6.5 Command Line Options

QGIS supports a number of options when started from the command line. To get a list of the options, enter `qgis --help` on the command line. The usage statement for QGIS is:

```
    qgis --help
    QGIS - 2.0.1-Dufour 'Dufour' (exported)
QGIS is a user friendly Open Source Geographic Information System.
    Usage: qgis [OPTION] [FILE]
      options:
        [--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
        [--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
        [--help]                       this text

      FILES:
```

```
Files specified on the command line can include rasters,
vectors, and QGIS project files (.qgs):
1. Rasters - Supported formats include GeoTiff, DEM
   and others supported by GDAL
2. Vectors - Supported formats include ESRI Shapefiles
   and others supported by OGR and PostgreSQL layers using
   the PostGIS extension
```

Tip: Example Using command line arguments

You can start QGIS by specifying one or more data files on the command line. For example, assuming you are in the qgis_sample_data directory, you could start QGIS with a vector layer and a raster file set to load on startup using the following command: `qgis ./raster/landcover.img ./gml/lakes.gml`

Command line option `--snapshot`

This option allows you to create a snapshot in PNG format from the current view. This comes in handy when you have a lot of projects and want to generate snapshots from your data.

Currently it generates a PNG-file with 800x600 pixels. This can be adapted using the `--width` and `--height` command line arguments. A filename can be added after `--snapshot`.

Command line option `--lang`

Based on your locale QGIS, selects the correct localization. If you would like to change your language, you can specify a language code. For example: `--lang=it` starts QGIS in italian localization. A list of currently supported languages with language code and status is provided at http://hub.qgis.org/wiki/quantum-gis/GUI_Translation_Progress

Command line option `--project`

Starting QGIS with an existing project file is also possible. Just add the command line option `--project` followed by your project name and QGIS will open with all layers loaded described in the given file.

Command line option `--extent`

To start with a specific map extent use this option. You need to add the bounding box of your extent in the following order separated by a comma:

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

Command line option `--nologo`

This command line argument hides the splash screen when you start QGIS.

Command line option `--noplugins`

If you have trouble at startup with plugins, you can avoid loading them at startup. They will still be available in Plugins Manager after-wards.

Command line option `--nocustomization`

Using this command line argument existing GUI customization will not be applied at startup.

Command line option `--optionspath`

You can have multiple configurations and decide which one to use when starting QGIS using this option. See *Options* to check where does the operating system save the settings files. Presently there is no way to specify in which file where to write the settings, therefore you can create a copy of the original settings file and rename it.

Command line option `--configpath`

This option is similar to the one above, but furthermore overrides the default path (~/.qgis) for user configuration and forces QSettings to use this directory, too. This allows users to e.g. carry QGIS installation on a flash drive together with all plugins and settings.

6.6 Projects

The state of your QGIS session is considered a Project. QGIS works on one project at a time. Settings are either considered as being per-project, or as a default for new projects (see Section *Options*). QGIS can save the state of your workspace into a project file using the menu options *Project* → *Save* or *Project* → *Save As*.

Load saved projects into a QGIS session using *Project* → *Open ...*, *Project* → *New from template* or *Project* → *Open Recent*.

If you wish to clear your session and start fresh, choose *Project* → *New*. Either of these menu options will prompt you to save the existing project if changes have been made since it was opened or last saved.

The kinds of information saved in a project file include:

- Layers added
- Layer properties, including symbolization
- Projection for the map view
- Last viewed extent

The project file is saved in XML format, so it is possible to edit the file outside QGIS if you know what you are doing. The file format was updated several times compared to earlier QGIS versions. Project files from older QGIS versions may not work properly anymore. To be made aware of this, in the *General* tab under *Settings* → *Options* you can select:

Prompt to save project and data source changes when required

Warn when opening a project file saved with an older version of QGIS

6.7 Output

There are several ways to generate output from your QGIS session. We have discussed one already in Section *Projects* saving as a project file. Here is a sampling of other ways to produce output files:

- Menu option *Project* → Save as Image opens a file dialog where you select the name, path and type of image (PNG or JPG format). A world file with extension PNGW or JPGW saved in the same folder georeferences the image.
- Menu option *Project* → *New Print Composer* opens a dialog where you can layout and print the current map canvas (see Section *Print Composer*).

QGIS GUI

When QGIS starts, you are presented with the GUI as shown below (the numbers 1 through 5 in yellow circles refer to the five major areas of the interface as discussed below):

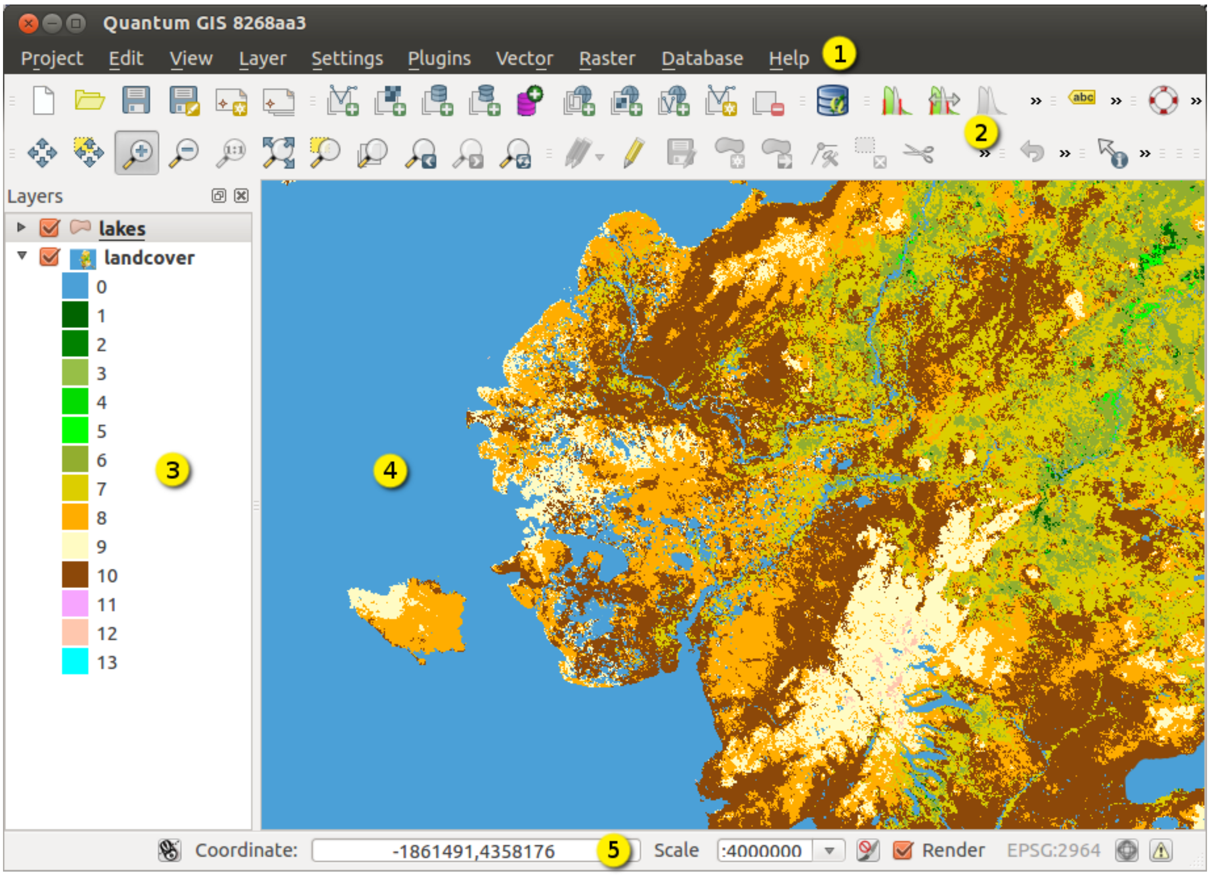


Figure 7.1: QGIS GUI with Alaska sample data

Note: Your window decorations (title bar, etc.) may appear different depending on your operating system and window manager.

The QGIS GUI is divided into five areas:

1. Menu Bar
2. Tool Bar
3. Map Legend

4. Map View
5. Status Bar

These five components of the QGIS interface are described in more detail in the following sections. Two more sections present keyboard shortcuts and context help.

7.1 Menu Bar

The menu bar provides access to various QGIS features using a standard hierarchical menu. The top-level menus and a summary of some of the menu options are listed below, together with the icons of the corresponding tools as they appear on the toolbar, as well as keyboard shortcuts. Keyboard shortcuts can also be configured manually (shortcuts presented in this section are the defaults), using the [**Configure Shortcuts**] tool under *Settings*.

Although most menu options have a corresponding tool and vice-versa, the menus are not organized quiet like the toolbars. The toolbar containing the tool is listed after each menu option as a checkbox entry. Some menu options only appear if the corresponding plugin is loaded. For more information about tools and toolbars, see Section *Toolbar*.

7.1.1 Project

Menu Option	Shortcut	Reference	Toolbar
New	`Ctrl+N`	see *Projects*	*Project*
Open	`Ctrl+O`	see *Projects*	*Project*
New from template →		see *Projects*	*Project*
Open Recent →		see *Projects*	
Save	`Ctrl+S`	see *Projects*	*Project*
Save As	`Ctrl+Shift+S`	see *Projects*	*Project*
Save as Image		see *Output*	
New Print Composer	`Ctrl+P`	see *Print Composer*	*Project*
Composer manager ...		see *Print Composer*	*Project*
Print Composers →		see *Print Composer*	
Exit |qg|	`Ctrl+Q`		

7.1.2 Edit

Menu Option	Shortcut	Reference	Toolbar
Undo	`Ctrl+Z`	see *Advanced digitizing*	*Advanced Digitizing*
Redo	`Ctrl+Shift+Z`	see *Advanced digitizing*	*Advanced Digitizing*
Cut Features	`Ctrl+X`	see *Digitizing an existing layer*	*Digitizing*
Copy Features	`Ctrl+C`	see *Digitizing an existing layer*	*Digitizing*
Paste Features	`Ctrl+V`	see *Digitizing an existing layer*	*Digitizing*
Add Feature	`Ctrl+.`	see *Digitizing an existing layer*	*Digitizing*
Move Feature(s)		see *Digitizing an existing layer*	*Digitizing*
Delete Selected		see *Digitizing an existing layer*	*Digitizing*
Rotate Feature(s)		see *Advanced digitizing*	*Advanced Digitizing*
Simplify Feature		see *Advanced digitizing*	*Advanced Digitizing*
Add Ring		see *Advanced digitizing*	*Advanced Digitizing*
Add Part		see *Advanced digitizing*	*Advanced Digitizing*
Delete Ring		see *Advanced digitizing*	*Advanced Digitizing*
Delete Part		see *Advanced digitizing*	*Advanced Digitizing*
Reshape Features		see *Advanced digitizing*	*Advanced Digitizing*
Offset Curves		see *Advanced digitizing*	*Advanced Digitizing*
Split Features		see *Advanced digitizing*	*Advanced Digitizing*
Merge Selected Features		see *Advanced digitizing*	*Advanced Digitizing*
Merge Attr. of Selected Features		see *Advanced digitizing*	*Advanced Digitizing*
Node Tool		see *Digitizing an existing layer*	*Digitizing*
Rotate Point Symbols		see *Advanced digitizing*	*Advanced Digitizing*

After activating Toggle editing mode for a layer, you will find the `Add Feature` icon in the *Edit* menu depending on the layer type (point, line or polygon).

7.1.3 Edit (extra)

Menu Option	Shortcut	Reference	Toolbar
Add Feature		see *Digitizing an existing layer*	*Digitizing*
Add Feature		see *Digitizing an existing layer*	*Digitizing*
Add Feature		see *Digitizing an existing layer*	*Digitizing*

7.1.4 View

Menu Option	Shortcut	Reference	Toolbar
Pan Map			*Map Navigation*
Pan Map to Selection			*Map Navigation*
Zoom In	`Ctrl++`		*Map Navigation*
Zoom Out	`Ctrl+-`		*Map Navigation*
Select →		see *Select and deselect features*	*Attributes*
Identify Features	`Ctrl+Shift+I`		*Attributes*
Measure →		see *Measuring*	*Attributes*
Zoom Full	`Ctrl+Shift+F`		*Map Navigation*
Zoom To Layer			*Map Navigation*
Zoom To Selection	`Ctrl+J`		*Map Navigation*
Zoom Last			*Map Navigation*
Zoom Next			*Map Navigation*
Zoom Actual Size			*Map Navigation*
Decorations →		see *Decorations*	
Map Tips			*Attributes*
New Bookmark	`Ctrl+B`	see *Spatial Bookmarks*	*Attributes*
Show Bookmarks	`Ctrl+Shift+B`	see *Spatial Bookmarks*	*Attributes*
Refresh	`Ctrl+R`		*Map Navigation*

7.1.5 Layer

Menu Option	Shortcut	Reference	Toolbar
New →		see *Creating new Vector layers*	*Manage Layers*
Embed Layers and Groups ...		see *Nesting Projects*	
Add Vector Layer	`Ctrl+Shift+V`	see *Working with Vector Data*	*Manage Layers*
Add Raster Layer	`Ctrl+Shift+R`	see *Loading raster data in QGIS*	*Manage Layers*
Add PostGIS Layer	`Ctrl+Shift+D`	see *PostGIS Layers*	*Manage Layers*
Add SpatiaLite Layer	`Ctrl+Shift+L`	see *SpatiaLite Layers*	*Manage Layers*
Add MSSQL Spatial Layer	`Ctrl+Shift+M`	see label_mssql	*Manage Layers*
Add Oracle GeoRaster Layer		see *Oracle GeoRaster Plugin*	*Manage Layers*
Add SQL Anywhere Layer		see *SQL Anywhere Plugin*	*Manage Layers*
Add WMS/WMTS Layer	`Ctrl+Shift+W`	see *WMS/WMTS Client*	*Manage Layers*
Add WCS Layer		see *WCS Client*	*Manage Layers*
Add WFS Layer		see *WFS and WFS-T Client*	*Manage Layers*
Add Delimited Text Layer		see *Add Delimited Text Layer*	*Manage Layers*
Copy style		see *Style Menu*	
Paste style		see *Style Menu*	

Continued on next page

Table 7.1 – continued from previous page

Menu Option	Shortcut	Reference	Toolbar
Open Attribute Table		see *Working with the Attribute Table*	*Attributes*
Toggle Editing		see *Digitizing an existing layer*	*Digitizing*
Save Layer Edits		see *Digitizing an existing layer*	*Digitizing*
Current Edits →		see *Digitizing an existing layer*	*Digitizing*
Save as...			
Save selection as vector file...		See *Working with the Attribute Table*	
Remove Layer(s)	`Ctrl+D`		
Set CRS of Layer(s)	`Ctrl+Shift+C`		
Set project CRS from Layer			
Properties			
Query...			
Labeling			
Add to Overview	`Ctrl+Shift+O`		*Manage Layers*
Add All To Overview			
Remove All From Overview			
Show All Layers	`Ctrl+Shift+U`		*Manage Layers*
Hide All Layers	`Ctrl+Shift+H`		*Manage Layers*

7.1.6 Settings

Menu Option	Shortcut	Reference	Toolbar
Panels →		see *Panels and Toolbars*	
Toolbars →		see *Panels and Toolbars*	
Toggle Full Screen Mode	`Ctrl-F`		
Project Properties ...	`Ctrl+Shift+P`	see *Projects*	
Custom CRS ...		see *Custom Coordinate Reference System*	
Style Manager...		see *vector_style_manager*	
Configure shortcuts ...			
Customization ...		see *Customization*	
Options ...		see *Options*	
Snapping Options ...			

7.1.7 Plugins

Menu Option	Shortcut	Reference	Toolbar
Manage and Install Plugins		see *Managing Plugins*	
Python Console			
GRASS →		see *GRASS GIS Integration*	*GRASS*

When starting QGIS for the first time not all core plugins are loaded.

7.1.8 Vector

Menu Option	Shortcut	Reference	Toolbar
Coordinate Capture →		see *Coordinate Capture Plugin*	*Vector*
Dxf2Shp →		see *Dxf2Shp Converter Plugin*	*Vector*
GPS →		see *GPS Plugin*	*Vector*
Open Street Map →		see *Loading OpenStreetMap Vectors*	
Road Graph →		see *Road Graph Plugin*	
Spatial Query →		see *Spatial Query Plugin*	*Vector*

When starting QGIS for the first time not all core plugins are loaded.

7.1.9 Raster

Menu Option	Shortcut	Reference	Toolbar
Raster calculator		see *Raster Calculator*	
Georeferencer →		see *Georeferencer Plugin*	*Raster*
Heatmap →		see *Heatmap Plugin*	*Raster*
Interpolation →		see *Interpolation Plugin*	*Raster*
Zonal Statistics →		see *Zonal Statistics Plugin*	*Raster*

When starting QGIS for the first time not all core plugins are loaded.

7.1.10 Database

Menu Option	Shortcut	Reference	Toolbar
eVis →		see *eVis Plugin*	*Database*
Spit →		see *label_spit*	*Database*

When starting QGIS for the first time not all core plugins are loaded.

7.1.11 Processing

Menu Option	Shortcut	Reference	Toolbar
Toolbox		see *The toolbox*	*Toolbox*
Graphical Modeler		see *The graphical modeler*	
History and Logs		see *The history manager*	
Options and configuration		see *Configuring the processing framework*	
Results viewer		see *Configuring external applications*	
Commander	`Ctrl+Alt+M`	see *The SEXTANTE Commander*	

When starting QGIS for the first time not all core plugins are loaded.

7.1.12 Help

Menu Option	Shortcut	Reference	Toolbar
Help Contents	`F1`		*Help*
What's This?	`Shift+F1`		*Help*
API Documentation			
Need support ?			
|qg| *Home Page*	`Ctrl+H`		
Check |qg| Version			
About			
|qg| *Sponsors*			

Please note that for Linux the Menu Bar items listed above are the default ones in KDE window manager. In GNOME, Settings menu has different content and its items have to be found here:

Project Properties	*Project*
Options	*Edit*
Configure Shortcuts	*Edit*
Style Manager	*Edit*
Custom CRS	*Edit*
Panels →	*View*
Toolbars →	*View*
Toggle Full Screen Mode	*View*
Tile scale slider	*View*
Live GPS tracking	*View*

7.2 Toolbar

The toolbar provides access to most of the same functions as the menus, plus additional tools for interacting with the map. Each toolbar item has popup help available. Hold your mouse over the item and a short description of the tool's purpose will be displayed.

Every menubar can be moved around according to your needs. Additionally every menubar can be switched off using your right mouse button context menu holding the mouse over the toolbars (read also *Panels and Toolbars*).

Tip: Restoring toolbars

If you have accidentally hidden all your toolbars, you can get them back by choosing menu option *Settings* → *Toolbars* →. If a toolbar disappears under Windows, which seems to be a problem in QGIS from time to time, you have to remove `\HKEY_CURRENT_USER\Software\QGIS\qgis\UI\state` in the registry. When you restart QGIS, the key is written again with the default state, and all toolbars are visible again.

7.3 Map Legend

The map legend area lists all the layers in the project. The checkbox in each legend entry can be used to show or hide the layer.

A layer can be selected and dragged up or down in the legend to change the z-ordering. Z-ordering means that layers listed nearer the top of the legend are drawn over layers listed lower down in the legend.

Note: This behaviours can be overridden by 'Layer order' panel.

Layers in the legend window can be organised into groups. There are two ways to do so:

1. Right click in the legend window and choose *Add Group*. Type in a name for the group and press `Enter`. Now click on an existing layer and drag it onto the group.
2. Select some layers, right click in the legend window and choose *Group Selected*. The selected layers will automatically be placed in a new group.

To bring a layer out of a group you can drag it out, or right click on it and choose *Make to toplevel item*. Groups can be nested inside other groups.

The checkbox for a group will show or hide all the layers in the group with one click.

The content of the right mouse button context menu depends on whether the selected legend item is a raster or a vector layer. For GRASS vector layers Toggle editing is not available. See section *Digitizing and editing a GRASS vector layer* for information on editing GRASS vector layers.

Right mouse button menu for raster layers

- *Zoom to layer extent*
- *Zoom to Best Scale (100%)*
- *Stretch Using Current Extent*
- *Show in overview*
- *Remove*
- *Duplicate*
- *Set Layer CRS*
- *Set Project CRS from Layer*
- *Save as ...*
- *Properties*
- *Rename*
- *Copy Style*
- *Add New Group*
- *Expand all*
- *Collapse all*
- *Update Drawing Order*

Additionally, according to layer position and selection

- *Make to toplevel item*
- *Group Selected*

Right mouse button menu for vector layers

- *Zoom to Layer Extent*
- *Show in Overview*
- *Remove*
- *Duplicate*
- *Set Layer CRS*
- *Set Project CRS from Layer*
- *Open Attribute Table*

- *Toggle Editing* (not available for GRASS layers)
- *Save As ...*
- *Save Selection As*
- *Filter*
- *Show Feature Count*
- *Properties*
- *Rename*
- *Copy Style*
- *Add New Group*
- *Expand all*
- *Collapse all*
- *Update Drawing Order*

Additionally, according to layer position and selection

- *Make to toplevel item*
- *Group Selected*

Right mouse button menu for layer groups

- *Zoom to Group*
- *Remove*
- *Set Group CRS*
- *Rename*
- *Add New Group*
- *Expand all*
- *Collapse all*
- *Update Drawing Order*

It is possible to select more than one layer or group at the same time by holding down the `Ctrl` key while selecting the layers with the left mouse button. You can then move all selected layers to a new group at the same time.

You are also able to delete more than one Layer or Group at once by selecting several Layers with the `Ctrl` key and pressing `Ctrl+D` afterwards. This way all selected Layers or groups will be removed from the layer's list.

7.3.1 Working with the Legend independent layer order

There is a widget that allows to define a legend independent drawing order. You can activate it in the menu *Settings* → *Panels* → *Layer order*. Determine the drawing order of the layers in the map view here. Doing so makes it possible to order your layers in order of importance, for example, but to still display them in the correct order (see figure_layer_order). Checking the *Control rendering order* box underneath the list of layers will cause a revert to default behavior.

7.4 Map View

This is the "business end" of QGIS - maps are displayed in this area! The map displayed in this window will depend on the vector and raster layers you have chosen to load (see sections that follow for more information on how to load layers). The map view can be panned (shifting the focus of the map display to another region) and zoomed in and out. Various other operations can be performed on the map as described in the toolbar description

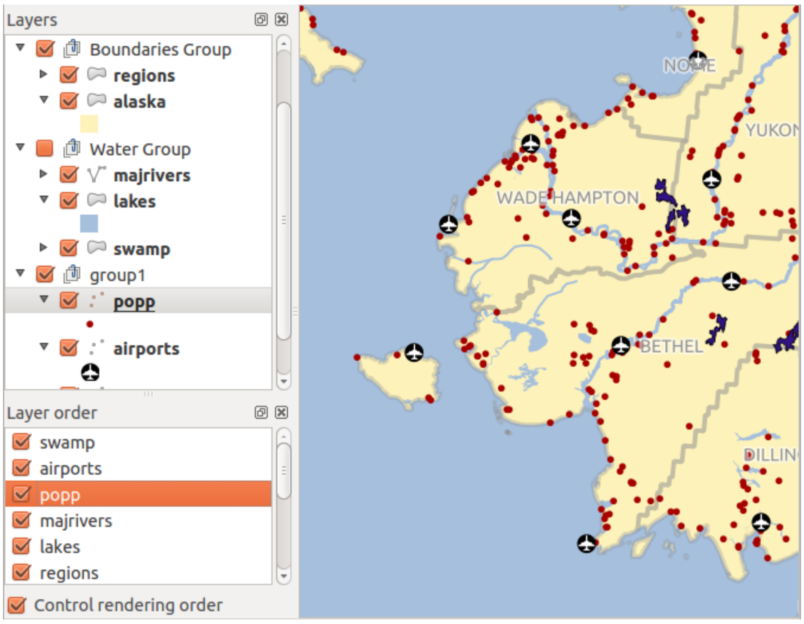


Figure 7.2: Define a legend independent layer order

above. The map view and the legend are tightly bound to each other - the maps in view reflect changes you make in the legend area.

Tip: Zooming the Map with the Mouse Wheel

You can use the mouse wheel to zoom in and out on the map. Place the mouse cursor inside the map area and roll the wheel forward (away from you) to zoom in and backwards (towards you) to zoom out. The mouse cursor position is the center where the zoom occurs. You can customize the behavior of the mouse wheel zoom using the *Map tools* menu under the *Settings* → *Options* menu.

Tip: Panning the Map with the Arrow Keys and Space Bar

You can use the arrow keys to pan in the map. Place the mouse cursor inside the map area and click on the right arrow key to pan East, left arrow key to pan West, up arrow key to pan North and down arrow key to pan South. You can also pan the map using the space bar or the click on mouse wheel: just move the mouse while holding down space bar or click on mouse wheel.

7.5 Status Bar

The status bar shows you your current position in map coordinates (e.g. meters or decimal degrees) as the mouse pointer is moved across the map view. To the left of the coordinate display in the status bar is a small button that will toggle between showing coordinate position or the view extents of the map view as you pan and zoom in and out.

Next to the coordinate display you find the scale display. It shows the scale of the map view. If you zoom in or out QGIS shows you the current scale. There is a scale selector which allows you to choose between predefined scales from 1:500 until 1:1000000.

A progress bar in the status bar shows progress of rendering as each layer is drawn to the map view. In some cases, such as the gathering of statistics in raster layers, the progress bar will be used to show the status of lengthy operations.

If a new plugin or a plugin update is available, you will see a message at the far left of the status bar. On the right side of the status bar is a small checkbox which can be used to temporarily prevent layers being rendered to the map view (see Section *Rendering* below). The icon immediately stops the current map rendering process.

To the right of the render functions you find the EPSG code of the current project CRS and a projector icon. Clicking on this opens the projection properties for the current project.

Tip: Calculating the correct Scale of your Map Canvas

When you start QGIS, degrees is the default unit, and it tells QGIS that any coordinate in your layer is in degrees. To get correct scale values, you can either change this to meter manually in the *General* tab under *Settings* → *Project Properties* or you can select a project Coordinate Reference System (CRS) clicking on the CRS status icon in the lower right-hand corner of the statusbar. In the last case, the units are set to what the project projection specifies, e.g. '+units=m'.

CHAPTER 8

General Tools

8.1 Identify features

Identify features allow to interact with map canvas to get data attribut on a pop-up windows. To identify feature use *View* → *Identify features* or Ctrl+Shift+I, or click on the Identify features icon in the toolbar.

If you click on several feature, this pop-up will list all data attributes of all features. The first item is the number of the item in the list of result followed by layer name. Then its first child will be the name of a field with its value. Finally all informations of the feature is displayed.

This window can be customized to display custom fields but by default it will display three kind of information:

- Actions: actions can be added to the identify feature windows. When clicking on the action label, action will be run. By default only one action is added to View feature form for editing.
- Derived: those informations are calculated or derived from other information. You can find clicked coordinate, X and Y coordinates, area in map unit and perimeter in map unit for polygon, length in map unit for line and feature id.
- Data attributes: this is the list of attribute fields from data.

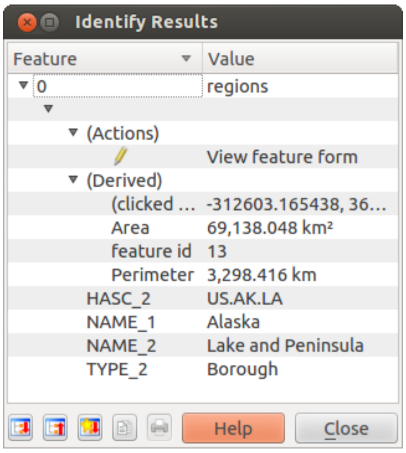


Figure 8.1: Identify feaures dialog (Gnome)

At the bottom of the windows, you have five icons:

- Expand tree
- Collapse tree
- Default behaviour

- Copy attributes
- Print selected HTML response

More feature can be found in the menu display with a right click of the mouse somewhere in the response tree.

This menu allows to:

- View Feature form
- Zoom to feature
- Copy feature: copy all feature ie geometry and attributes;
- Copy attribute value: copy only the value of the attribut you click on;
- Copy feature attributes: copy only attributes;
- Clear result: result in the window are removed
- Clear highlights: features highlight on the map are removed
- Highlight all
- Highlight layer
- Layer properties: open layer properties window
- Expand all
- Collapse all

8.2 Keyboard shortcuts

QGIS provides default keyboard shortcuts for many features. You find them in Section *Menu Bar*. Additionally the menu option *Settings* → *Configure Shortcuts* allows to change the default keyboard shortcuts and to add new keyboard shortcuts to QGIS features.

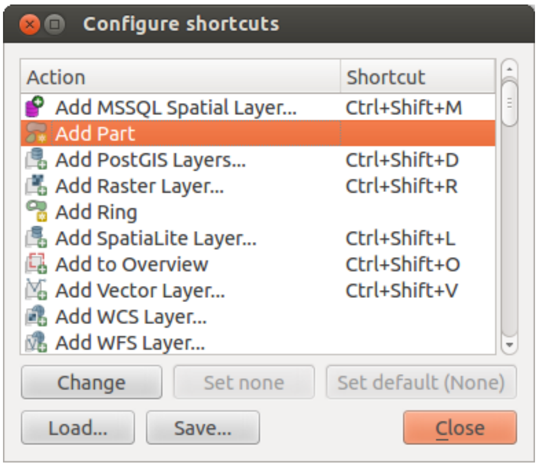


Figure 8.2: Define shortcut options (Gnome)

Configuration is very simple. Just select a feature from the list and click on **[Change]**, **[Set none]** or **[Set default]**. Once you have found your configuration, you can save it as XML file and load it to another QGIS installation.

8.3 Context help

When you need help on a specific topic, you can access context help via the *Help* button available in most dialogs - please note that third-party plugins can point to dedicated web pages.

8.4 Rendering

By default, QGIS renders all visible layers whenever the map canvas must be refreshed. The events that trigger a refresh of the map canvas include:

- Adding a layer
- Panning or zooming
- Resizing the QGIS window
- Changing the visibility of a layer or layers

QGIS allows you to control the rendering process in a number of ways.

8.4.1 Scale Dependent Rendering

Scale dependent rendering allows you to specify the minimum and maximum scales at which a layer will be visible. To set scale dependency rendering, open the *Properties* dialog by double-clicking on the layer in the legend. On the *General* tab, click on the *Use scale dependent rendering* checkbox to activate the feature then set the minimum and maximum scale values. .

You can determine the scale values by first zooming to the level you want to use and noting the scale value in the QGIS status bar.

8.4.2 Controlling Map Rendering

Map rendering can be controlled in the following ways:

Suspending Rendering

To suspend rendering, click the *Render* checkbox in the lower right corner of the statusbar. When the *Render* checkbox is not checked, QGIS does not redraw the canvas in response to any of the events described in Section *Rendering*. Examples of when you might want to suspend rendering include:

- Add many layers and symbolize them prior to drawing
- Add one or more large layers and set scale dependency before drawing
- Add one or more large layers and zoom to a specific view before drawing
- Any combination of the above

Checking the *Render* checkbox enables rendering and causes an immediate refresh of the map canvas.

Setting Layer Add Option

You can set an option to always load new layers without drawing them. This means the layer will be added to the map, but its visibility checkbox in the legend will be unchecked by default. To set this option, choose menu option *Settings* → *Options* → and click on the *Rendering* menu. Uncheck the *By default new layers added to the map should be displayed* checkbox. Any layer added to the map will be off (invisible) by default.

Another option in *Settings* → *Options* → *Rendering* menu is the *Enable back buffer* checkbox. It provides better graphics performance at the cost of loosing the possibility to cancel rendering and incremental feature drawing. If it is unchecked, you can set the 'Number of features to draw before updating the display', otherwise it is inactive.

Finally you can activate the *Use render caching where possible to speed up redraws* checkbox.

Stopping Rendering

To stop the map drawing, press the `ESC` key. This will halt the refresh of the map canvas and leave the map partially drawn. It may take a bit of time between pressing `ESC` and the time the map drawing is halted.

Note: It is currently not possible to stop rendering - this was disabled in qt4 port because of User Interface (UI) problems and crashes.

Updating the Map Display During Rendering

You can set an option to update the map display as features are drawn. By default, QGIS does not display any features for a layer until the entire layer has been rendered. To update the display as features are read from the datastore, choose menu option *Settings* → *Options* click on the *Rendering* menu. Set the feature count to an appropriate value to update the display during rendering. Setting a value of 0 disables update during drawing (this is the default). Setting a value too low will result in poor performance as the map canvas is continually updated during the reading of the features. A suggested value to start with is 500.

Influence Rendering Quality

To influence the rendering quality of the map you have 2 options. Choose menu option *Settings* → *Options* click on the *Rendering* menu and select or deselect following checkboxes.

- *Make lines appear less jagged at the expense of some drawing performance*
- *Fix problems with incorrectly filled polygons*

8.5 Measuring

Measuring works within projected coordinate systems (e.g., UTM) and unprojected data. If the loaded map is defined with a geographic coordinate system (latitude/longitude), the results from line or area measurements will be incorrect. To fix this you need to set an appropriate map coordinate system (See Section *Working with Projections*). All measuring modules also use the snapping settings from the digitizing module. This is useful, if you want to measure along lines or areas in vector layers.

To select a measure tool click on and select the tool you want to use.

8.5.1 Measure length, areas and angles

Measure Line: QGIS is able to measure real distances between given points according to a defined ellipsoid. To configure this, choose menu option *Settings* → *Options*, click on the *Map tools* tab and choose the appropriate ellipsoid. There you can also define a rubberband color and your preferred measurement units (meters or feet) and angle units (degrees, radians and gon). The tools then allows you to click points on the map. Each segment-length as well as the total shows up in the measure-window. To stop measuring click your right mouse button.

Measure Area: Areas can also be measured. In the measure window the accumulated area size appears. In addition, the measuring tool will snap to the currently selected layer, provided that layer has its snapping tolerance

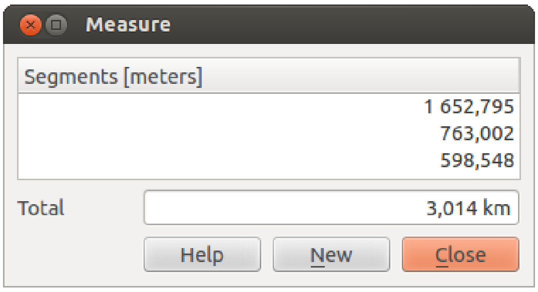


Figure 8.3: Measure Distance (Gnome)

set. (See Section *Setting the Snapping Tolerance and Search Radius*). So if you want to measure exactly along a line feature, or around a polygon feature, first set its snapping tolerance, then select the layer. Now, when using the measuring tools, each mouse click (within the tolerance setting) will snap to that layer.

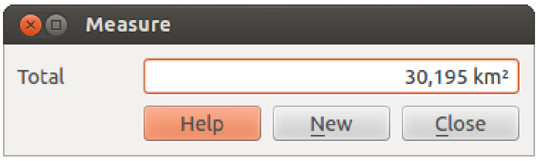


Figure 8.4: Measure Area (Gnome)

Measure Angle: You can also measure angles. The cursor becomes cross-shaped. Click to draw the first segment of the angle you wish to measure, then move the the cursor to draw the desired angle. The measure is displayed in a popup dialog.

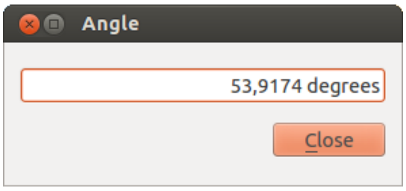


Figure 8.5: Measure Angle (Gnome)

8.5.2 Select and deselect features

The QGIS toolbar provides several tools to select features in the map canvas. To select one or several features just click on and select your tool:

- Select Single Feature
- Select Features by Rectangle
- Select Features by Polygon
- Select Features by Freehand
- Select Features by Radius

To deselect all selected features click on Deselect features from all layers.

8.6 Decorations

The Decorations of QGIS includes the Grid, Copyright Label, the North Arrow and the Scale Bar. They are used to 'decorate' the map by adding cartographic elements.

8.6.1 Grid

Grid allows to add a coordinate grid and coordinate annotations to the map canvas.

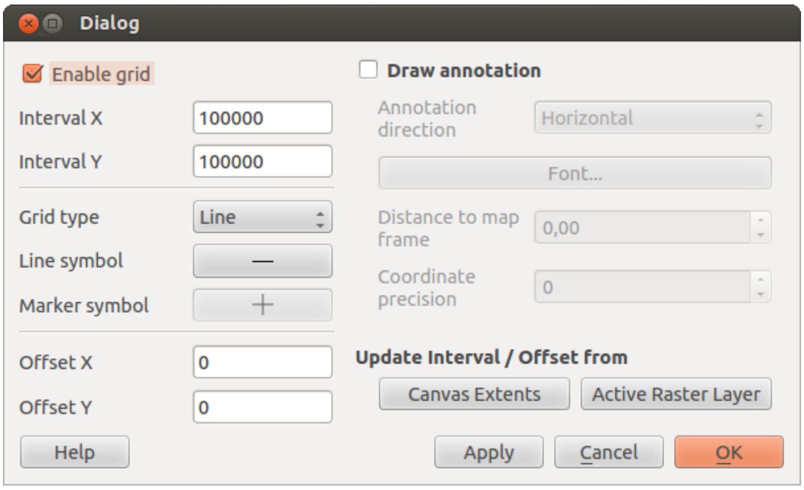


Figure 8.6: The Grid Dialog

1. Select from menu *View* → *Decorations* → *Grid*. The dialog starts (see figure_decorations_1).
2. Activate the *Enable grid* checkbox and set grid definitions according to the layers loaded in the map canvas.
3. Activate the *Draw annotations* checkbox and set annotation definitions according to the layers loaded in the map canvas.
4. Click **[Apply]** to check, if it looks as expected.
5. Click **[OK]** to close the dialog.

8.6.2 Copyright Label

Copyright label adds a Copyright label using the text you prefer to the map.

1. Select from menu *View* → *Decorations* → *Copyright Label*. The dialog starts (see figure_decorations_2).
2. Enter the text you want to place on the map. You can use HTML as shown in the example
3. Choose the placement of the label from the *Placement* 'Bottom Right' drop-down box
4. Make sure the *Enable Copyright Label* checkbox is checked
5. Click **[OK]**

In the example above (default) QGIS places a copyright symbol followed by the date in the lower right hand corner of the map canvas.

Figure 8.7: The copyright Dialog

8.6.3 North Arrow

North Arrow places a simple north arrow on the map canvas. At present there is only one style available. You can adjust the angle of the arrow or let QGIS set the direction automatically. If you choose to let QGIS determine the direction, it makes its best guess as to how the arrow should be oriented. For placement of the arrow you have four options, corresponding to the four corners of the map canvas.

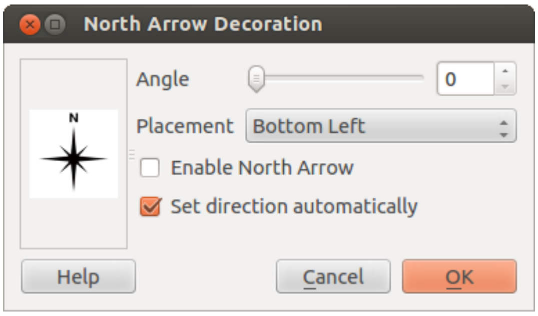


Figure 8.8: The North Arrow Dialog

8.6.4 Scale Bar

Scale Bar adds a simple scale bar to the map canvas. You control the style and placement, as well as the labeling of the bar.

QGIS only supports displaying the scale in the same units as your map frame. So if the units of your layers are in meters, you can't create a scale bar in feet. Likewise if you are using decimal degrees, you can't create a scale bar to display distance in meters.

To add a scale bar:

1. Select from menu *View* → *Decorations* → *Scale Bar* The dialog starts (see figure_decorations_4)
2. Choose the placement from the *Placement* 'Bottom Left' drop-down list
3. Choose the style from the *Scale bar style* 'Tick Down' list
4. Select the color for the bar *Color of bar* 'black' or use the default black color
5. Set the size of the bar and its label *Size of bar* '30 degrees'
6. Make sure the *Enable scale bar* checkbox is checked

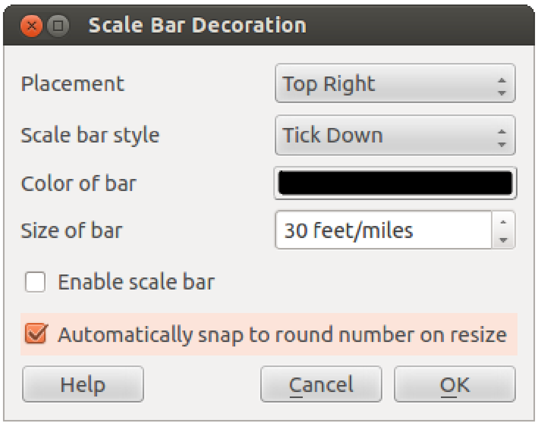


Figure 8.9: The Scale Bar Dialog

7. Optionally choose to automatically snap to a round number when the canvas is resized *Automatically snap to round number on resize*
8. Click **[OK]**

Tip: Settings of Decorations

When you save a .qgs project, any changes you have made to Grid, NorthArrow, ScaleBar and Copyright will be saved in the project and restored the next time you load the project.

8.7 Annotation Tools

The Text Annotation tools in the attribute toolbar provides the possibility to place formatted text in a balloon on the QGIS map canvas. Use the *Text Annotation* tool and click into the map canvas.

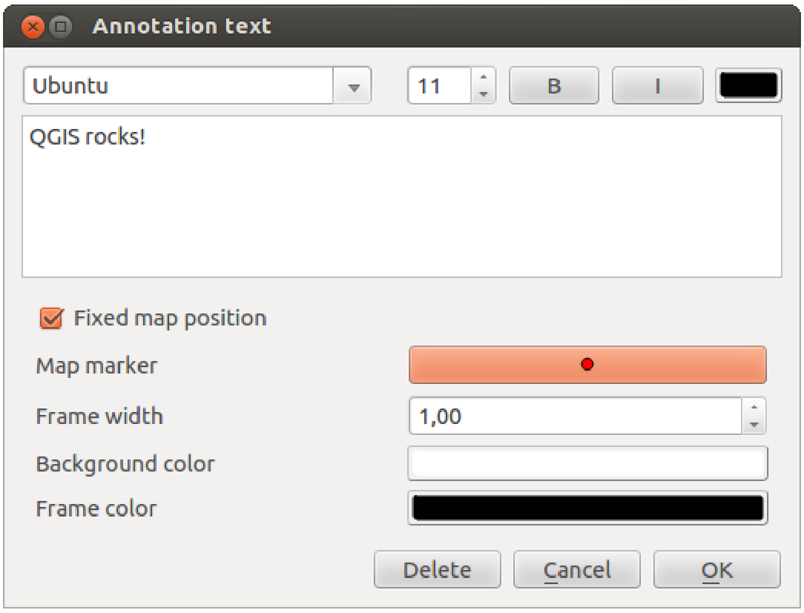


Figure 8.10: Annotation text dialog

Double click on the item opens a dialog with various options. There is the text editor to enter the formatted text and other item settings. E.g. there is the choice of having the item placed on a map position (displayed by a marker symbol) or to have the item on a screen position (not related to the map). The item can be moved by map position (drag the map marker) or by moving only the balloon. The icons are part of GIS theme, and are used by default in the other themes too.

The Move Annotation tool allows to move the annotation on the map canvas.

8.7.1 Html annotations

The Html Annotation tools in the attribute toolbar provides the possibility to place the content of a html file in a balloon on the QGIS map canvas. Use the *Html Annotation* tool, click into the map canvas and add the path to the html file into the dialog.

8.7.2 SVG annotations

The SVG Annotation tools in the attribute toolbar provides the possibility to place a SVG Symbol in a balloon on the QGIS map canvas. Use the *SVG Annotation* tool, click into the map canvas and add the path to the SVG file into the dialog.

8.7.3 Form annotations

Additionally you can also create your own annotation forms. The Form Annotation tool is useful to display attributes of a vector layer in a customized qt designer form (see figure_custom_annotation). It is similar to the designer forms for the *Identify features* tool, but displayed in an annotation item. Also see QGIS blog http://blog.qgis.org/node/143 for more information.

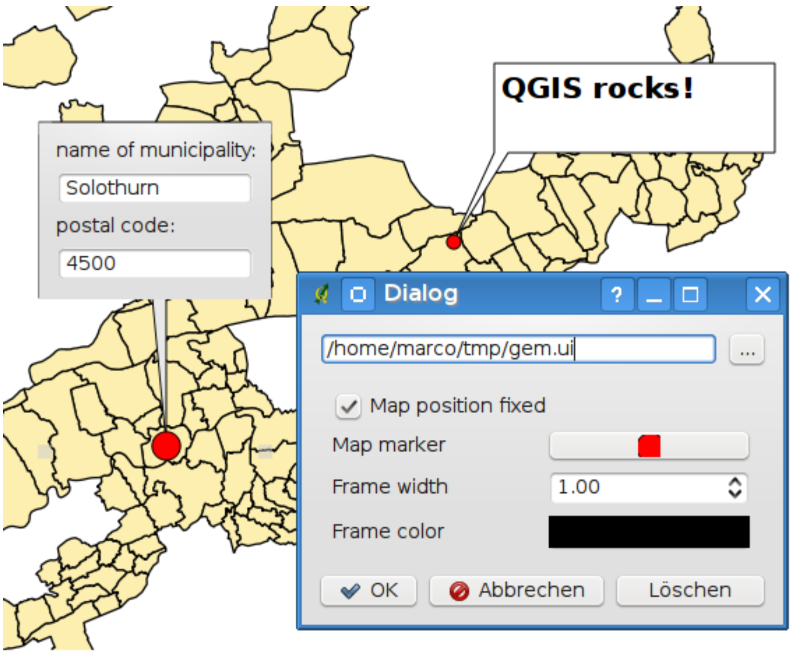


Figure 8.11: Customized qt designer annotation form

Note: If you press `Ctrl+T` while an *Annotation* tool is active (move annotation, text annotation, form annotation), the visibility states of the items are inverted.

8.8 Spatial Bookmarks

Spatial Bookmarks allow you to "bookmark" a geographic location and return to it later.

8.8.1 Creating a Bookmark

To create a bookmark:

1. Zoom or pan to the area of interest.
2. Select the menu option *View* → *New Bookmark* or press `Ctrl-B`.
3. Enter a descriptive name for the bookmark (up to 255 characters).
4. Press `Enter` to add the bookmark or **[Delete]** to remove the bookmark.

Note that you can have multiple bookmarks with the same name.

8.8.2 Working with Bookmarks

To use or manage bookmarks, select the menu option *View* → *Show Bookmarks*. The *Geospatial Bookmarks* dialog allows you to zoom to or delete a bookmark. You can not edit the bookmark name or coordinates.

8.8.3 Zooming to a Bookmark

From the *Geospatial Bookmarks* dialog, select the desired bookmark by clicking on it, then click **[Zoom To]**. You can also zoom to a bookmark by double-clicking on it.

8.8.4 Deleting a Bookmark

To delete a bookmark from the *Geospatial Bookmarks* dialog, click on it then click **[Delete]**. Confirm your choice by clicking **[Yes]** or cancel the delete by clicking **[No]**.

8.9 Nesting Projects

If you want to embed content from other project files into your project you can choose *Layer* → *Embed Layers and Groups*.

8.9.1 Embedding layers

The following dialog allows you to embed layers from other projects:

1. Press ... to look for another project from the Alaska dataset.
2. Select the project file grassland. You can see the content of the project (see figure_embed_dialog).
3. Press `Ctrl` and klick on the layers grassland and regions. The layers are embedded in the map legend and the map view now.

While the embedded layers are editable you can't change it's properties like Style and Labeling.

Removing embedded layers

Right-click on the embedded layer and choose Remove.

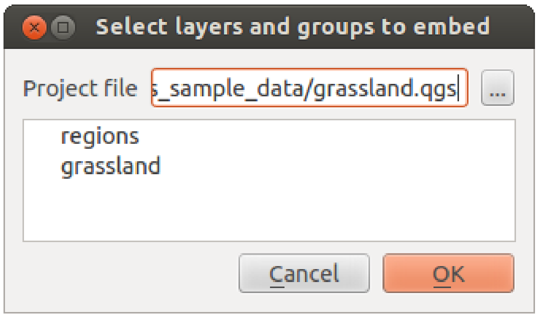


Figure 8.12: Select layers and groups to embed

8.10 Add Delimited Text Layer

This function allows you to load a delimited text file as a layer in QGIS. Following settings need to be defined:

1. The **File format** usually is *CSV (comma separated values)*. If another delimiter is used, activate the *custom delimiter* radiobutton and if each line in the file is split using a regular expression, please activate the *Regular expression delimiter* radiobutton.
2. As **Record options** a text file usually provides a delimited header row of field names. This is usually the first line in the text file. If there is no header row available, deactivate the *first records have field names* checkbox. And if the header row isn't the first line of the text file, define the number of header lines to discard.
3. As **Field options** you can trim leading and trailing spaces from fields activating the *Trim fields* checkbox. You can *Discard empty fields* in each record and you can define that the *Decimal separator is comma*. Otherwise it will be point.
4. As **Geometry definitions** a typical text file provides *Point coordinates*. This means there must be an 'X' and 'Y' field with coordinate values. If the text file provides a *Well Known Text* field, there must be a 'WKT' field with geometry information for point, line or polygon objects. These fields can have any name. Otherwise for attribute tables define *no geometry*. The x and y coordinates must be specified as a number. The coordinate system is not important. If they are defined in degree/minutes/seconds, activate the *DMS coordinates* checkbox.
5. As **Layer settings** you can activate *Use spatial index* to improve performance of displaying and spatially selecting features. You can define to *Use Subset index* and to *Watch file* to watch for changes to the file by other applications, while QGIS is running.

As an example of a valid text file we import the elevation point data file `elevp.csv` coming with the QGIS sample dataset (See Section *Sample Data*):

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

Some items of note about the text file are:

1. The example text file uses `;` (semicolon) as delimiter. Any character can be used to delimit the fields.
2. The first row is the header row. It contains the fields `X`, `Y` and `ELEV`.
3. No quotes (`"`) are used to delimit text fields.
4. The x coordinates are contained in the `X` field.
5. The y coordinates are contained in the `Y` field.

8.10.1 Using the function

Click the toolbar icon Add Delimited Text Layer in the *Manage layers* toolbar to open the *Create a Layer from a Delimited Text File* dialog as shown in figure_delimited_text_1.

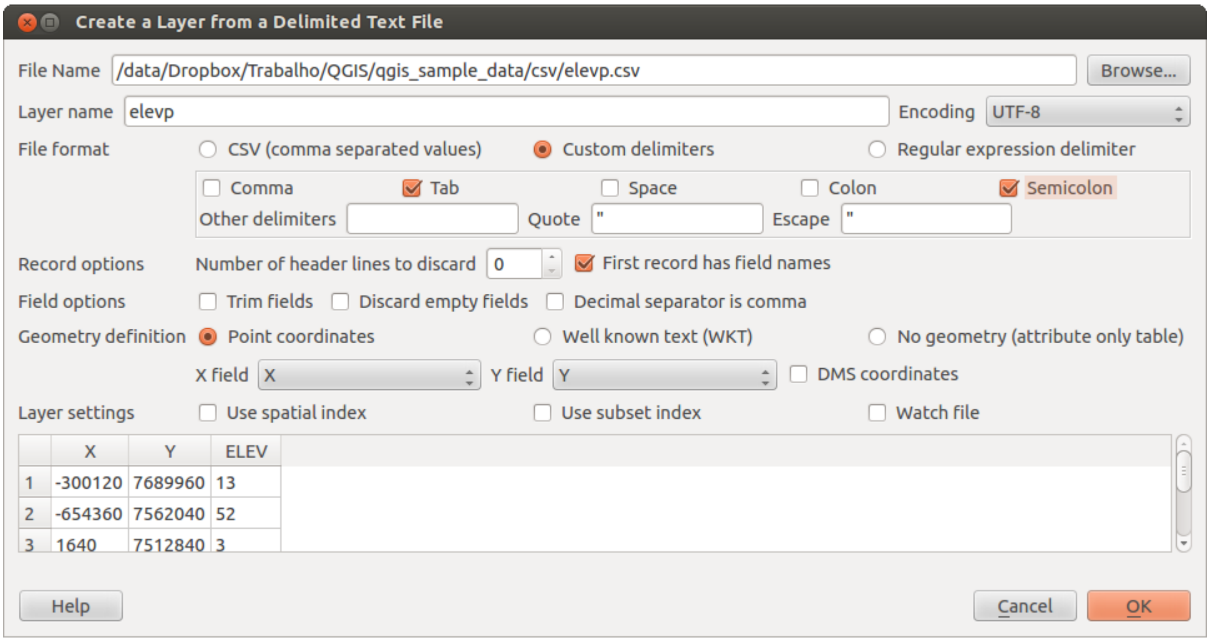


Figure 8.13: Delimited Text Dialog

First select the file (e.g., `qgis_sample_data/csv/elevp.csv`) to import by clicking on the **[Browse]** button. Once the file is selected, QGIS attempts to parse the file using the last used delimiter, in this case a semicolon (`;`). To properly parse the file, it is important to select the correct delimiter. To change the delimiter to tab use `\t` (this is a regular expression for the tab character).

Once the file is parsed, make a *Geometry definition* *Point coordinates* and choose the `X` and `Y` fields from the dropdown lists. Finally enter a Layer name (e.g., `elevp`) as shown in figure_delimited_text_1 . To add the layer to the map, click **[OK]**. The delimited text file now behaves as any other map layer in QGIS.

CHAPTER 9

QGIS Configuration

QGIS is highly configurable through the *Settings* menu. Choose between Panels, Toolbars, Project properties, Options and Customization.

9.1 Panels and Toolbars

In the *Panels*→ menu you can switch on and off QGIS widgets. The *Toolbars*→ menu provides the possibility to switch on and off icon groups in the QGIS toolbar (see figure_panels_toolbars).

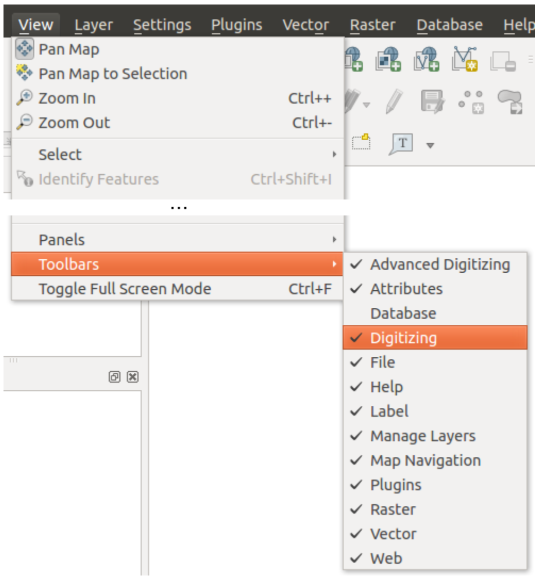


Figure 9.1: The Panels and Toolbars menu

Tip: Activating the QGIS Overview

In QGIS you can use an overview panel that provides a full extent view of layers added to it. It can be selected under the menu *View* → *Panels*. Within the view is a rectangle showing the current map extent. This allows you to quickly determine which area of the map you are currently viewing. Note that labels are not rendered to the map overview even if the layers in the map overview have been set up for labeling. If you click and drag the red rectangle in the overview that shows your current extent, the main map view will update accordingly.

Tip: Show Log Messages

It's possible to track the QGIS messages. You can activate *Log Messages* in the menu *Settings* → *Panels* and follow the messages in the *General* and *Plugin* tab during loading and operation.

9.2 Project Properties

In the properties window for the project under *Project* → *Project Properties* or *Project* → *Project Properties* you set project specific options. These include:

- In the *General* menu the project title, selection and background color, layer units, precision, and the option to save relative paths to layers can be defined. If the CRS transformation is on you can choose an ellipsoid for distance calculations. You can define the canvas units (only used when CRS transformation is disabled) and the precision of decimal places to use. You also can define a project scale list, that overrides the global predefined scales.
- The *CRS* menu enables you to choose the Coordinate Reference System for this project, and to enable on-the-fly re-projection of raster and vector layers when displaying layers from a different CRS.
- With the third *Identify layers* menu you set (or disable) which layers will respond to the identify tool. (See the Map tools paragraph from the *Options* section to enable identifying of multiple layers.)
- The menu *Default Styles* lets you control how new layers will be drawn when they do not have an existing .qml style defined. You can also set the default transparency level for new layers and whether symbols should have random colours assigned to them.
- The tab *OWS Server* allows to define information about the QGIS Server WMS and WFS Capabilities, the Extent and the CRS Restrictions.
- The menu *Macros* is to create a Python module, saved into a project.qgs file to be loaded and have specific functions run on the following project events: openProject(), saveProject() and closeProject().

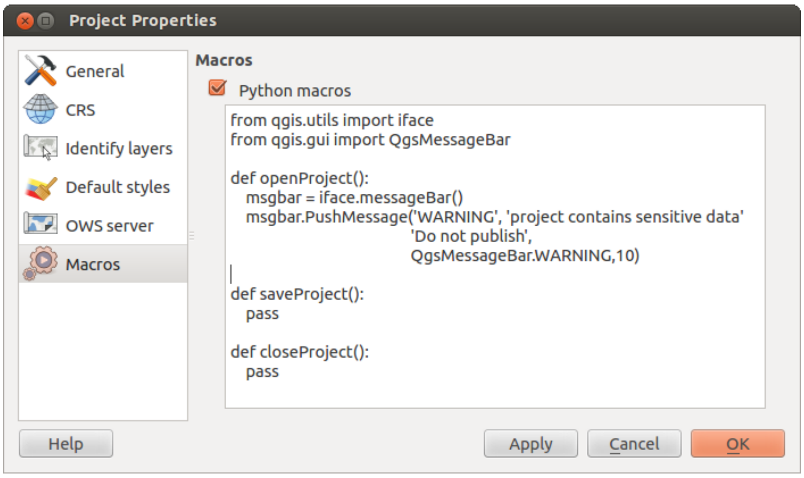


Figure 9.2: Macro settings in QGIS

9.3 Options

Some basic options for QGIS can be selected using the *Options* dialog. Select the menu option *Settings* → *Options*. The menus where you can optimize your options are:

9.3.1 General Menu

Application

- Select the *Style (QGIS restart required)* and choose between 'Oxygen','Windows','Motif','CDE', 'Plastique' and 'Cleanlux' ( ).

- Define the *Icon theme* . Currently only 'default' is possible.
- Define the *Icon size* .
- Define the *Font*. Choose between *QT default* and a user-defined font.
- Change the *Timeout for timed messages or dialogs* .
- *Hide splash screen at startup*
- *Show tips at startup*
- *Bold group box titles*
- *QGIS-styled group boxes*
- *Use live-updating color chooser dialog*

Project files

- *Open project on launch* (choose between 'New'. 'Most recent' and 'Specific'. When choosing 'Specific' use the to define a project)
- *Create new project from default project*. You have the possibility to press on *Set current project as default* or on *Reset default*. You can browse through your files and define a directory where you find your user-defined project templates. There will be an entry in *Project* → *New From Template* if you first activate *Create new project from default project* and then save a project in the project templates folder.
- *Prompt to save project and data source changes when required*
- *Warn when opening a project file saved with an older version of |qg|*
- *Enable macros* . This option was created to handle macros which are written to perform an action

on project events. You can chosse between 'Never', 'Ask', 'For this session only' and 'Always (not recommended)'.

9.3.2 System Menu

Environment

System environment variables can now be viewed and many configured in the **Environment** menu (see figure_environment_variables). This is useful for platforms, such as Mac, where a GUI application does not necessarily inherit the user's shell environment. It's also useful for setting/viewing environment variables for the external tool sets controlled by the processing toolbox, e.g. SAGA, GRASS; and, for turning on debugging output for specific sections of the source code.

- *Use custom variables (restart required - include separators)*. You can *Add* and *Remove* variables. *Current environment variables* → are displayed below and it's possible to *Show only QGIS-specific variables*.

Plugin paths

- *Add* or *Remove Path(s) to search for additional C++ plugin libraries*

9.3.3 Data Sources Menu

Feature attributes and table

- *Open attribute table in a dock window (QGIS restart required)*

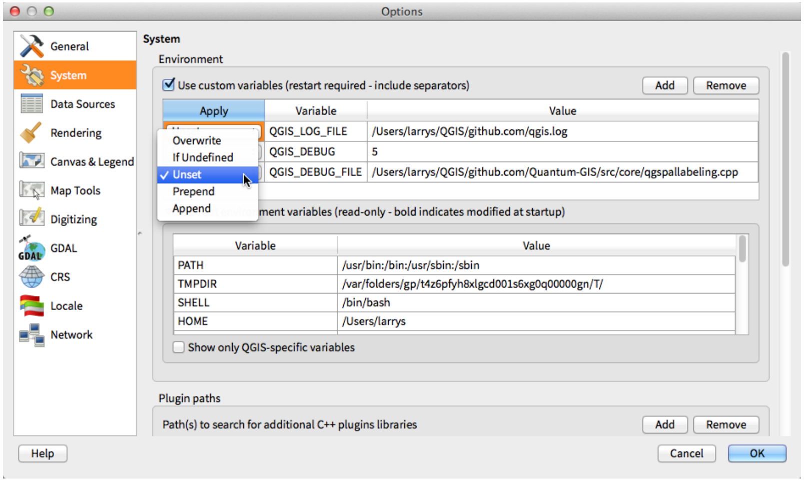


Figure 9.3: System environment variables in QGIS

- *Copy geometry in WKT representation from attribute table*. When using Copy selected rows to clipboard from the *Attribute table* menu then this has the result that also the coordinates of points or vertices are copied to the clipboard.
- *Attribute table behaviour* . There are three possibilities: ‘Show all features’, ‘Show selected features’ and ‘Show features visible on map’
- *Attribute table row cache* . This row cache makes it possible to save the last loaded x attribute rows so that working with the attribute table will be quicker. The cache will be deleted when closing the attribute table.
- *Representation for NULL values* Here you can define a value for data fields containing a NULL value.

Data source handling

- *Scan for valid items in the browser dock* . You can choose between ‘Check extension’ and ‘Check file contents’.
- *Scan for contents of compresses files (.zip) in browser dock* . ‘No’, ‘Basic scan’ and ‘Full scan’ are possible.
- *Prompt for raster sublayers when opening*. Some rasters support sublayers - they are called subdatasets in GDAL. An example is netcdf files - if there are many netcdf variables, GDAL sees every variable as a subdataset. The option is to control how to deal with sublayers when a file with sublayers is opened. You have the following choices:
  - ‘Always’: always ask (if there are existing sublayers)
  - ‘If needed’: ask if layer has no bands, but has sublayers
  - ‘Never’: never prompt, will not load anything
  - ‘Load all’: never prompt, but load all sublayers
- *Ignore shapefile encoding declaration*. If a shapefile has encoding information this will be ignored by QGIS.
- *Add PostGIS layer with double click and select in extended mode*
- *Add Oracle layers with double click and select in extended mode*

9.3.4 Rendering Menu

Rendering quality

- *Make lines appear less jagged at the expense of some drawing performance*
- *Fix problems with incorrectly filled polygons*

Rasters

- with *RGB band selection* you can define the number for the Red, Green and Blue band.

Contrast enhancement

- *Single band gray* . A single band gray can have 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax' and also 'Clip to MinMax'
- *Multi band color (byte/band)* . 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax' and 'Clip to MinMax'.
- *Multi band color (>byte/band)* . 'No stretch', 'Stretch to MinMax', 'Stretch and Clip to MinMax' and 'Clip to MinMax'.
- *Limits (minimum/maximum)* . 'Cumulative pixel count cut', 'Minimum/Maximum', 'Mean +/- standard deviation'
- *Cumulative pixel count cut limits*
- *Standard deviation multiplier*

Debugging

- *Show these events in the Log Message panel (under rendering tab):* *Map canvas refresh*

9.3.5 Canvas and Legend Menu

Default map appearance (overridden by project properties)

- Define a *Selection color* and a *Background color*.

Layer legend

- *Double click action in legend* . You can either 'Open layer properties' or 'Open attribute table' with the double click.
- the following *Legend item styles* are possible
  - *Capitalise layer names*
  - *Bold layer names*
  - *Bold group names*
  - *Display classification attribute names*
  - *Create raster icons (may be slow)*
  - *Add new layers to selected or current group*

9.3.6 Map tools Menu

Identify

- *Open identify results in a dock window (QGIS restart required)*

- The *Mode* setting determines which layers will be shown by the Identify tool. By switching to 'Top down' or 'Top down, stop at first' instead of 'Current layer' attributes for all identifiable layers (see the Project properties section under: *Projects* to set which layers are identifiable) will be shown with the Identify tool.
- *Open feature form, if a single feature is identified*
- Define *Search radius for identifying and displaying map tips as a percentage of the map width*

Measure tool

- Define *Rubberband color* for measure tools
- Define *Decimal places*
- *Keep base unit*
- *Preferred measurements units* ('meters' or 'feet')'
- *Preferred angle units* ('Angle', 'Radians' or 'Gon')

Panning and zooming

- Define *Mouse wheel action* ('Zoom', 'Zoom and recenter', 'Zoom to mouse cursor', 'Nothing')
- Define *Zoom factor* for wheel mouse

Predefined scales

Here you find a list of predefined scales. With the '+' and '-' buttons you can add or remove your individual scales.

9.3.7 Digitizing Menu

Feature creation

- *Suppress attributes pop-up windows after each created feature*
- *Reuse last entered attribute values*
- *Validate geometries*. Editing complex lines/polygons with many nodes can end up with very slow rendering. This is because the default validation procedures in QGIS can use a lot of time. To speed up rendering it is possible to select GEOS geometry validation (starting from GEOS 3.3) or to switch it off. GEOS geometry validation is much faster, but the disadvantage is that only the first geometry problem will be reported.

Rubberband

- Define Rubberband *Line width* and *Line color*

Snapping

- *Open snapping options in a dock window (QGIS restart required)*
- Define *Default snap mode* ('To vertex', 'To segment', 'To vertex and segment', 'Off')
- Define *Default snapping tolerance* in map units or pixels
- Define the *Search radius for vertex edits* in map units or pixels

Vertex markers

- *Show markers only for selected features*
- Define vertex *Marker style* ('Cross' (default), 'Semi transparent circle' or 'None')
- Define vertex *Marker size*

Curve offset tool

The next 3 options refer to the Offset Curve tool in *Advanced digitizing*. Through the various settings, it is possible to influence the shape of the line offset. These options are possible from GEOS 3.3 .

- *Join style for curve offset*
- *Quadrant segments for curve offset*
- *Miter limit for curve offset*

9.3.8 GDAL Menu

GDAL is a data exchange library for raster files. In this tab you can *Edit create options* and *Edit Pyramids Options* of the raster formats. Define which GDAL driver to be used for a raster format as in some cases more than one GDAL driver is available.

9.3.9 CRS Menu

Default CRS for new projects

- *Automatically enable 'on the fly' reprojection if layers have different CRS*
- *Enable on the fly re-projection by default*
- Select a CRS and *Always start new projects with this CRS*

CRS for new layers

This area allows to define the action, when a new layer is created, or when a layer without CRS is loaded.

- *Prompt for CRS*
- *Use project CRS*
- *Use default CRS displayed below*

9.3.10 Locale Tab

- *Overwrite system locale* and *Locale to use instead*
- Information about active system locale

9.3.11 Network Tab

General

- Define *WMS search address*, default is `http://geopole.org/wms/search?search=\%1\&type=rss`
- Define *Timeout for network requests (ms)* - default is 60000
- Define *Default expiration period for WMSC/WMTS tiles (hours)* - default is 24

Cache settings

Define the *Directory* and a *Size* for the cache.

- *Use proxy for web access* and define 'Host', 'Port', 'User', and 'Password'.
- Set the *Proxy type* according to your needs.
  - *Default Proxy*: Proxy is determined based on the application proxy set using
  - *Socks5Proxy*: Generic proxy for any kind of connection. Supports TCP, UDP, binding to a port (incoming connections) and authentication.
  - *HttpProxy*: Implemented using the "CONNECT" command, supports only outgoing TCP connections; supports authentication.

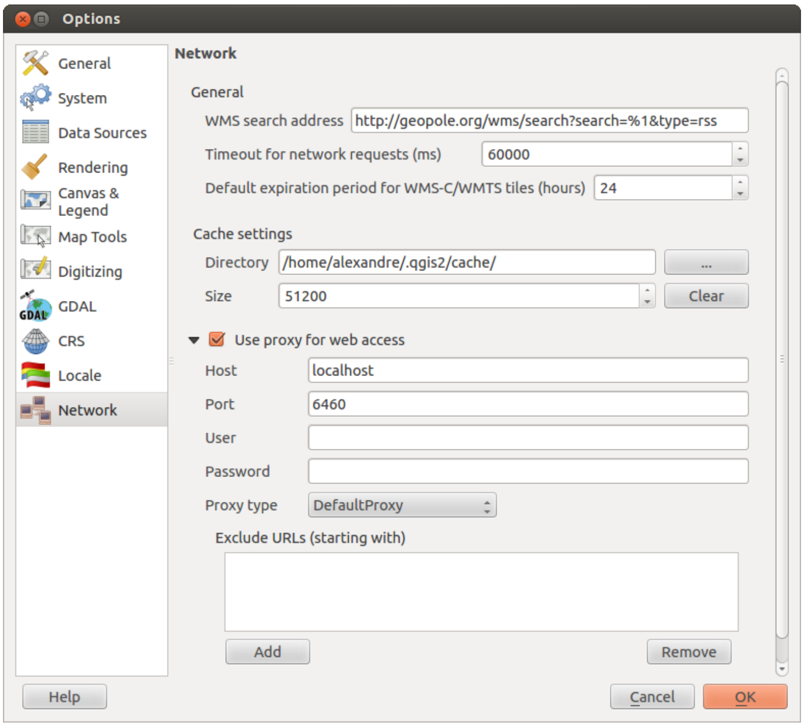


Figure 9.4: Proxy-settings in QGIS

- *HttpCachingProxy*: Implemented using normal HTTP commands, it is useful only in the context of HTTP requests
- *FtpCachingProxy*: Implemented using an FTP proxy, it is useful only in the context of FTP requests

Excluding some URLs can be added to the text box below the proxy-settings (see Figure_Network_Tab).

If you need more detailed information about the different proxy-settings, please refer to the manual of the underlying QT-library-documentation at http://doc.trolltech.com/4.5/qnetworkproxy.html#ProxyType-enum.

Tip: Using Proxies

Using proxies can sometimes be tricky. It is useful to 'trial and error' the above proxy types and check if they succeed in your case.

You can modify the options according to your needs. Some of the changes may require a restart of QGIS before they will be effective.

- settings are saved in a text file: `$HOME/.config/QGIS/qgis.conf`
- you can find your settings in: `$HOME/Library/Preferences/org.qgis.qgis.plist`
- settings are stored in the registry under: `HKEY\CURRENT_USER\Software\QGIS\qgis`

9.4 Customization

The customization tool lets you (de)activate almost every element in the QGIS user interface. This can get very useful if you have a lot of plug-ins installed that you never use and that are filling your screen.

QGIS Customization is divided into five groups. In *Menus* you can hide entries in the Menu bar. In *Panel* you find the panel windows. Panel windows are applications that can be started and used as a floating, top-level window or embedded to the QGIS main window as a docked widget (see also *Panels and Toolbars*). In the

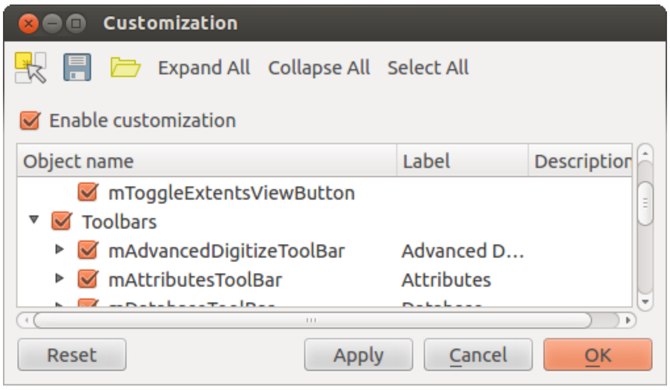


Figure 9.5: The Customization dialog

Status Bar features like the coordinate information can be deactivated. In *Toolbars* you can (de)activate the toolbar icons of QGIS and in *Widgets* you can (de)activate dialogs as well as their buttons.

With Switch to catching widgets in main application you can click on elements in QGIS you want to be hidden and find the corresponding entry in Customization (see figure_customization). You can also save your various different setups for different use cases as well. Before your changes are applied, you need to restart QGIS.

CHAPTER 10

Working with Projections

QGIS allows users to define a global and project-wide CRS (Coordinate Reference System) for layers without a pre-defined CRS. It also allows the user to define custom coordinate reference systems and supports on-the-fly (OTF) projection of vector and raster layers. All these features allow the user to display layers with different CRS and have them overlay properly.

10.1 Overview of Projection Support

QGIS has support for approximately 2,700 known CRS. Definitions for each of these CRS are stored in a SQLite database that is installed with QGIS. Normally you do not need to manipulate the database directly. In fact, doing so may cause projection support to fail. Custom CRS are stored in a user database. See Section *Custom Coordinate Reference System* for information on managing your custom coordinate reference systems.

The CRS available in QGIS are based on those defined by the European Petroleum Search Group (EPSG) and the Institut Geographique National de France (IGNF) and are largely abstracted from the spatial reference tables used in GDAL. EPSG identifiers are present in the database and can be used to specify a CRS in QGIS.

In order to use OTF projection, your data must contain information about its coordinate reference system or you have to define a global, layer or project-wide CRS. For PostGIS layers QGIS uses the spatial reference identifier that was specified when the layer was created. For data supported by OGR, QGIS relies on the presence of a recognized means of specifying the CRS. In the case of shapefiles, this means a file containing the Well Known Text (WKT) specification of the CRS. This projection file has the same base name as the shapefile and a prj extension. For example, a shapefile named `alaska.shp` would have a corresponding projection file named `alaska.prj`.

Whenever you select a new CRS, the used layer units will automatically be changed in the *General* menu of the *Project Properties* dialog under the *Project* (Gnome, OSX) or *Settings* (KDE, Windows) menu.

10.2 Global Projection Specification

QGIS starts each new project using the global default projection. The global default CRS is EPSG:4326 - WGS 84 (`proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`) and comes predefined in QGIS. This default can be changed using the **[Select...]** button in the first section, used to defining the Default Coordinate Reference System to use when starting new projects, as shown in figure_projection_1. This choice will be saved for use in subsequent QGIS sessions.

When you use layers that do not have a CRS, you need to define how QGIS responds to these layers. This can be done globally or project-wide in the *CRS* menu under *Settings* → *Options*.

The options shown in figure_projection_1 are:

- *Prompt for CRS*
- *Use project CRS*

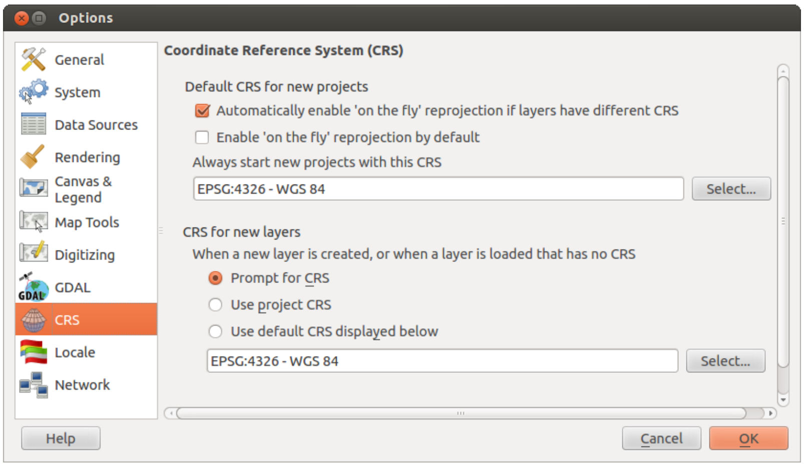


Figure 10.1: CRS tab in the QGIS Options Dialog

- *Use default CRS displayed below*

If you want to define the coordinate reference system for a certain layer without CRS information, you can also do that in the *General* menu of the raster (see *General Menu*) and vector (see *General Menu*) properties dialog. If your layer already has a CRS defined, it will be displayed as shown in *Vector Layer Properties Dialog* .

Tip: CRS in the Map Legend

Right-clicking on a layer in the Map Legend (Section *Map Legend*) provides two CRS short cuts. *Set layer CRS* takes you directly to the Coordinate Reference System Selector dialog (see figure_projection_2). *Set project CRS from Layer* redefines the project CRS using the layer's CRS

10.3 Define On The Fly (OTF) Reprojection

QGIS supports OTF reprojection for both raster and vector data. However, OTF is not activated by default. To use OTF projection, you must activate the *Enable on the fly CRS transformation* checkbox in the *CRS* menu of the *Project Properties* dialog.

There are three ways to achieve this end:

1. Select *Project Properties* from the *Project* (Gnome, OSX) or *Settings* (KDE, Windows) menu.
2. Click on the CRS status icon in the lower right-hand corner of the statusbar.
3. Turn OTF on by default, by selecting the *CRS* tab of the *Options* dialog and selecting *Enable 'on the fly' reprojection by default*.

If you have already loaded a layer, and want to enable OTF projection, the best practice is to open the *Coordinate Reference System* menu of the *Project Properties* dialog, select a CRS, and activate the *Enable on the fly CRS transformation* checkbox. The CRS status icon will no longer be greyed-out and all layers will be OTF projected to the CRS shown next to the icon.

The *Coordinate Reference System* menu of the *Project Properties* dialog contains five important components as shown in Figure_projection_2 and described below.

1. **Enable on the fly CRS transformation** - this checkbox is used to enable or disable OTF projection. When off, each layer is drawn using the coordinates as read from the data source and the components described below are inactive. When on, the coordinates in each layer are projected to the coordinate reference system defined for the map canvas.

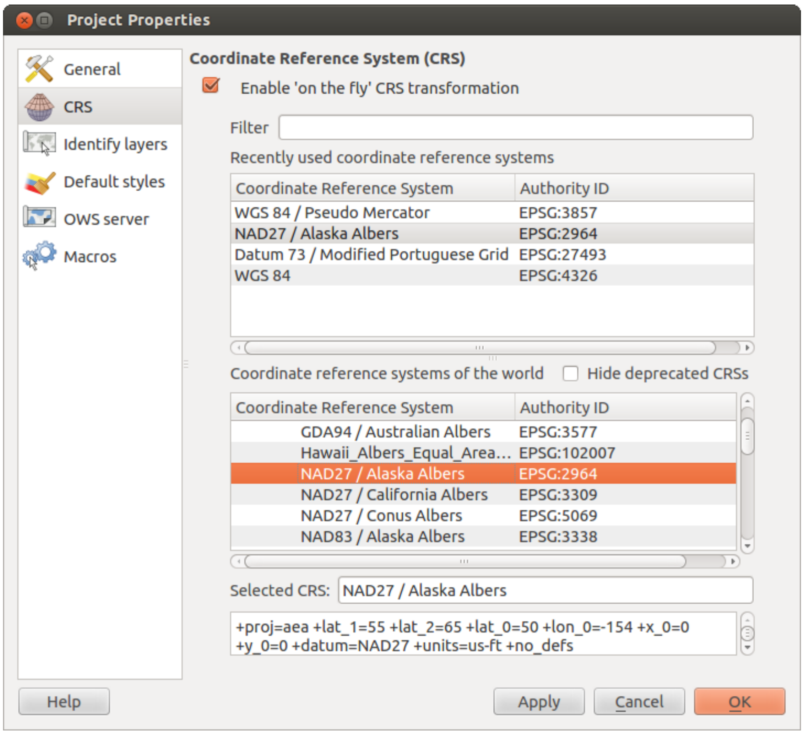


Figure 10.2: Project Properties Dialog

2. **Coordinate Reference System** - this is a list of all CRS supported by QGIS, including Geographic, Projected and Custom coordinate reference systems. To define a CRS, select it from the list by expanding the appropriate node and selecting the CRS. The active CRS is preselected.
3. **Proj4 text** - this is the CRS string used by the Proj4 projection engine. This text is read-only and provided for informational purposes.
4. **Filter** - if you know the EPSG code, the identifier or the name for a Coordinate Reference System, you can use the search feature to find it. Enter the EPSG code, the identifier or the name.
5. **Recently used CRS** - if you have certain CRS that you frequently use in your everyday GIS work, these will be displayed in the table at the bottom of the Projection Dialog. Click on one of these buttons to select the associated CRS.

Tip: Project Properties Dialog

If you open the *Project Properties* dialog from the *Project* (Gnome, OSX) or *Settings* (KDE, Windows) menu, you must click on the *CRS* menu to view the Coordinate Reference System settings.

The CRS status icon will also automatically bring the *CRS* menu to the front.

10.4 Custom Coordinate Reference System

If QGIS does not provide the coordinate reference system you need, you can define a custom CRS. To define a CRS, select *Custom CRS* from the *Settings* menu. Custom CRS are stored in your QGIS user database. In addition to your custom CRS, this database also contains your spatial bookmarks and other custom data.

Defining a custom CRS in QGIS requires a good understanding of the Proj.4 projection library. To begin, refer to the 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 (available at ftp://ftp.remotesensing.org/proj/OF90-284.pdf).

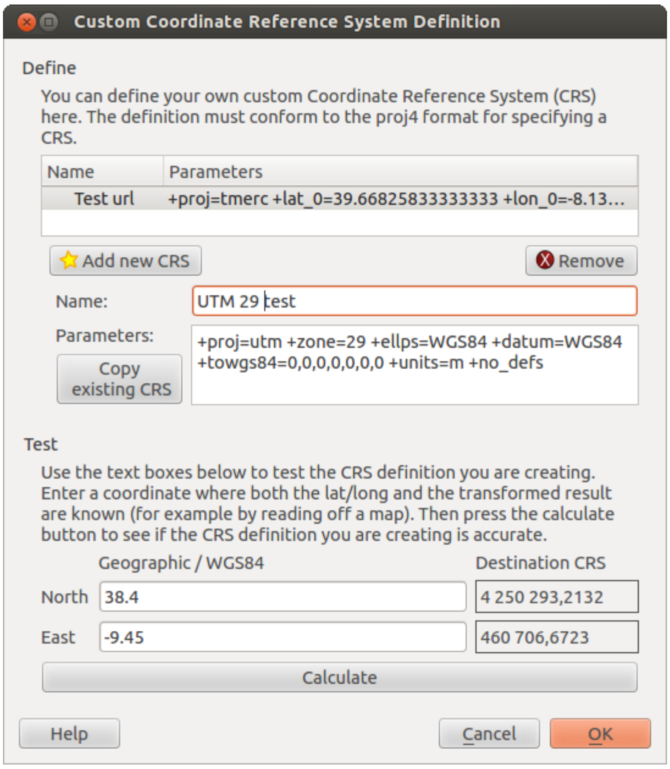


Figure 10.3: Custom CRS Dialog

This manual describes the use of the `proj.4` and related command line utilities. The cartographic parameters used with `proj.4` are described in the user manual and are the same as those used by QGIS.

The *Custom Coordinate Reference System Definition* dialog requires only two parameters to define a user CRS:

1. a descriptive name and
2. the cartographic parameters in PROJ.4 format.

To create a new CRS, click the New button and enter a descriptive name and the CRS parameters. After that you can save your CRS by clicking the Save button.

Note that the *Parameters* must begin with a `+proj=`-block, to represent the new coordinate reference system.

You can test your CRS parameters to see if they give sane results by clicking on the **[Calculate]** button inside the *Test* block and pasting your CRS parameters into the *Parameters* field. Then enter known WGS 84 latitude and longitude values in *North* and *East* fields respectively. Click on **[Calculate]** and compare the results with the known values in your coordinate reference system.

QGIS Browser

The QGIS Browser is a panel in QGIS that lets you easily navigate in your database. You can have access to common vector files (e.g. ESRI shapefile or MapInfo files), databases (e.g.PostGIS, Oracle, Spatialite or MSSQL Spatial) and WMS/WFS connections. You can also view your GRASS data (to get the data into QGIS see *GRASS GIS Integration*).

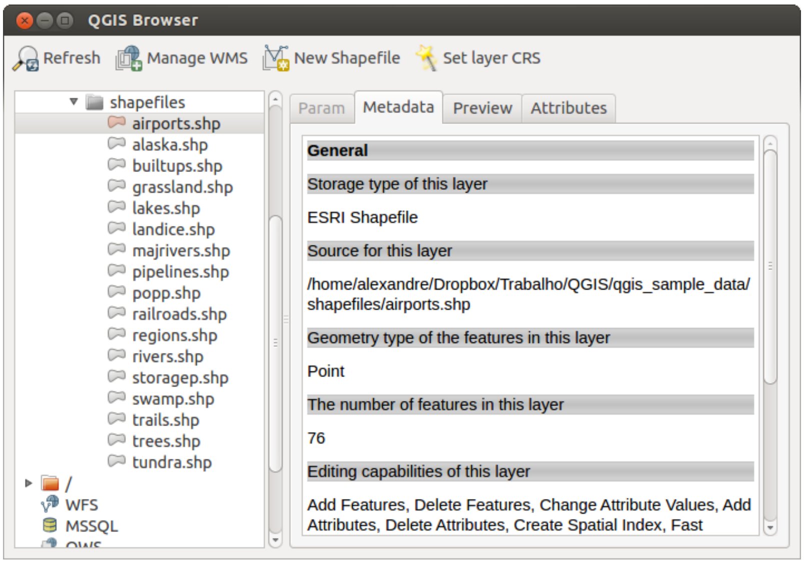


Figure 11.1: QGIS browser as a standalone application

Use QGIS Browser to preview your data. The drag and drop function makes it easy to get your data into the Map view and the Map legend.

1. Activate QGIS Browser: Right-click on the tool bar and click *Browser* or select from *Settings* → *Panels*.
2. Drag the panel into the legend window.
3. Click on the *Browser* tab.
4. Browse in your database and choose the shapefile folder from qgis_sample_data.
5. Press the `Shift` key and click on airports.shp and alaska.shp .
6. Press the left mouse button then drag and drop the files into the map canvas.
7. Right-click on a layer and choose *Set project CRS from layer*. For more information see *Working with Projections*.
8. Click on Zoom Full to make the layers visible.

There is a second browser available under *Settings* → *Panels*. Additional functions like the Add Selected Layers and the Filter files function were inserted.

1. Activate a second QGIS Browser: Right-click on the toolbar and click *Browser (2)* .
2. Drag the panel into the legend window.
3. Navigate to the *Browser (2)* tab and browse for a shapefile in your file system.
4. Select a file with the left mouse button. Now you can use the Add Selected Layers icon. Use the icon to be able to work with a file in QGIS without much effort.

QGIS automatically looks for the Coordinate Reference System (CRS) and zooms to the layer extent if you work in a blank QGIS project. If there are already files in your project the file just will be added and in case it has the same extent and CRS it will be visualized. If the file has got another CRS and layer extent you must first right-click on the layer and choose *Set Project CRS from Layer*. Then choose *Zoom to Layer Extent*.

The Filter files function works on a directory level. Browse to the folder where you want to filter files and give in a search word or a wildcard. The browser only shows matching filenames then, other data won't be displayed then.

It's also possible to run QGIS Browser as a standalone application.

Start QGIS browser

- Type in "qbrowser" at a command prompt.
- Start QGIS browser using the Start menu or desktop shortcut, or double click on a QGIS project file.
- QGIS browser is available from your your Applications folder.

In figure_browser_standalone_metadata you can see the enhanced functionality of QGIS browser. The *Param* tab provides the details of your connection based datasets like PostGIS or MSSQL Spatial. The *Metadata* tab contains general information about the file (see *Metadata Menu*). With the *Preview* tab you can have a look at your files without importing them into your QGIS project. It's also possible to preview the attributes of your files in the *Attributes* tab.

CHAPTER 12

Working with Vector Data

12.1 Supported Data Formats

QGIS uses the OGR library to read and write vector data formats including ESRI Shapefiles, MapInfo, Microstation file formats and many more. GRASS vector, PostGIS, MSSQL Spatial and Oracle Spatial support is supplied by native QGIS data providers. Vector data can also be loaded in read mode from zip and gzip archives into QGIS. At the date of this document, 69 vector formats are supported by the OGR library (see OGR-SOFTWARE-SUITE *Literature and Web References*). The complete list of supported vector data formats by the OGR library (see OGR-SOFTWARE-SUITE *Literature and Web References*) is available at http://www.gdal.org/ogr/ogr_formats.html.

Note: Not all of the listed formats may work in QGIS for various reasons. For example, some require external commercial libraries or the GDAL/OGR installation of your OS was not build to support the format you want to use. Only those formats that have been well tested will appear in the list of file types when loading a vector into QGIS. Other untested formats can be loaded by selecting `*.*`.

Working with GRASS vector data is described in Section *GRASS GIS Integration*.

This section describes how to work with several common formats: ESRI Shapefiles, PostGIS layers, SpatiaLite layers, OpenStreetMap vectors etc. Many of the features available in QGIS work the same, regardless of the vector data source. This is by design and includes the identify, select, labeling and attributes functions.

12.1.1 ESRI Shapefiles

The standard vector file format used in QGIS is the ESRI Shapefile. Support is provided by the OGR Simple Feature Library (http://www.gdal.org/ogr/).

A shapefile actually consists of several files. The following three are required:

1. `.shp` file containing the feature geometries.
2. `.dbf` file containing the attributes in dBase format.
3. `.shx` index file.

Shapefiles also can include a file with a `.prj` suffix, which contains the projection information. While it is very useful to have a projection file, it is not mandatory. A shapefile dataset can contain additional files. For further details see the ESRI technical specification at: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

Loading a Shapefile

To load a shapefile, start QGIS and click on the Add Vector Layer toolbar button or simply type `Ctrl+Shift+V`. This will bring up a new window (see figure_vector_1).

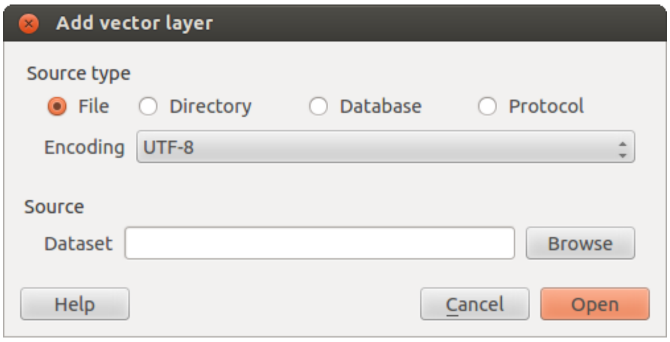


Figure 12.1: Add Vector Layer Dialog

From the available options check *File*. Click on button **[Browse]**. That will bring up a standard open file dialog (see figure_vector_2) which allows you to navigate the file system and load a shapefile or other supported data source. The selection box *Filter* allows you to preselect some OGR supported file formats.

You can also select the Encoding type for the shapefile if desired.

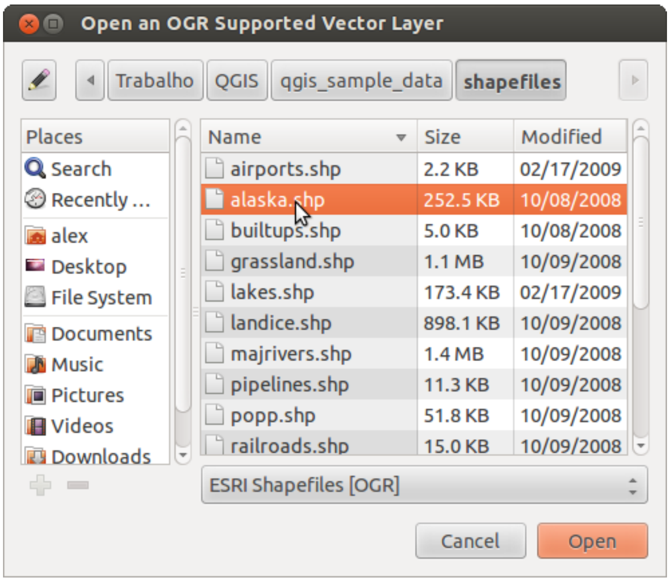


Figure 12.2: Open an OGR Supported Vector Layer Dialog

Selecting a shapefile from the list and clicking **[Open]** loads it into QGIS. Figure_vector_3 shows QGIS after loading the `alaska.shp` file.

Tip: Layer Colors

When you add a layer to the map, it is assigned a random color. When adding more than one layer at a time, different colors are assigned to each layer.

Once loaded, you can zoom around the shapefile using the map navigation tools. To change the style of a layer, open the *Layer Properties* dialog by double clicking on the layer name or by right-clicking on the name in the legend and choosing *Properties* from the popup menu. See Section *Style Menu* for more information on setting symbology of vector layers.

Tip: Load layer and project from mounted external drives on OS X

On OS X, portable drives that are mounted besides the primary hard drive do not show up under *File* → *Open Project* as expected. We are working on a more OSX-native open/save dialog to fix this. As a workaround you can

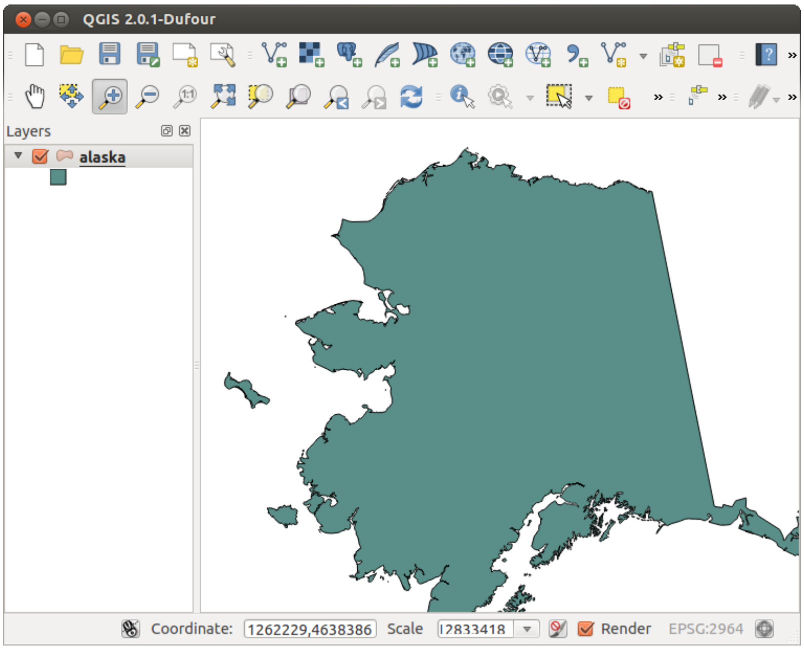


Figure 12.3: QGIS with Shapefile of Alaska loaded

type '/Volumes' in the File name box and press `return`. Then you can navigate to external drives and network mounts.

Improving Performance for Shapefiles

To improve the performance of drawing a shapefile, you can create a spatial index. A spatial index will improve the speed of both zooming and panning. Spatial indexes used by QGIS have a `.qix` extension.

Use these steps to create the index:

- Load a shapefile, clicking on the Add Vector Layer toolbar button or type `Ctrl+Shift+V`.
- Open the *Layer Properties* dialog by double-clicking on the shapefile name in the legend or by right-clicking and choosing *Properties* from the popup menu.
- In the tab *General* click the **[Create Spatial Index]** button.

Problem loading a shape .prj file

If you load a shapefile with `.prj` file and QGIS is not able to read the coordinate reference system from that file, you have to define the proper projection manually within the *General* tab of the *Layer Properties* dialog of the layer by clicking the **[Specify...]** button. This is due to the fact, that `.prj` files often do not provide the complete projection parameters, as used in QGIS and listed in the *CRS* dialog.

For that reason, if you create a new shapefile with QGIS, two different projection files are created. A `.prj` file with limited projection parameters, compatible with ESRI software, and a `.qpj` file, providing the complete parameters of the used CRS. Whenever QGIS finds a `.qpj` file, it will be used instead of the `.prj`.

12.1.2 Loading a MapInfo Layer

To load a MapInfo layer, click on the Add Vector Layer toolbar button or type `Ctrl+Shift+V`, change the file type filter *Filter* : to 'Mapinfo File [OGR]' and select the MapInfo layer you want to load.

12.1.3 Loading an ArcInfo Binary Coverage

To load an ArcInfo binary coverage, click on the Add Vector Layer toolbar button or press `Ctrl+Shift+V` to open the *Add Vector Layer* dialog. Select *Directory*. Change to *Filter* to 'Arc/Info Binary Coverage'. Navigate to the directory that contains the coverage files and select it.

Similarly, you can load directory based vector files in the UK National Transfer Format as well as the raw TIGER Format of the US Census Bureau.

12.1.4 Loading OpenStreetMap Vectors

QGIS integrates Openstreetmap import as a core functionnality.

- To connect to the OSM server and download data, open the menu *Vector* → *Openstreetmap* → *Load data*. You can skip this step if you already got a .osm XML file using josm or overpass or any other source.
- The menu *Vector* → *Openstreetmap* → *Import topology from an XML file* will convert your .osm file into a spatialite database, and create a db connection.
- The menu *Vector* → *Openstreetmap* → *Export topology to Spatialite* then allows you to open the database connection, select the type of data you want (points, lines, or polygons) and choose tags to import. This creates a spatialite geometry layer that you can then add to your project by clicking on the Add SpatiaLite Layer toolbar button or by selecting the *Add SpatiaLite Layer...* option from the *Layer* menu, see Section *SpatiaLite Layers*.

12.1.5 PostGIS Layers

PostGIS layers are stored in a PostgreSQL database. The advantages of PostGIS are the spatial indexing, filtering and query capabilities it provides. Using PostGIS, vector functions such as select and identify work more accurately than with OGR layers in QGIS.

Creating a stored Connection

The first time you use a PostGIS data source, you must create a connection to the PostgreSQL database that contains the data. Begin by clicking on the Add PostGIS Layer toolbar button, selecting the *Add PostGIS Layer...* option from the *Layer* menu or typing `Ctrl+Shift+D`. You can also open the *Add Vector Layer* dialog and select *Database*. The *Add PostGIS Table(s)* dialog will be displayed. To access the connection manager, click on the **[New]** button to display the *Create a New PostGIS Connection* dialog. The parameters required for a connection are:

- **Name**: A name for this connection. Can be the same as *Database*
- **Service**: Service parameter to be used alternatively to hostname/port (and potentially database). This can be defined in `pg_service.conf`
- **Host**: Name of the database host. This must be a resolvable host name the same as would be used to open a telnet connection or ping the host. If the database is on the same computer as QGIS, simply enter *'localhost'* here.
- **Port**: Port number the PostgreSQL database server listens on. The default port is 5432.
- **Database**: Name of the database.
- **SSL mode**: How the SSL connection will be negotiated with the server. Note that massive speedups in PostGIS layer rendering can be achieved by disabling SSL in the connection editor. Following options are available:
  - disable: only try an unencrypted SSL connection

- allow: try a non-SSL connection, if that fails, try an SSL connection
- prefer (the default): try an SSL connection, if that fails, try a non-SSL connection;
- require: only try an SSL connection.

- **Username**: User name used to login to the database.
- **Password**: Password used with *Username* to connect to the database.

Optional you can activate following checkboxes:

- *Save Username*
- *Save Password*
- *Only look in the geometry_columns table*
- *Don't resolve type of unrestricted columns (GEOMETRY)*
- *Only look in the 'public' schema*
- *Also list tables with no geometry*
- *Use estimated table metadata*

Once all parameters and options are set, you can test the connection by clicking on the **[Test Connect]** button.

Tip: QGIS User Settings and Security

Depending on your computing environment, storing passwords in your QGIS settings may be a security risk. Your customized settings for QGIS are stored based on the operating system:

- , the settings are stored in your home directory in `.qgis2/`.
- , the settings are stored in the registry.

Loading a PostGIS Layer

Once you have one or more connections defined, you can load layers from the PostgreSQL database. Of course this requires having data in PostgreSQL. See Section *Importing Data into PostgreSQL* for a discussion on importing data into the database.

To load a layer from PostGIS, perform the following steps:

- If the *Add PostGIS layers* dialog is not already open, selecting the *Add PostGIS Layer...* option from the *Layer* menu or typing `Ctrl+Shift+D` opens the dialog.
- Choose the connection from the drop-down list and click **[Connect]**.
- Select or unselect *Also list tables with no geometry*
- Optionally use some *Search Options* to define which features to load from the layer or use the **[Build query]** button to start the *Query builder* dialog.
- Find the layer(s) you wish to add in the list of available layers.
- Select it by clicking on it. You can select multiple layers by holding down the `Shift` key while clicking. See Section *Query Builder* for information on using the PostgreSQL Query Builder to further define the layer.
- Click on the **[Add]** button to add the layer to the map.

Tip: PostGIS Layers

Normally a PostGIS layer is defined by an entry in the geometry_columns table. From version 0.9.0 on, QGIS can load layers that do not have an entry in the geometry_columns table. This includes both tables and views. Defining a spatial view provides a powerful means to visualize your data. Refer to your PostgreSQL manual for information on creating views.

Some details about PostgreSQL layers

This section contains some details on how QGIS accesses PostgreSQL layers. Most of the time QGIS should simply provide you with a list of database tables that can be loaded, and load them on request. However, if you have trouble loading a PostgreSQL table into QGIS, the information below may help you understand any QGIS messages and give you direction on changing the PostgreSQL table or view definition to allow QGIS to load it.

QGIS requires that PostgreSQL layers contain a column that can be used as a unique key for the layer. For tables this usually means that the table needs a primary key, or a column with a unique constraint on it. In QGIS, this column needs to be of type int4 (an integer of size 4 bytes). Alternatively the ctid column can be used as primary key. If a table lacks these items, the oid column will be used instead. Performance will be improved if the column is indexed (note that primary keys are automatically indexed in PostgreSQL).

If the PostgreSQL layer is a view, the same requirement exists, but views do not have primary keys or columns with unique constraints on them. You have to define a primary key field (has to be integer) in the QGIS dialog before you can load the view. If a suitable column cannot does not exist in the view, QGIS will not load the layer. If this occurs, the solution is to alter the view so that it does include a suitable column (a type of integer and either a primary key or with a unique constraint, preferably indexed).

QGIS offers a checkbox **Select at id** that is activated by default. This option gets the ids without the attributes which is faster in most cases. It can make sense to disable this option when you use expensive views.

12.1.6 Importing Data into PostgreSQL

Data can be imported into PostgreSQL/PostGIS using several tools, such as the SPIT plugin or the command line tools shp2pgsql or ogr2ogr.

DB Manager

QGIS comes with a core plugin named DB Manager. It can be used to load shapefiles and other data formats and includes support for schemas. See Section *DB Manager Plugin* for more information.

shp2pgsql

PostGIS includes an utility called **shp2pgsql** that can be used to import shapefiles into a PostGIS enabled database. For example, to import a shapefile named `lakes.shp` into a PostgreSQL database named `gis_data`, use the following command:

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

This creates a new layer named `lakes_new` in the `gis_data` database. The new layer will have a spatial reference identifier (SRID) of 2964. See Section *Working with Projections* for more information on spatial reference systems and projections.

Tip: Exporting datasets from PostGIS

Like the import-tool **shp2pgsql** there is also a tool to export PostGIS-datasets as shapefiles: **pgsql2shp**. This is shipped within your PostGIS distribution.

ogr2ogr

Beside **shp2pgsql** and **DB Manager** there is another tool for feeding geodata in PostGIS: **ogr2ogr**. This is part of your GDAL installation.

To import a shapefile into PostGIS, do the following:

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

This will import the shapefile `alaska.shp` into the PostGIS-database *postgis* using the user *postgres* with the password *topsecret* on host server *myhost.de*.

Note that OGR must be built with PostgreSQL to support PostGIS. You can see this by typing

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

If you like to use PostgreSQL's **COPY** -command instead of the default **INSERT INTO** method you can export the following environment-variable (at least available on and):

```
export PG_USE_COPY=YES
```

ogr2ogr does not create spatial indexes like **shp2pgsl** does. You need to create them manually using the normal SQL-command **CREATE INDEX** afterwards as an extra step (as described in the next section *Improving Performance*).

Improving Performance

Retrieving features from a PostgreSQL database can be time consuming, especially over a network. You can improve the drawing performance of PostgreSQL layers by ensuring that a PostGIS spatial index exists on each layer in the database. PostGIS supports creation of a GiST (Generalized Search Tree) index to speed up spatial searches of the data (GiST index information is taken from the PostGIS documentation available at http://postgis.refractions.net).

The syntax for creating a GiST index is:

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

Note that for large tables, creating the index can take a long time. Once the index is created, you should perform a `VACUUM ANALYZE`. See the PostGIS documentation (POSTGIS-PROJECT *Literature and Web References*) for more information.

The following is an example of creating a GiST index:

```
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$
```

12.1.7 Vector layers crossing 180° longitude

Many GIS packages don't wrap vector maps, with a geographic reference system (lat/lon), crossing the 180 degrees longitude line (http://postgis.refractions.net/documentation/manual-1.4/ST_Shift_Longitude.html). As result, if we open such map in QGIS, we will see two far, distinct locations, that should show near each other. In Figure_vector_4 the tiny point on the far left of the map canvas (Chatham Islands), should be within the grid, right of New Zealand main islands.

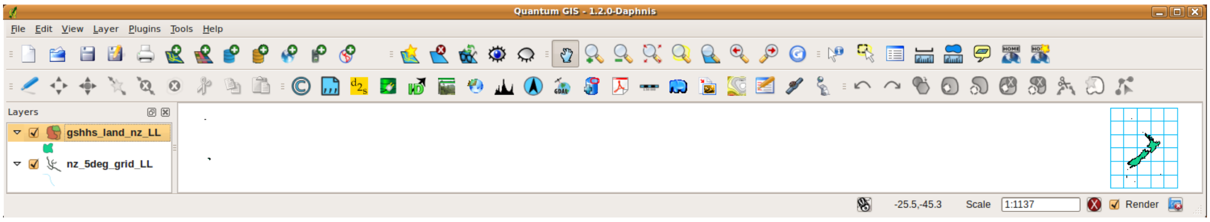


Figure 12.4: Map in lat/lon crossing the 180° longitude line

A workaround is to transform the longitude values using PostGIS and the **ST_Shift_Longitude** function This function reads every point/vertex in every component of every feature in a geometry, and if the longitude coordinate is < 0° adds 360° to it. The result would be a 0° - 360° version of the data to be plotted in a 180° centric map.

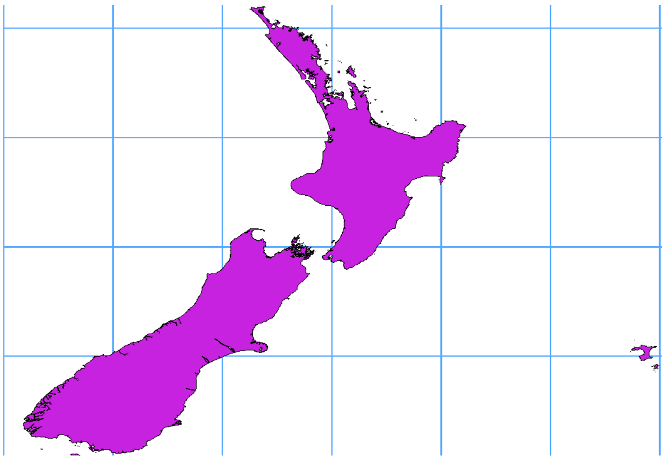


Figure 12.5: Crossing 180° longitude applying the **ST_Shift_Longitude** function

Usage

- Import data to PostGIS (*Importing Data into PostgreSQL*) using for example the DB Manager plugin.
- Use the PostGIS command line interface to issue the following command (this is an example where "TABLE" is the actual name of your PostGIS table)

  ```
  gis_data=# update TABLE set the_geom=ST_Shift_Longitude(the_geom);
  ```

- If everything went right you should receive a confirmation about the number of features that were updated, then you'll be able to load the map and see the difference (Figure_vector_5)

12.1.8 SpatiaLite Layers

The first time you load data from a SpatiaLite database, begin by clicking on the Add SpatiaLite Layer toolbar button or by selecting the *Add SpatiaLite Layer...* option from the *Layer* menu or by typing `Ctrl+Shift+L`.

This will bring up a window, which will allow you to either connect to a SpatiaLite database already known to QGIS, which you can choose from the dropdown menu or to define a new connection to a new database. To define a new connection, click on **[New]** and use the file browser to point to your SpatiaLite database, which is a file with a `.sqlite` extension.

If you want to save a vector layer to SpatiaLite format you can do this by right clicking the layer in the legend. Then click on *Save as..*, define the name of the output file, select 'SpatiaLite' as format and the CRS. Also you can select 'SQLite' as format, and then add `SPATIALITE=YES` in the OGR data source creation option field. This tells OGR to create a SpatiaLite database. See also http://www.gdal.org/ogr/drv_sqlite.html.

QGIS also supports editable views in SpatiaLite.

Creating a new SpatiaLite layer

If you want to create a new SpatiaLite layer, please refer to section *Creating a new SpatiaLite layer*.

Tip: SpatiaLite data management Plugins

For SpatiaLite data management you can also use several Python plugins: QSpatiaLite, SpatiaLite Manager or DB Manager (core plugin, recommended). They can be downloaded and installed with the Plugin Installer.

12.1.9 MSSQL Spatial Layers

QGIS also provides native MS SQL 2008 support. The Add MSSQL Spatial Layer is part of the new toolbar button or available in the MS SQL node in the QBrowser tree, providing drag and drop import support.

12.1.10 ORACLE Spatial Layers

QGIS also provides native ORACLE Locator/Spatial support. The Add ORACLE Spatial Layer is part of the new toolbar button or available in the ORACLE node in the QBrowser tree, providing drag and drop import support. ORACLE Spatial layers are stored in an ORACLE database.

Creating a stored Connection

The first time you use a ORACLE Spatial data source, you must create a connection to the database that contains the data. Begin by clicking on the Add ORACLE Spatial Layer toolbar button, selecting the *Add ORACLE Spatial Layer...* option from the *Layer* menu or typing `Ctrl+Shift+O`. To access the connection manager, click on the **[New]** button to display the *Create a New ORACLE Spatial Connection* dialog. The parameters required for a connection are:

- **Name**: A name for this connection. Can be the same as *Database*
- **Database** SID or SERVICE_NAME of the Oracle instance.
- **Host**: Name of the database host. This must be a resolvable host name the same as would be used to open a telnet connection or ping the host. If the database is on the same computer as QGIS, simply enter *'localhost'* here.
- **Port**: Port number the PostgreSQL database server listens on. The default port is 1521.
- **Username**: User name used to login to the database.
- **Password**: Password used with *Username* to connect to the database.

Optional you can activate following checkboxes:

- *Save Username* Indicates whether to save the database user name in the connection configuration.

- *Save Password* Indicates whether to save the database password in the connection settings. Passwords are saved in clear text in the system configuration and in the project files!
- *Only look in meta data table* Restricts the displayed tables to those that are in the all_sdo_geom_metadata view. This can speed up the initial display of spatial tables.
- *Only look for user's tables* When searching for spatial tables restrict the search to tables that are owner by the user.
- *Also list tables with no geometry* Indicates that tables without geometry should also be listed by default.
- *Use estimated table statistics for the layer metadata* When the layer is setup various metadata is required for the Oracle table. This includes information such as the table row count, geometry type and spatial extents of the data in the geometry column. If the table contains a large number of rows determining this metadata is time consuming. By activating this option the following fast table metadata operations are done: Row count is determined from all_tables.num_rows. Table extents are always determined with the SDO_TUNE.EXTENTS_OF function even if a layer filter is applied. The table geometry is determined from the first 100 non-null geometry rows in the table.
- *Only existing geometry types* Only list the existing geometry types and don't offer to add others.

Once all parameters and options are set, you can test the connection by clicking on the **[Test Connect]** button.

Tip: QGIS User Settings and Security

Depending on your computing environment, storing passwords in your QGIS settings may be a security risk. Passwords are saved in clear text in the system configuration and in the project files! Your customized settings for QGIS are stored based on the operating system:

- , the settings are stored in your home directory in `.config/QGIS/QGIS2.conf`.
- , the settings are stored in the registry.

Loading a ORACLE Spatial Layer

Once you have one or more connections defined, you can load layers from the ORACLE database. Of course this requires having data in ORACLE.

To load a layer from ORACLE Spatial, perform the following steps:

- If the *Add ORACLE Spatial layers* dialog is not already open, click on the Add ORACLE Spatial Layer toolbar button.
- Choose the connection from the drop-down list and click **[Connect]**.
- Select or unselect *Also list tables with no geometry*
- Optionally use some *Search Options* to define which features to load from the layer or use the **[Build query]** button to start the *Query builder* dialog.
- Find the layer(s) you wish to add in the list of available layers.
- Select it by clicking on it. You can select multiple layers by holding down the `Shift` key while clicking. See Section *Query Builder* for information on using the ORACLE Query Builder to further define the layer.
- Click on the **[Add]** button to add the layer to the map.

Tip: ORACLE Spatial Layers

Normally an ORACLE Spatial layer is defined by an entry in the **USER_SDO_METADATA** table.

12.2 The Vector Properties Dialog

The *Layer Properties* dialog for a vector layer provides information about the layer, symbology settings and labeling options. If your vector layer has been loaded from a PostgreSQL/PostGIS datastore, you can also alter the underlying SQL for the layer by invoking the *Query Builder* dialog on the *General* tab. To access the *Layer Properties* dialog, double-click on a layer in the legend or right-click on the layer and select *Properties* from the popup menu.

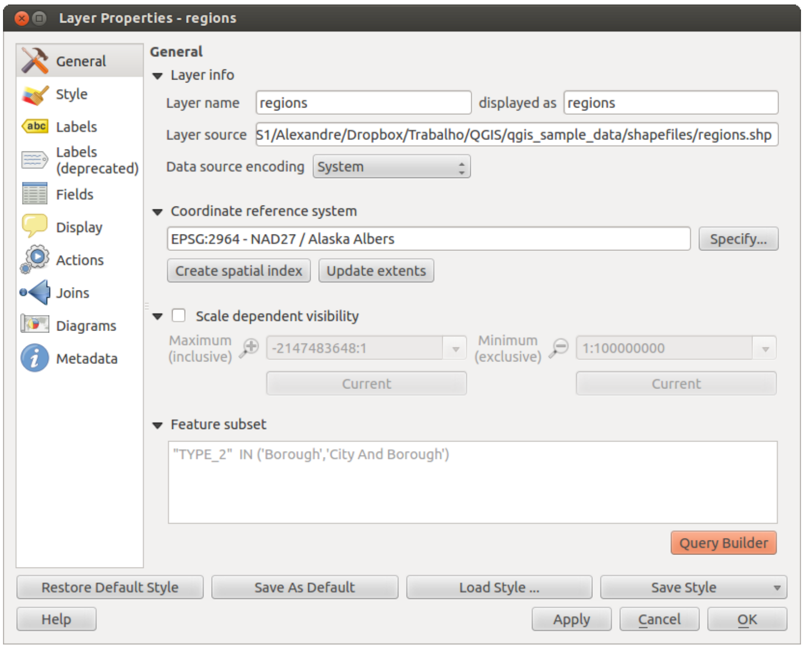


Figure 12.6: Vector Layer Properties Dialog

12.2.1 Style Menu

The Style menu provides you with a comprehensive tool for rendering and symbolizing your vector data. You can use *Layer rendering* → tools that are common to all vector data and special symbolizing tools that were designed for the different kinds of vector data.

Layer rendering

- *Layer transparency* : you can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the the menu beside the slider.
- *Layer blending mode*: you can achieve special rendering effects with these tools that you previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described below.
  - Normal: This is the standard blend mode which uses the alpha channel of the top pixel to blend with the Pixel beneath it; the colors aren't mixed
  - Lighten: It selects the maximum of each component from the foreground and background pixels. Be aware that the results tend to be jagged and harsh.
  - Screen: Light pixels from the source are painted over the destination, while dark pixels are not. This mode is most useful for mixing the texture of one layer with another layer. E.g. you can use a hillshade to texture another layer

- Dodge: Dodge will brighten and saturate underlying pixels based on the lightness of the top pixel. So brighter top pixels cause the saturation and brightness of the underlying pixels to increase. This works best if the top pixels aren't too bright, otherwise the effect is too extreme.
- Addition: This blend mode simply adds pixel values of one layer with the other. In case of values above 1 (in the case of RGB), white is displayed. This mode is suitable for highlighting features.
- Darken: Creates a resultant pixel that retains the smallest components of the foreground and background pixels. Like lighten, the results tend to be jagged and harsh
- Multiply: It multiplies the numbers for each pixel of the top layer with the corresponding pixel for the bottom layer. The results are darker pictures.
- Burn: Darker colors in the top layer causes the underlying layers to darken. Can be used to tweak and colorise underlying layers.
- Overlay: Combines multiply and screen blending modes. In the resulting picture light parts of the picture become lighter and dark parts become darker.
- Soft light: Very similar to overlay, but instead of using multiply/screen it uses color burn/dodge. This one is supposed to emulate shining a soft light onto an image.
- Hard light: Hard light is very similar to the overlay mode. It's supposed to emulate projecting a very intense light onto an image.
- Difference: Difference subtracts the top pixel from the bottom pixel or the other way round, to always get a positive value. Blending with black produces no change, as values for all colors are 0.
- Subtract: This blend mode simply subtracts pixel values of one layer with tshe other. In case of negative values, black is displayed.

Since QGIS 2.0 the old symbology is no longer available. In this version the new symbology has been redesigned and revised.

Renderers

The renderer is responsible for drawing a feature together with the correct symbol. There are four types of renderers: single symbol, categorized, graduated and rule-based. There is no continuous color renderer, because it is in fact only a special case of the graduated renderer. The categorized and graduated renderer can be created by specifying a symbol and a color ramp - they will set the colors for symbols appropriately. For point layers there is a point displacement renderer available. For each data type (points, lines and polygons) vector symbol layer types are available. Depending on the chosen renderer, the *Style* menu provides different following sections. On the bottom right of the symbology dialog there is a **[Symbol]** button which gives access to the Style Manager (see Section vector_style_manager). The Style Manager allows you to edit and remove existing symbols and add new ones.

Tip: Select and change multiple symbols

The Symbology allows to select multiple symbols and right click to change color, transparency, size, or width of selected entries.

Single Symbol Renderer

The Single Symbol Renderer is used to render all features of the layer using a single user-defined symbol. The properties, that can be adjusted in the *Style* menu, depend partially on the type of the layer, but all types share the following structure. In the top left part of the menu, there is a preview of the current symbol to be rendered. On the right part of the menu, there is a list of symbols already defined for the current style, prepared to be used via selecting them from the list. The current symbol can be modified using the menu on the right side. If you click on the first level in the *Symbol layers* dialog on the left side it's possible to define basic parameters like *Size*, *Transparency*, *Color* and *Rotation*. Here the layers are joined together.

More detailed settings can be made when clicking on the second level in the *Symbol layers* dialog. You can define *Symbol layers* that are combined afterwards. A symbol can consist of several *Symbol layers*. The following settings are possible:

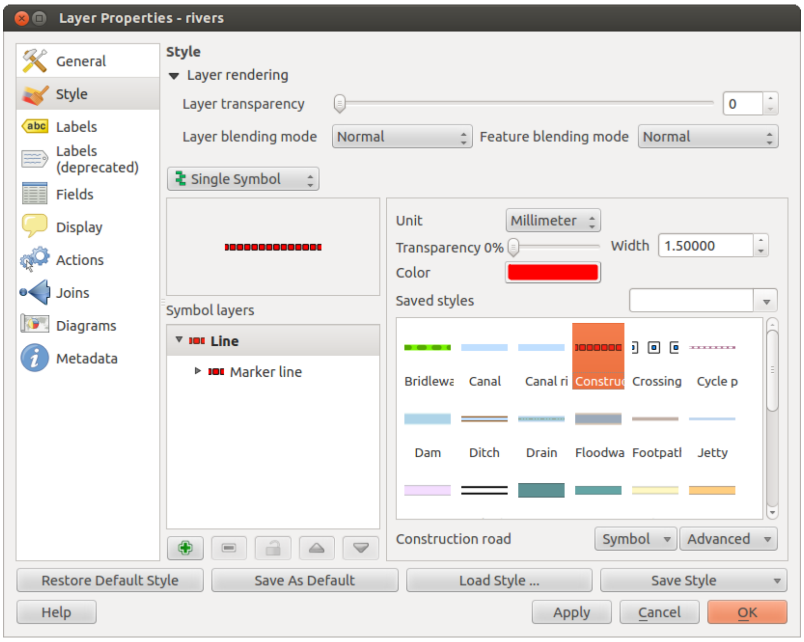


Figure 12.7: Single symbol line properties

- *Symbol layer type*: You have the possibility to use Ellipse markers, Font markers, Simple markers, SVG markers and Vector Field markers
- *Size*
- *Angle*
- *Colors*
- *Border width*
- *Offset X,Y*: You can shift the symbol in x- or y-direction

Note that once you have set the size in the lower levels the size of the whole symbol can be changed with the *Size* menu in the first level again. The size of the lower levels changes accordingly while the size ratio is maintained. After having done any needed changes, the symbol can be added to the list of current style symbols (using the **[Symbol]** *Save in symbol library*) and then easily be used in the future. Furthermore you can use the **[Save Style]** button to save the symbol as a QGIS layer style file (.qml) or SLD file(.sld). Currently in version 2.0 SLDs can be exported from any type of renderer: single symbol, categorized, graduated or rule-based, but when importing an SLD, either a single symbol or rule-based renderer is created. That means that categorized or graduated styles are converted to rule-based. If you want to preserve those renderers, you have to stick to the QML format. On the other hand, it could be very handy sometimes to have this easy way of converting styles to rule-based. With the *Style manager* from the **[Symbol]** menu you can administrate your symbols. You can add item, edit item, remove item and share item. 'Marker' symbols, 'Line' symbols, 'Fill' patterns and 'Color ramps' can be used to create the symbols (see defining_symbols). The symbols are assigned to 'All Symbols', 'Groups' or 'Smart groups' then.

Categorized Renderer

The Categorized Renderer is used to render all features from a layer, using a single user-defined symbol, which color reflects the value of a selected feature's attribute. The *Style* menu allows you to select:

- The attribute (using the Column listbox)
- The symbol (using the Symbol dialog)
- The colors (using the Color Ramp listbox)

The **[Advanced]** button in the lower right corner of the dialog allows to set the fields containing rotation and size scale information. For convenience, the list in the center of the menu lists the values of all currently selected attributes together, including the symbols that will be rendered.

The example in figure_symbology_2 shows the category rendering dialog used for the rivers layer of the QGIS sample dataset.

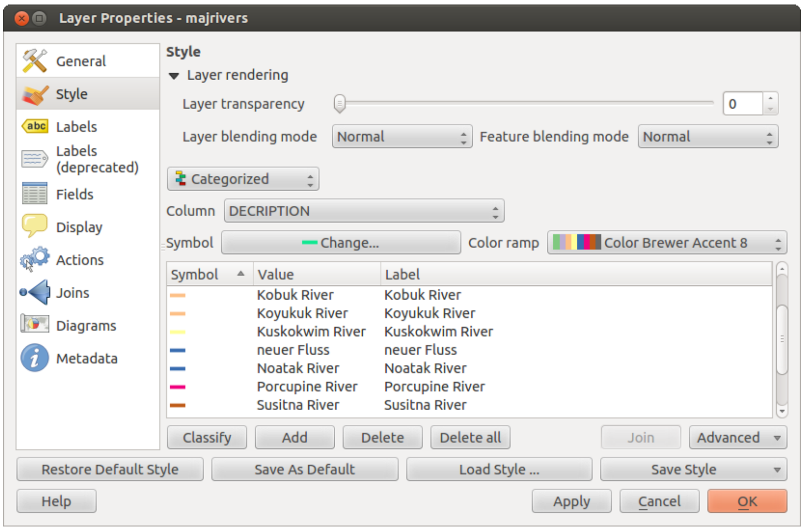


Figure 12.8: Categorized Symbolizing options

You can create a custom color ramp choosing *New color ramp...* from the *Color ramp* dropdown menu. A dialog will prompt for the ramp type: Gradient, Random, ColorBrewer, and cpt-city. The first three have options for number of steps and/or multiple stops in the color ramp. See figure_symbology_3 for an example of custom color ramp and figure_symbology_3a for the cpt-city dialog.

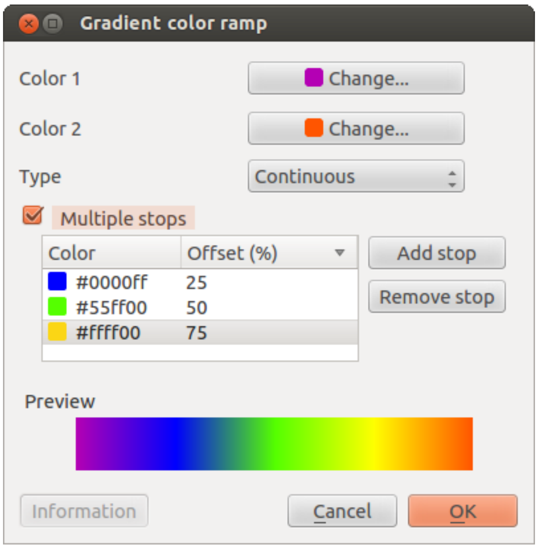


Figure 12.9: Example of custom gradient color ramp with multiple stops

The cpt-city option opens a new dialog with hundreds of themes included 'out of the box'.

Graduated Renderer

The Graduated Renderer is used to render all the features from a layer, using a single user-defined symbol, whose color reflects the classification of a selected feature's attribute to a class.

Like Categorized Renderer, it allows to define rotation and size scale from specified columns.

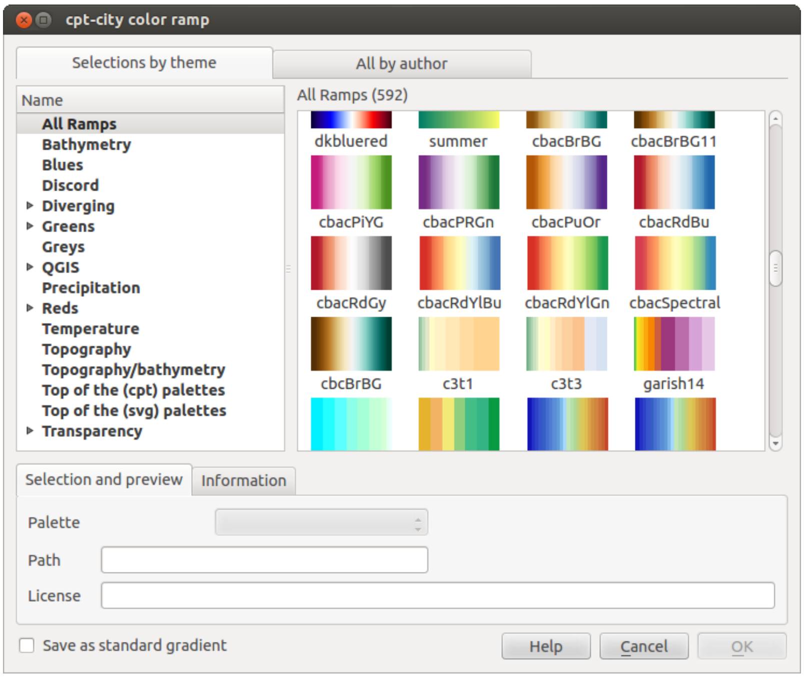


Figure 12.10: cpt-city dialog with hundreds of color ramps

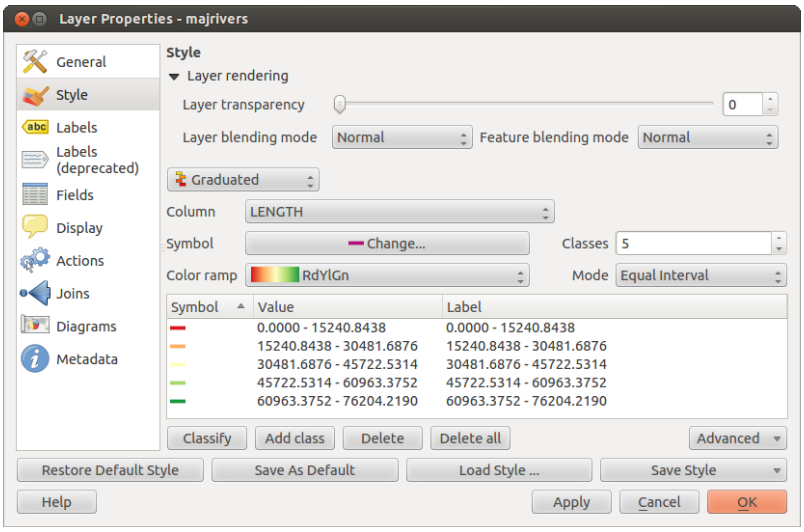


Figure 12.11: Graduated Symbolizing options

Analogue to the categorized rendered, the *Style* tab allows you to select:

- The attribute (using the Column listbox)
- The symbol (using the Symbol Properties button)
- The colors (using the Color Ramp list)

Additionally, you can specify the number of classes and also the mode how to classify features inside the classes (using the Mode list). The available modes are:

- Equal Interval
- Quantile
- Natural Breaks (Jenks)
- Standard Deviation
- Pretty Breaks

The listbox in the center part of the *Style* menu lists the classes together with their ranges, labels and symbols that will be rendered.

The example in figure_symbology_4 shows the graduated rendering dialog for the rivers layer of the QGIS sample dataset.

Rule-based rendering

The rule-based renderer is used to render all the features from a layer, using rule based symbols, whose color reflects the classification of a selected feature's attribute to a class. The rules are based on SQL statements. The dialog allows rule grouping by filter or scale and you can decide if you want to enable symbol levels or use only first matched rule.

The example in figure_symbology_5 shows the rule-based rendering dialog for the rivers layer of the QGIS sample dataset.

To create a rule, activate an existing row by clicking on it or click on '+' and click on the new rule. Then press the **[Edit rule]** button. In the *Rule properties* dialog you can define a label for the rule. Press the ... button to open the Expression builder. In the **Function List**, click on *Fields and Values* to view all attributes of the attribute table to be searched. To add an attribute to the Field calculator **Expression** field, double click its name in the *Fields and Values* list. Generally you can use the various fields, values and functions to construct the calculation expression or you can just type it into the box (see *Field Calculator*).

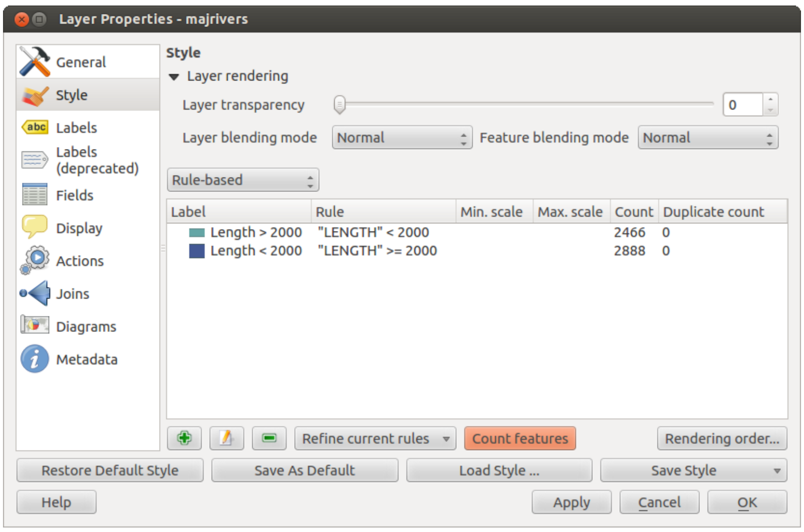


Figure 12.12: Rule-based Symbolizing options

Point displacement

The point displacement renderer offers to visualize all features of a point layer, even if they have the same location. To do this, the symbols of the points are placed on a displacement circle around a center symbol.

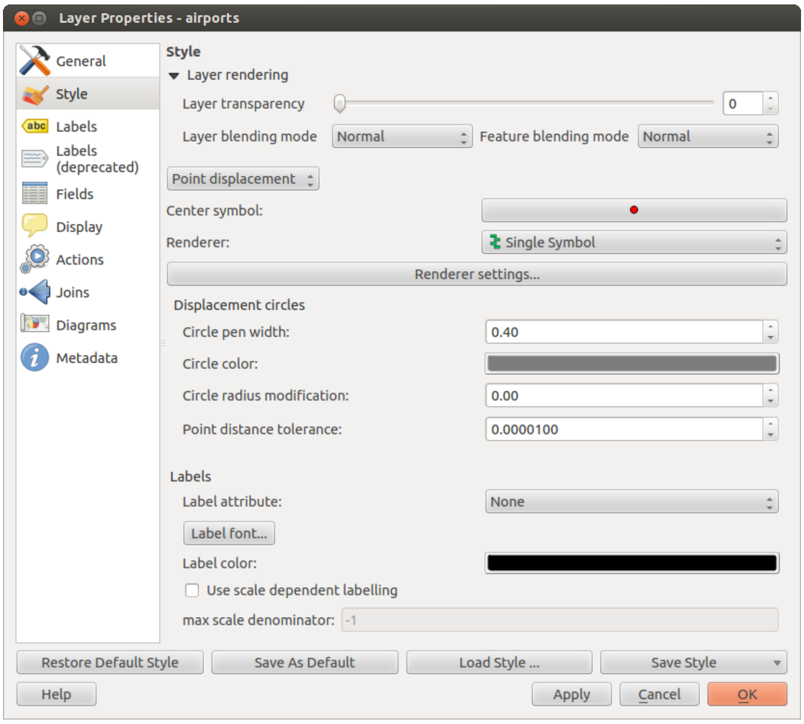


Figure 12.13: Point displacement dialog

Tip: Export vector symbology

You have the possibility to export vector symbology from QGIS into the Google *.kml, *.dxf and MapInfo*.tab files. Just open the right mouse menu of the layer and click on *Save selection as* → to define the name of the output file and its format. Use the *Symbology export* menu to save the symbology either as *Feature symbology* → or as *Symbol layer symbology* →. If you have used symbol layers it is recommended to use the second setting.

12.2.2 Labels Menu

The Labels core application provides smart labeling for vector point, line and polygon layers and only requires a few parameters. This new application also supports on-the-fly transformated layers. The core functions of the application have been redesigned. In QGIS 2.0. there are now a number of other features which improve the labeling. The following menus have been created for labeling the vector layers:

- Text
- Formatting
- Buffer
- Background
- Shadow
- Placement
- Rendering

Let us see how the new menus can be used for various vector layers. **Labeling point layers**

Start QGIS and load a vector point layer. Activate the layer in the legend and click on the Layer Labeling Options icon in the QGIS toolbar menu.

First step is to activate the *Label this layer with* checkbox and select an attribute column to use for labeling. Click if you want to define labels based on expressions.

The following steps describe a simple labeling without using the *Data defined override* functions that are situated next to the drop-down menus.

You can define the text style in the *Text* menu (see Figure_labels_1). A new function is the *Type case* option where you can influence the text rendering. You have the possibility to render the text 'All uppercase', 'All lowercase' or 'Capitalize first letter'. Also, a new function in QGIS 2.0 is the use of blend modes (see blend_modes).

In the *Formatting* menu you can define a character for a line break in the labels with the wrap label on character function. Use the *Formatted numbers* option to format the numbers in an attribute table. Here decimal places are inserted. If you enable this option three decimal places ist set by default.

To create a buffer just activate *Draw text buffer* checkbox in the *Buffer* menu. The buffer color is variable. Also, a new function in QGIS 2.0 is the use of blend modes (see blend_modes).

If the *Color buffer's fill* checkbox is activated, it will interact with partially transparent text and give mixed color transparency results. Turning off the buffer fill fixes that issue (except where the interior aspect of the buffer's stroke intersects with the text's fill) and also allows the user to make outlined text.

In the *Background* menu you can define with *Size X* and *Size Y* the shape of your background. Use *Size type* to insert an additional 'Buffer' into your background. The buffer size one is set by default here. The background then consists of the buffer plus the background in *Size X* and *Size Y*. You can set a *Rotation* where you can choose between 'Sync with label', 'Offset of label' and 'Fixed'. Using 'Offset of label' and 'Fixed' you can rotate the background. Define an *Offset X,Y* with X and Y values and the background will be shifted. When applying *Radius X,Y* the background gets rounded corners. Again, it is possible to mix the background with the underlying layers in the map canvas using the *Blend mode* (see blend_modes).

Use the *Shadow* menu for a user-defined *Drop shadow*. The drawing of the background is very variable. Choose between 'Lowest label component', 'Text', 'Buffer' and 'Background'. The *Offset* angle depends on the orientation of the label. If you choose the *Use global shadow* checkbox then the zero point of the angle is always oriented to the north and doesn't depend on the orientation of the label. Influence the appearance of the shadow with the *Blur radius*. The higher the number, the softer the shadows.

The appearance of the drop shadow can also be altered by choosing a blend mode (see blend_modes).

Choose the *Placement* menu for the label placement and the labeling priority. Using the *Offset from point* setting you now have the possibility to use *Quadrants* to place your label. Additionally you can alter the angle of the label placement with the *Rotation* setting. Thus, a placement in a certain quadrant with a certain rotation is possible.

In the *Rendering* menu you can define label and feature options. In the *Label options* you find the scale-based visibility setting now. You can prevent QGIS from rendering only selected labels with the *Show all labels for this layer (including colliding labels)* checkbox. In the *Feature options* you can define if every part of a multipart feature is to be labeled. In QGIS 2.0 now it's possible to define if the number of features to be labeled is limited and to *Discourage labels from covering features*.

Labeling line layers

First step is to activate the *Label this layer* checkbox in the *Label settings* tab and select an attribute column to use for labeling. Click if you want to define labels based on expressions.

After that you can define the text style in the *Text* menu. Here the same settings as for point layers are possible.

Also in the *Formatting* menu the same settings as for point layers are possible.

The *Buffer* menu has the same functions as described in section labeling_point_layers .

The *Background* menu has the same entries as described in section labeling_point_layers .

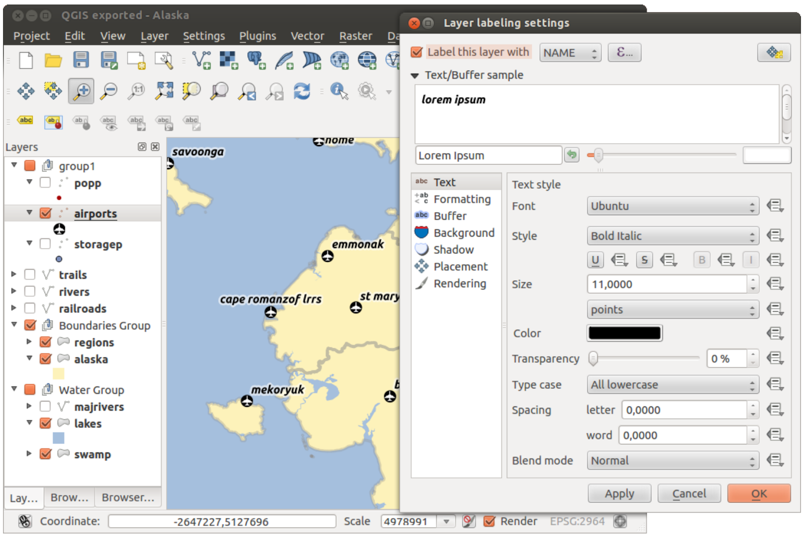


Figure 12.14: Smart labeling of vector point layers

Also the *Shadow* menu has the same entries as described in section labeling_point_layers .

In the *Placement* menu you find special settings for line layers. The label can be placed *Parallel*, *Curved* or *Horizontal*. With the *Parallel* and *Curved* option come the following settings: You can define the position *Above line*, *On line* and *Below line*. It's possible to select several options at once. QGIS will look for the optimal position of the label then. Remember that here you can also use the line orientation for the position of the label. Additionally you can define a *Maximum angle between curved characters* when selecting the *Curved* option (see Figure_labels_2).

The *Rendering* menu has nearly the same entries as for point layers. In the *Feature options* you can now *Suppress labeling of features smaller than*.

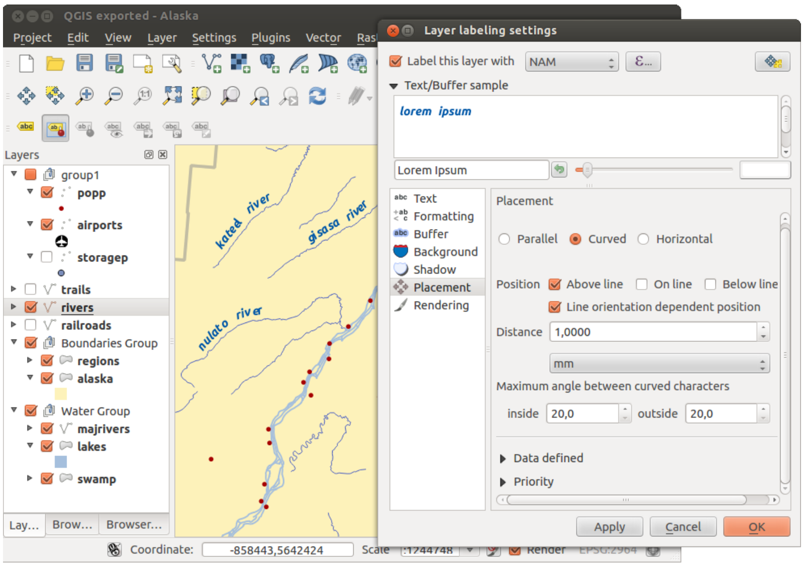


Figure 12.15: Smart labeling of vector line layers

Labeling polygon layers

First step is to activate the *Label this layer* checkbox and select an attribute column to use for labeling. Click ε... if you want to define labels based on expressions.

In *Text* menu define the text style. The entries are the same as for point and line layers.

The *Formatting* menu allows you to format multiple lines like for point and line layers.

As with point and line layers you can create a text buffer in the *Buffer* menu.

Use the *Background* menu to create a complex user-defined background for the polygon layer. You can use the menu as well as for the point and line layers.

The entries in the *Shadow* menu are the same as for point and line layers.

In the *Placement* menu you find special settings for polygon layers (see Figure_labels_3). *Offset from centroid*, *Horizontal (slow)*, *Around centroid*, *Free* and *Using perimeter* are possible.

In the *Offset from centroid* settings you can define if the centroid is *visible polygon* or *whole polygon*. That means that either the centroid is used for the polygon you can see on the map or the centroid is used for the whole polygon, no matter if you can see the whole feature on the map. You can place your label with the quadrants here and define offset and rotation. The *Around centroid* setting makes it possible to place the label around the centroid with a certain distance. Again, you can define *visible polygon* or *whole polygon* for the centroid. With the *Using perimeter* settings you can define a position and a distance for the label. For the position *Above line*, *On line*, *Below line* and *Line orientation dependend position* are possible.

The entries in the *Rendering* menu are the same as for line layers. You can also use *Suppress labeling of features smaller than* in the *Feature options*.

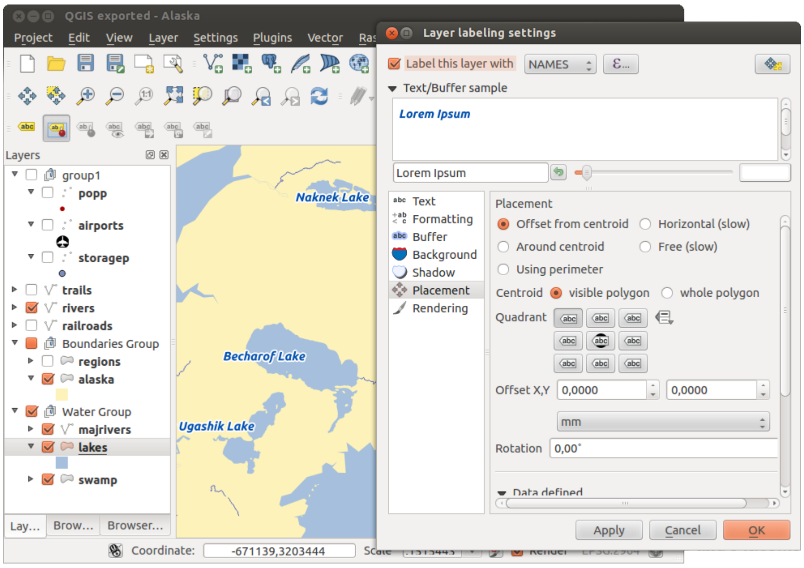


Figure 12.16: Smart labeling of vector polygon layers

Using data-defined override for labeling

With the data-defined override functions the settings for the labeling are overwritten by entries in the attribute table. You can activate/deactivate the function with the right-mouse button. Hover over the symbol and you see the information about the data-defined override, including the current definition field. We now describe an example how to use the data-defined override function for the Move label function (see figure_labels_4).

1. Import the lakes.shp from the QGIS sample dataset.
2. Double-klick the layer to open the Layer Properties. Klick on *Labels* and *Placement*. Select *Offset from centroid*.
3. Look for the *Data defined* entries. Klick the -Icon to define the field type for the *Coordinate*. Choose 'xlabel' for X and 'ylabel' for Y. The Icons are now highlighted in yellow.
4. Zoom into a lake.

5. Go to the Label toolbar and klick the Icon. Now you can shift the label manually to another position (see figure_labels_5). The new position of the label is saved in the 'xlabel' and 'ylabel' columns of the attribute table.

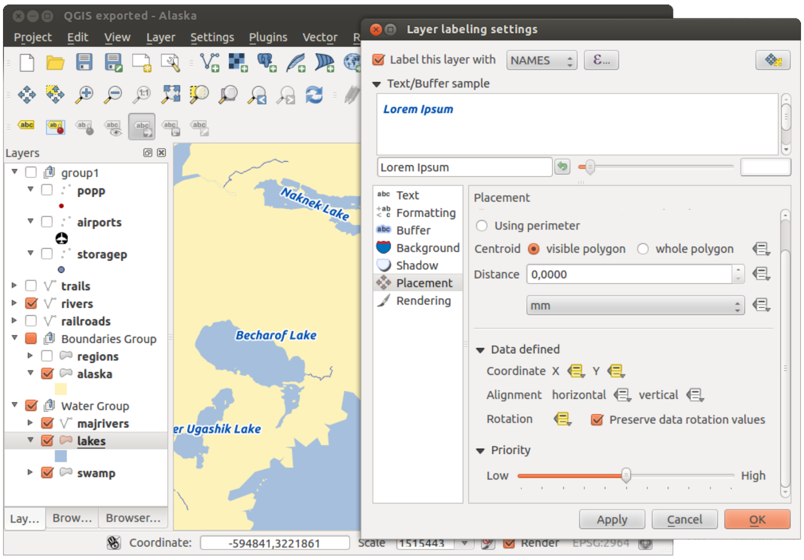


Figure 12.17: Labeling of vector polygon layers with data-defined override

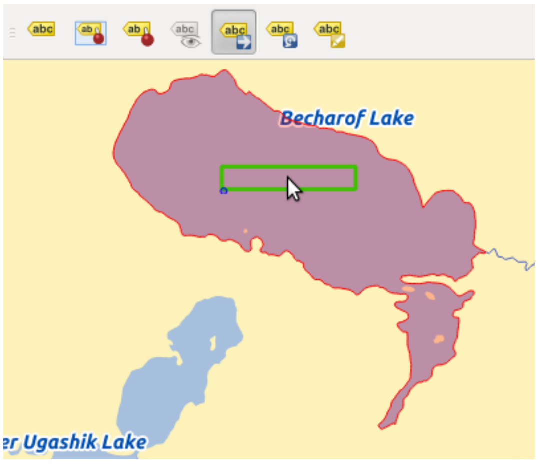


Figure 12.18: Move labels

12.2.3 Fields Menu

Within the *Fields* menu the field attributes of the selected dataset can be manipulated. The buttons New Column and Delete Column can be used, when the dataset is Editing mode.

Edit Widget

Within the *Fields* menu you also find an **edit widget** column. This column can be used to define values or a range of values that are allowed to be added to the specific attribute table column. If you click on the **[edit widget]** button, a dialog opens, where you can define different widgets. These widgets are:

- **Line edit**: an edit field which allows to enter simple text (or restrict to numbers for numeric attributes).
- **Classification**: Displays a combo box with the values used for classification, if you have chosen 'unique value' as legend type in the *Style* menu of the properties dialog.

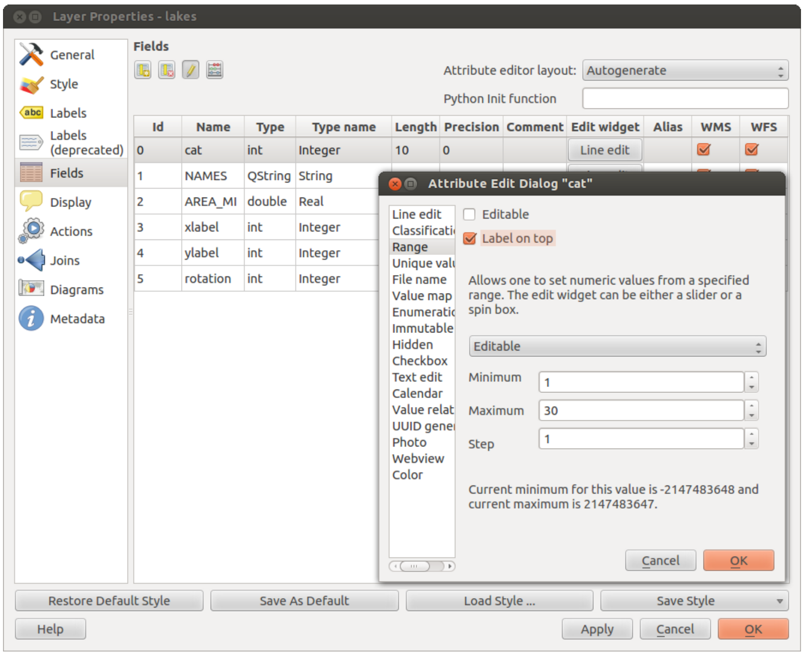


Figure 12.19: Dialog to select an edit widget for an attribute column

- **Range**: Allows to set numeric values from a specific range. The edit widget can be either a slider or a spin box.
- **Unique values**: The user can select one of the values already used in the attribute table. If editable is activated, a line edit is shown with autocompletion support, otherwise a combo box is used.
- **File name**: Simplifies the selection by adding a file chooser dialog.
- **Value map**: a combo box with predefined items. The value is stored in the attribute, the description is shown in the combo box. You can define values manually or load them from a layer or a CSV file.
- **Enumeration**: Opens a combo box with values that can be used within the columns type. This is currently only supported by the postgres provider.
- **Immutable**: The immutable attribute column is read-only. The user is not able to modify the content.
- **Hidden**: A hidden attribute column is invisible. The user is not able to see its content.
- **Checkbox**: Displays a checkbox and you can define what attribute is added to the column when the checkbox is activated or not.
- **Text edit**: This opens a text edit field that allows multiple lines to be used.
- **Calendar**: Opens a calendar widget to enter a date. Column type must be text.
- **Value Relation**: Offers values from a related table in a combobox. You can select layer, key column and value column.
- **UUID Generator**: Generates a read-only UUID (Universally Unique Identifiers) field, if empty.
- **Photo**: Field contains a filename for a picture. The width and height of the field can be defined.
- **Webview**: Field contains an URL. The width and height of the field is variable.
- **Color**: A field which allows to enter color codes. During data entry the color is visible through a color bar included in the field.

With the **Attribute editor layout** you can now define builtin forms for data entry jobs (see figure_fields_2). Choose 'Drag and drop designer' and an attribute column. Use the Icon to create a category that then will be shown during the digitizing session (see figure_fields_3). Next step will be to assign the relevant fields to the category with the  Icon. You can create more categories and use the same fields again. When creating a new category QGIS will insert a new tab for the category in the built in form.

Other options in the dialog are ‘Autogenerate’ and ‘Provide ui-file’. ‘Autogenerate’ just creates Editors for all fields and tabulates them. The ‘Provide ui-file’ option allows you to use complex dialogs made with the Qt-Designer. Using an UI-file allows a large freedom in creating a dialog. For detailed information see http://nathanw.net/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/ .

QGIS dialogs can have a python function that is called when the dialog is opened. Use this function to add extra logic to your dialogs. An example is (in module MyForms.py):

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

Reference in Python Init Function like so: MyForms.open

MyForms.py must live on PYTHONPATH, .qgis2/python, or inside the project folder

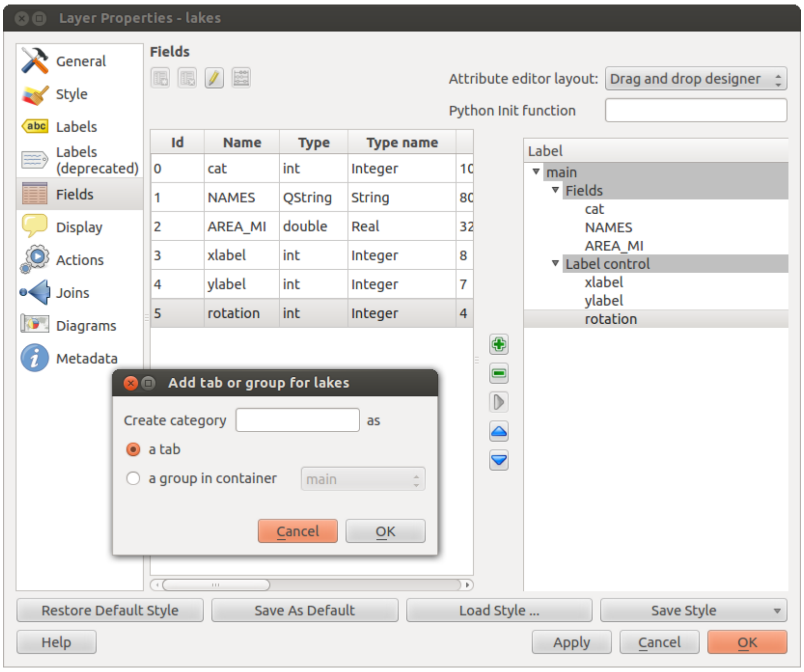


Figure 12.20: Dialog to create categories with the **Attribute editor layout**

12.2.4 General Menu

 Use this menu to make general settings for the vector layer. There are several options available:

Layer Info

- Change the display name of the layer in *displayed as*
- Define the *Layer source* of the vector layer
- Define the *Data source encoding* to define provider specific option and to be able to read the file

Coordinate Reference System

- *Specify* the Coordinate Reference System. Here you can view or change the projection of the specific vector layer.
- Create a *Spatial Index* (only for OGR supported formats)

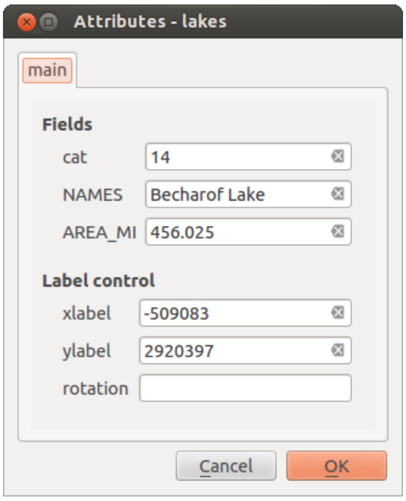


Figure 12.21: Resulting built in form in a data entry session

- *Update Extents* information for a layer
- View or change the projection of the specific vector layer, clicking on *Specify ...*

Scale dependent visibility

- You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale. The scale can also be set by the **[Current]** buttons

Feature subset

- With the **[Query Builder]** button you can create a subset of the features in the layer that will be visualized (also refer to section *Save selected features as new layer*).

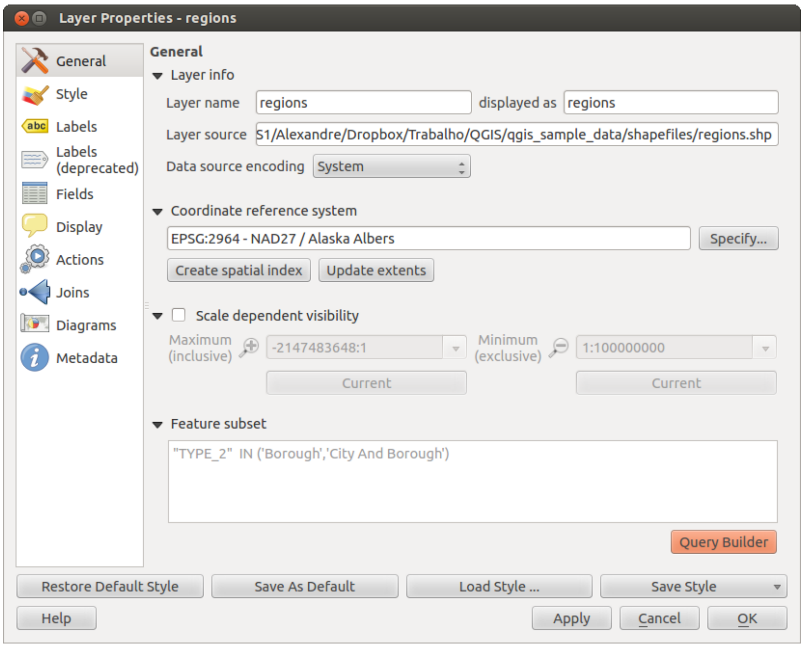


Figure 12.22: General menu in vector layers properties dialog

12.2.5 Display Menu

In QGIS 2.0 there is now an own menu for the map tips. It includes a new feature: Map Tip display text in HTML. While you can still choose a *Field* to be displayed when hovering over a feature on the map it is now possible to insert HTML code that creates a complex display when hovering over a feature. To activate Map Tips, select the menu option *View* → *MapTips*.Figure Display 1 shows an example of HTML code.

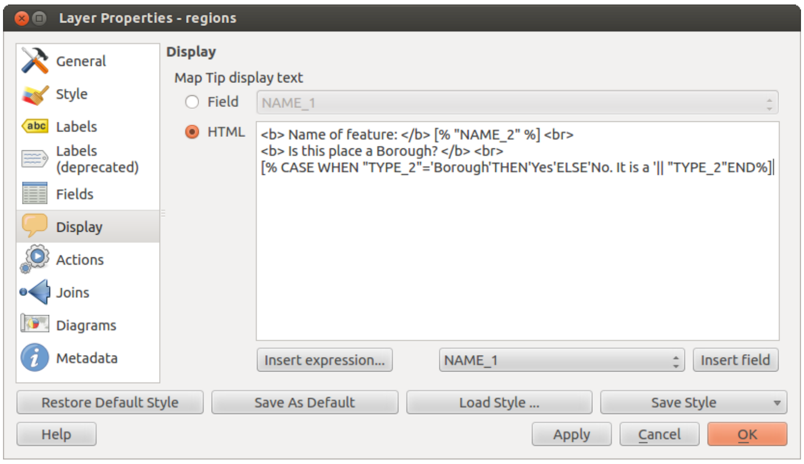


Figure 12.23: HTML code for map tip

Figure 12.24: Map tip made with HTML code

12.2.6 Actions Menu

QGIS provides the ability to perform an action based on the attributes of a feature. This can be used to perform any number of actions, for example, running a program with arguments built from the attributes of a feature or passing parameters to a web reporting tool.

Actions are useful when you frequently want to run an external application or view a web page based on one or more values in your vector layer. They are devided into 6 types and can be used like this:

- Generic, Mac, Windows and Unix actions start an external process,
- Python actions execute a python expression,
- Generic and Python actions are visible everywhere,

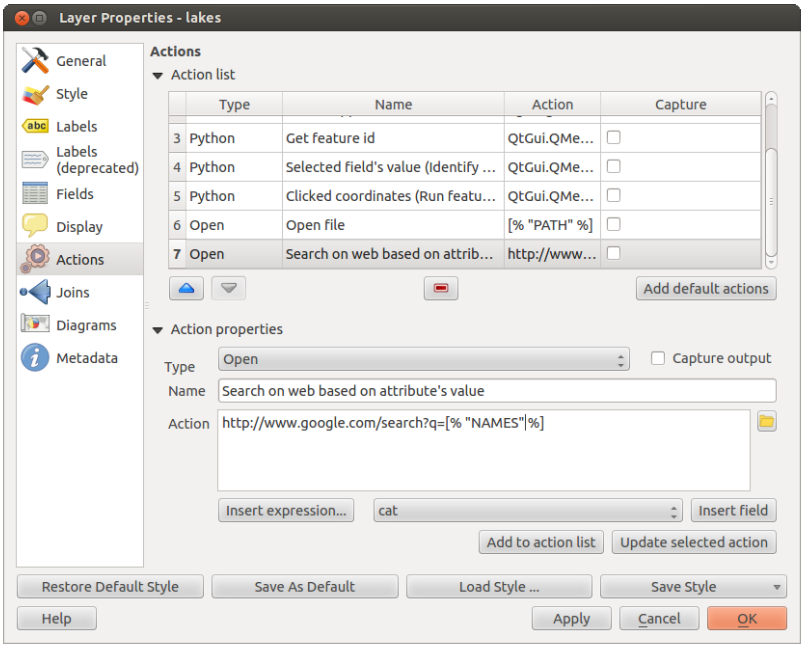


Figure 12.25: Overview action dialog with some sample actions

- Mac, Windows and Unix actions are visible only on the respective platform (i.e. you can define three 'Edit' actions to open an editor and the users can only see and execute the one 'Edit' action for their platform to run the editor).

There are several examples included in the dialog. You can load them clicking on **[Add default actions]**. An example is performing a search based on an attribute value. This concept is used in the following discussion.

Defining Actions

Attribute actions are defined from the vector *Layer Properties* dialog. To define an action, open the vector *Layer Properties* dialog and click on the *Actions* menu. Go to the *Action properties*. Select 'Generic' as type and provide a descriptive name for the action. The action itself must contain the name of the application that will be executed when the action is invoked. You can add one or more attribute field values as arguments to the application. When the action is invoked any set of characters that start with a `%` followed by the name of a field will be replaced by the value of that field. The special characters %% will be replaced by the value of the field that was selected from the identify results or attribute table (see using_actions below). Double quote marks can be used to group text into a single argument to the program, script or command. Double quotes will be ignored if preceded by a backslash.

If you have field names that are substrings of other field names (e.g., `col1` and `col10`) you should indicate so, by surrounding the field name (and the % character) with square brackets (e.g., `[%col10]`). This will prevent the `%col10` field name being mistaken for the `%col1` field name with a `0` on the end. The brackets will be removed by QGIS when it substitutes in the value of the field. If you want the substituted field to be surrounded by square brackets, use a second set like this: `[[%col10]]`.

Using the *Identify Features* tool you can open *Identify Results* dialog. It includes a *(Derived)* item that contains information relevant to the layer type. The values in this item can be accessed in a similar way to the other fields by using preceeding the derived field name by `(Derived).`. For example, a point layer has an `X` and `Y` field and the value of these can be used in the action with `%(Derived).X` and `%(Derived).Y`. The derived attributes are only available from the *Identify Results* dialog box, not the *Attribute Table* dialog box.

Two example actions are shown below:

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

In the first example, the web browser konqueror is invoked and passed a URL to open. The URL performs a Google search on the value of the `nam` field from our vector layer. Note that the application or script called by the action must be in the path or you must provide the full path. To be sure, we could rewrite the first example as: `/opt/kde3/bin/konqueror http://www.google.com/search?q=%nam`. This will ensure that the konqueror application will be executed when the action is invoked.

The second example uses the %% notation which does not rely on a particular field for its value. When the action is invoked, the %% will be replaced by the value of the selected field in the identify results or attribute table.
Using Actions

Actions can be invoked from either the *Identify Results* dialog, an *Attribute Table* dialog or from *Run Feature Action* (recall that these dialogs can be opened by clicking Identify Features or Open Attribute Table or Run Feature Action). To invoke an action, right click on the record and choose the action from the popup menu. Actions are listed in the popup menu by the name you assigned when defining the actions. Click on the action you wish to invoke.

If you are invoking an action that uses the `%%` notation, right-click on the field value in the *Identify Results* dialog or the *Attribute Table* dialog that you wish to pass to the application or script.

Here is another example that pulls data out of a vector layer and inserts them into a file using bash and the `echo` command (so it will only work or perhaps). The layer in question has fields for a species name `taxon_name`, latitude `lat` and longitude `long`. I would like to be able to make a spatial selection of a localities and export these field values to a text file for the selected record (shown in yellow in the QGIS map area). Here is the action to achieve this:

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

After selecting a few localities and running the action on each one, opening the output file will show something like this:

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

As an exercise we create an action that does a Google search on the `lakes` layer. First we need to determine the URL needed to perform a search on a keyword. This is easily done by just going to Google and doing a simple search, then grabbing the URL from the address bar in your browser. From this little effort we see that the format is: http://google.com/search?q=qgis, where `QGIS` is the search term. Armed with this information, we can proceed:

1. Make sure the `lakes` layer is loaded.
2. Open the *Layer Properties* dialog by double-clicking on the layer in the legend or right-click and choose *Properties* from the popup menu.
3. Click on the *Actions* menu.
4. Enter a name for the action, for example `Google Search`.
5. For the action, we need to provide the name of the external program to run. In this case, we can use Firefox. If the program is not in your path, you need to provide the full path.
6. Following the name of the external application, add the URL used for doing a Google search, up to but not included the search term: `http://google.com/search?q=`
7. The text in the *Action* field should now look like this: `firefox http://google.com/search?q=`
8. Click on the drop-down box containing the field names for the `lakes` layer. It's located just to the left of the **[Insert Field]** button.
9. From the drop-down box select 'NAMES' and click **[Insert Field]**.
10. Your action text now looks like this:

    `firefox http://google.com/search?q=%NAMES`
11. To finalize the action click the **[Add to action list]** button.

This completes the action and it is ready to use. The final text of the action should look like this:

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

We can now use the action. Close the *Layer Properties* dialog and zoom in to an area of interest. Make sure the `lakes` layer is active and identify a lake. In the result box you'll now see that our action is visible:

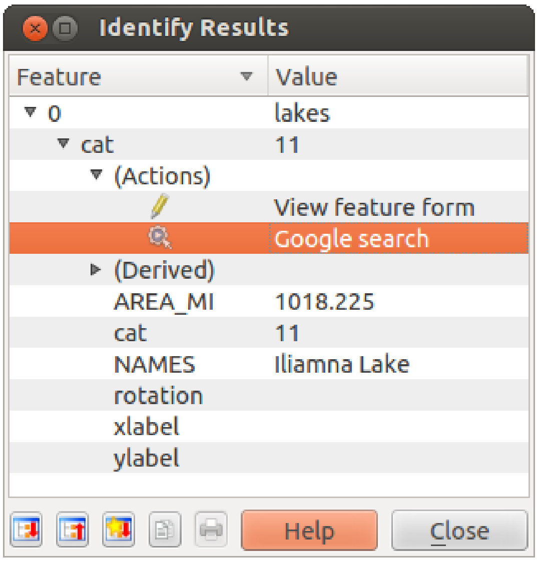


Figure 12.26: Select feature and choose action

When we click on the action, it brings up Firefox and navigates to the URL http://www.google.com/search?q=Tustumena. It is also possible to add further attribute fields to the action. Therefore you can add a + to the end of the action text, select another field and click on **[Insert Field]**. In this example there is just no other field available that would make sense to search for.

You can define multiple actions for a layer and each will show up in the *Identify Results* dialog.

You can think of all kinds of uses for actions. For example, if you have a point layer containing locations of images or photos along with a file name, you could create an action to launch a viewer to display the image. You could also use actions to launch web-based reports for an attribute field or combination of fields, specifying them in the same way we did in our Google search example.

We can also make more complex examples, for instance on how to use **Python** actions.

Usually when we create an action to open a file with an external application we can use absolute paths, or eventually relative paths, in the second case the path is relative to the location of the external program executable file. But what about we need to use relative paths, relative to the selected layer (a file based one, like a shapefile or spatialite)? The following code will do the trick:

```
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 ] );
```

we have to just remember that the action is one of type *Python* and to change the *command* and *imagerelpath* variables to fit our needs.

But what about if the relative path need to be relative to the (saved) project file? The code of the Python action would be:

```
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 ] );
```

Another Python actions example if the one that allows us to add new layers to the project. For instance the following examples will add to the project respectively a vector and a raster. The name of files to be added to the project and the name to be given to the layer are data driven (*filename* and *layname* are column names of the table of attributes of the vector where the action was created):

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

To add a raster (a tif image in this example) it becomes:

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

12.2.7 Joins Menu

The *Joins* menu allows you to join a loaded attribute table to a loaded vector layer. After clicking the *Add vector join* dialog appears. As key columns you have to define a join layer you want to connect with the target vector layer , a join field that corresponds to an attribute column in the target layer and a target field you find in the attribute table of the target vector layer here. As a result, all information of the join layer and the target layer are displayed in the attribute table of the target layer as joined information.

QGIS currently supports to join non spatial table formats supported by OGR (e.g. CSV, DBF and Excel), delimited text and the PostgreSQL provider (see figure_joins_1).

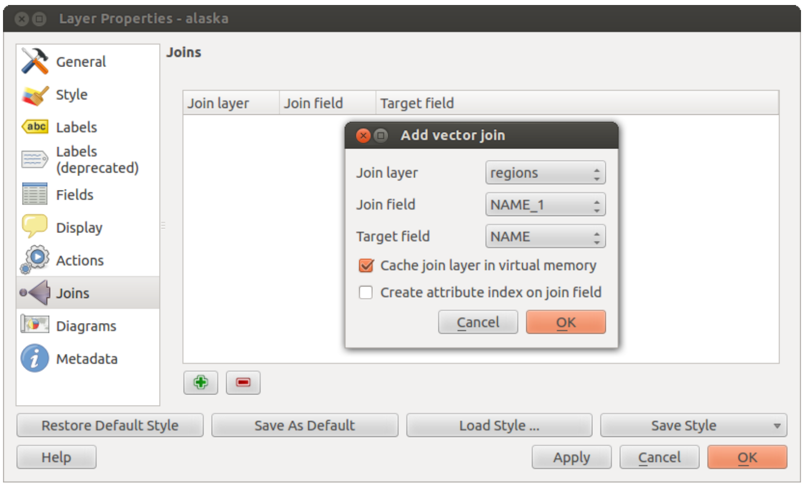


Figure 12.27: Join an attribute table to an existing vector layer

Additionally the add vector join dialog allows to:

- *Cache join layer in virtual memory*
- *Create attribute index on the join field*

12.2.8 Diagrams Menu

The *Diagrams* menu allows you to add a graphic overlay to a vector layer (see figure_diagrams_1).

The current core implementation of diagrams provides support for piecharts, text diagrams and histograms.

The menu is divided into four tabs now: *Appearance*, *Size*, *Postion* and *Options*.

In the case of the text diagram and piechart text values of different data columns are displayed one below the other with a circle or a box and dividers. In the *Size* tab diagram size is based on a fixed size or on linear scaling according to a classification attribute. The placement of the diagrams which is done in the *Position* tab interacts with the new labeling, so position conflicts between diagrams and labels are detected and solved. In addition to chart positions can be fixed by the users hand.

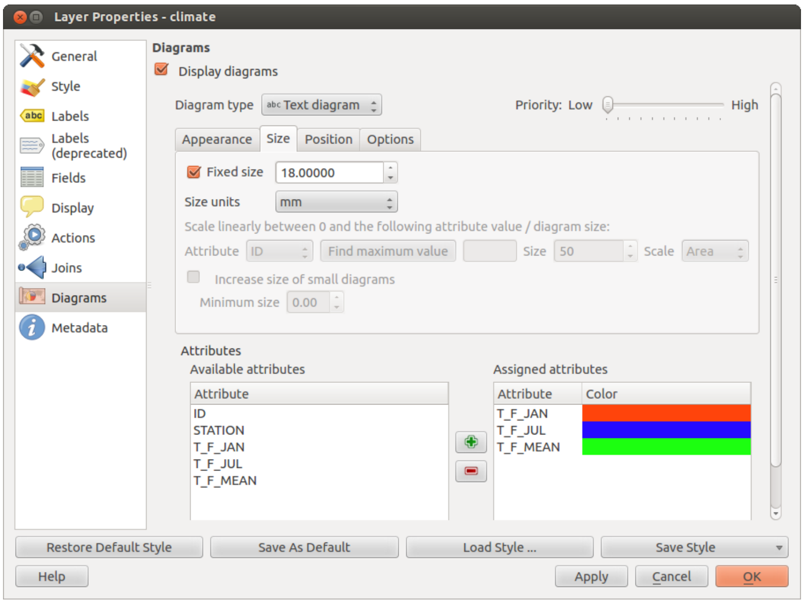


Figure 12.28: Vector properties dialog with diagram menu

We will demonstrate an example and overlay the alaska boundary layer a text diagram showing some temperature data from a climate vector layer. Both vector layers are part of the QGIS sample dataset (see Section *Sample Data*).

1. First click on the Load Vector icon, browse to the QGIS sample dataset folder and load the two vector shape layers `alaska.shp` and `climate.shp`.
2. Double click the `climate` layer in the map legend to open the *Layer Properties* dialog.
3. Click on the *Diagrams* menu, activate *Display diagrams* and from *Diagram type* combobox select 'Text diagram'
4. In the *Appearance* tab we choose a light blue as Background color and in the *Size* tab we set a fixed size to 18 mm.
5. In the *Position* tab Placement could be set to AroundPoint.
6. In the diagram we want to display the values of the three columns `T_F_JAN`, `T_F_JUL` and `T_F_MEAN`. First select `T_F_JAN` as *Attributes* and click the button, then `T_F_JUL` and finally `T_F_MEAN`.
7. Now click **[Apply]** to display the diagram in the QGIS main window.
8. You can now adapt the chart size in the *Size* tab. Deactivate the *Fixed size* and set the size of the diagrams on the basis of an Attribute with the **[Find maximum value]** button and the *Size* menu. If diagrams appear

too small on the screen you can activate the *Increase size of small diagrams* checkbox and define the Minimum size of the diagrams.

9. Change the Attribute Colors by double clicking on the color values in the *Assigned attributes* field. Figure_diagrams_2 gives an impression.

10. Finally click **[Ok]**.

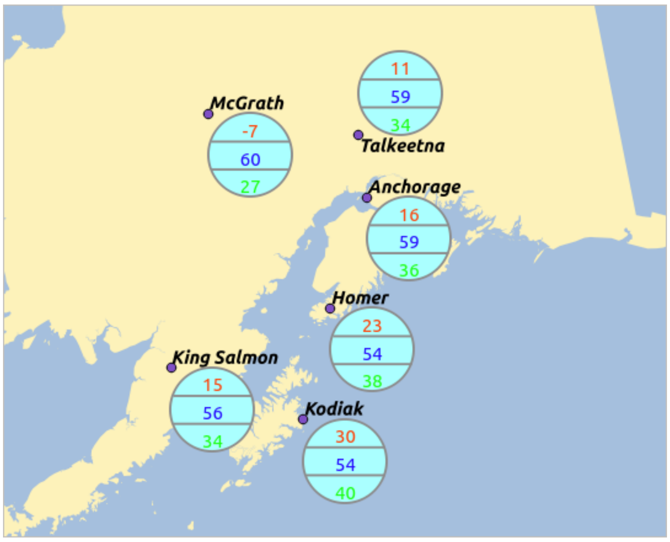


Figure 12.29: Diagram from temperature data overlayed on a map

Remember that in the *Position* tab a *Data defined position* of the diagrams is possible. Here you can use attributes to define the position of the diagram. Also, a scale dependent visibility that you can find in the *Appearance* tab is possible.

12.2.9 Metadata Menu

The *Metadata* menu consists of a *Description*, *Attribution*, *MetadataURL* and *Properties* section.

In the *Properties* section you get general information about the layer, including specifics about the type and location, number of features, feature type, and the editing capabilities in the *Properties* section. The *Extents* table provides you with layer extent information, and the *Layer Spatial Reference System* information, providing information about the CRS of the layer. This is a quick way to get information about the layer.

Additionally you can add/edit a title for the layer and some abstract information in the *Description*. Also, it's possible to define a *Keyword list* here. These keyword lists can be used in a metadata catalogue. If you want to use a title from an XML metadata file you have to fill in a link in the *DataUrl* field. Use *Attribution* to get Atrribute data from an XML metadata catalogue. In *MetadataUrl* you can define the general path to the XML metadata catalogue. These information will be saved in the QGIS project file for following sessions and will be used for QGIS server.

12.3 Editing

QGIS supports various capabilities for editing OGR, SpatiaLite, PostGIS, MSSQL Spatial and Oracle Spatial vector layers and tables.

Note: The procedure for editing GRASS layers is different - see Section *Digitizing and editing a GRASS vector layer* for details.

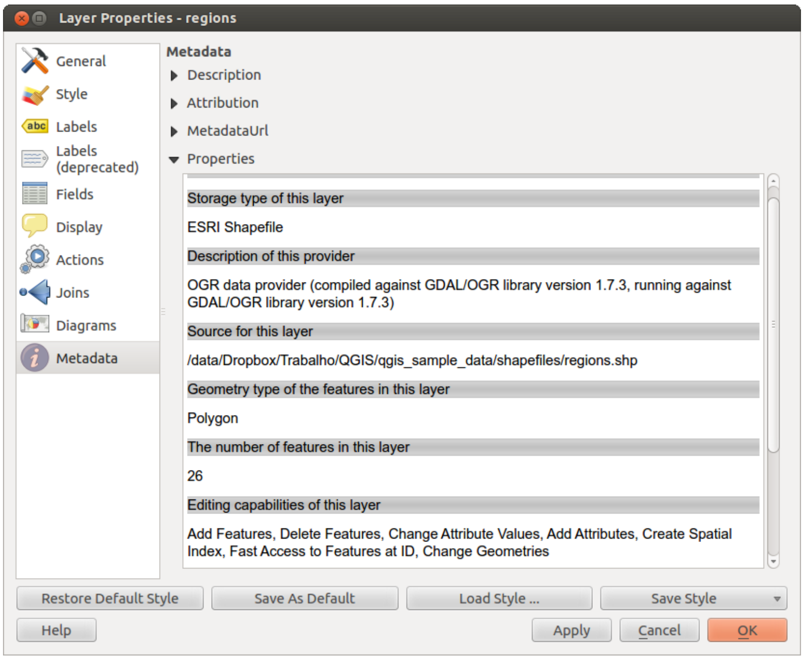


Figure 12.30: Metadata menu in vector layers properties dialog

Tip: Concurrent Edits

This version of QGIS does not track if somebody else is editing a feature at the same time as you. The last person to save their edits wins.

12.3.1 Setting the Snapping Tolerance and Search Radius

Before we can edit vertices, we must set the snapping tolerance and search radius to a value that allows us an optimal editing of the vector layer geometries.

Snapping tolerance

Snapping tolerance is the distance QGIS uses to `search` for the closest vertex and/or segment you are trying to connect when you set a new vertex or move an existing vertex. If you aren't within the snapping tolerance, QGIS will leave the vertex where you release the mouse button, instead of snapping it to an existing vertex and/or segment. The snapping tolerance setting affects all tools which work with tolerance.

1. A general, project wide snapping tolerance can be defined choosing *Settings* → *Options*. On Mac: go to *QIS* → *Preferences...*, on Linux: *Edit* → *Options*. In the *Digitizing* tab you can select between to vertex, to segment or to vertex and segment as default snap mode. You can also define a default snapping tolerance and a search radius for vertex edits. The tolerance can be set either in map units or in pixels. The advantage of choosing pixels, is that the snapping tolerance doesn't have to be changed after zoom operations. In our small digitizing project (working with the Alaska dataset), we define the snapping units in feet. Your results may vary, but something on the order of 300ft should be fine at a scale of 1:10 000 should be a reasonable setting.
2. A layer based snapping tolerance can be defined by choosing *Settings* → (or *File* →) *Snapping options...* to enable and adjust snapping mode and tolerance on a layer basis (see figure_edit_1).

Note that this layer based snapping overrides the global snapping option set in the Digitizing tab. So if you need to edit one layer, and snap its vertices to another layer, then enable snapping only on the `snap to` layer, then decrease the global snapping tolerance to a smaller value. Furthermore, snapping will never occur to a layer which is not checked in the snapping options dialog, regardless of the global snapping tolerance. So be sure to mark the checkbox for those layers that you need to snap to.

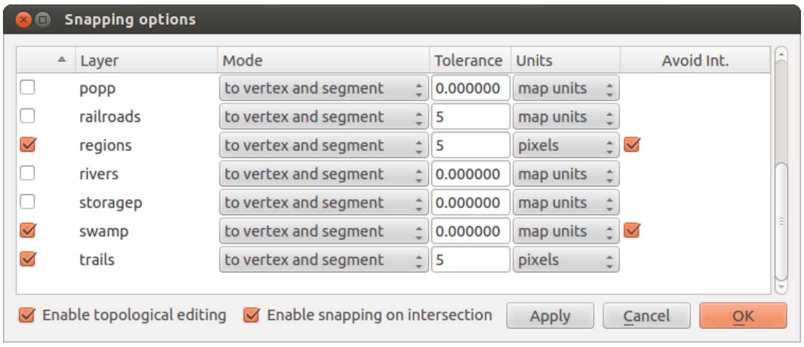


Figure 12.31: Edit snapping options on a layer basis

Search radius

Search radius is the distance QGIS uses to `search` for the closest vertex you are trying to move when you click on the map. If you aren't within the search radius, QGIS won't find and select any vertex for editing and it will pop up an annoying warning to that effect. Snap tolerance and search radius are set in map units or pixels, so you may find you need to experiment to get them set right. If you specify too big of a tolerance, QGIS may snap to the wrong vertex, especially if you are dealing with a large number of vertices in close proximity. Set search radius too small and it won't find anything to move.

The search radius for vertex edits in layer units can be defined in the *Digitizing* tab under *Settings* → *Options*. The same place where you define the general, project wide snapping tolerance.

12.3.2 Zooming and Panning

Before editing a layer, you should zoom in to your area of interest. This avoids waiting while all the vertex markers are rendered across the entire layer.

Apart from using the pan and zoom-in / zoom-out icons on the toolbar with the mouse, navigating can also be done with the mouse wheel, spacebar and the arrow keys.

Zooming and panning with the mouse wheel

While digitizing you can press the mouse wheel to pan inside of the main window and you can roll the mouse wheel to zoom in and out on the map. For zooming place the mouse cursor inside the map area and roll it forward (away from you) to zoom in and backwards (towards you) to zoom out. The mouse cursor position will be the center of the zoomed area of interest. You can customize the behavior of the mouse wheel zoom using the *Map tools* tab under the *Settings* → *Options* menu.

Panning with the arrow keys

Panning the map during digitizing is possible with the arrow keys. Place the mouse cursor inside the map area and click on the right arrow key to pan east, left arrow key to pan west, up arrow key to pan north and down arrow key to pan south.

You can also use the spacebar to temporarily cause mouse movements to pan then map. The `PgUp` and `PgDown` keys on your keyboard will cause the map display to zoom in or out without interrupting your digitizing session.

12.3.3 Topological editing

Besides layer based snapping options you can also define some topological functionalities in the *Snapping options...* dialog in the *Settings* (or *File*) menu. Here you can define *Enable topological editing* and/or for polygon layers you can activate the column *Avoid Int.* which avoids intersection of new polygons.

Enable topological editing

The option *Enable topological editing* is for editing and maintaining common boundaries in polygon mosaics. QGIS 'detects' a shared boundary in a polygon mosaic and you only have to move the vertex once and QGIS will take care about updating the other boundary.

Avoid intersections of new polygons

The second topological option in the *Avoid Int.* column, called *Avoid intersections of new polygons* avoids overlaps in polygon mosaics. It is for quicker digitizing of adjacent polygons. If you already have one polygon, it is possible with this option to digitize the second one such that both intersect and QGIS then cuts the second polygon to the common boundary. The advantage is that users don't have to digitize all vertices of the common boundary.

Enable snapping on intersections

Another option is to use *Enable snapping on intersection*. It allows to snap on an intersection of background layers, even if there's no vertex on the intersection.

12.3.4 Digitizing an existing layer

By default, QGIS loads layers read-only: This is a safeguard to avoid accidentally editing a layer if there is a slip of the mouse. However, you can choose to edit any layer as long as the data provider supports it, and the underlying data source is writable (i.e. its files are not read-only).

In general, editing vector layers is divided into a digitizing and an advanced digitizing toolbar, described in Section *Advanced digitizing*. You can select and unselect both under *Settings* → *Toolbars* →. Using the basic digitizing tools you can perform the following functions:

Icon	Purpose	Icon	Purpose
	Current edits		Toggle editing
	Adding Features: Capture Point		Adding Features: Capture Line
	Adding Features: Capture Polygon		Move Feature
	Node Tool		Delete Selected
	Cut Features		Copy Features
	Paste Features		Save layer edits

Table Editing: Vector layer basic editing toolbar

All editing sessions start by choosing the Toggle editing option. This can be found in the context menu after right clicking on the legend entry for that layer.

Alternately, you can use the Toggle Editing Toggle editing button from the digitizing toolbar to start or stop the editing mode. Once the layer is in edit mode, markers will appear at the vertices, and additional tool buttons on the editing toolbar will become available.

Tip: Save Regularly

Remember to Save Layer Edits regularly. This will also check that your data source can accept all the changes.

Adding Features

You can use the Add Feature, Add Feature or Add Feature icons on the toolbar to put the QGIS cursor into digitizing mode.

For each feature, you first digitize the geometry, then enter its attributes. To digitize the geometry, left-click on the map area to create the first point of your new feature.

For lines and polygons, keep on left-clicking for each additional point you wish to capture. When you have finished adding points, right-click anywhere on the map area to confirm you have finished entering the geometry of that feature.

The attribute window will appear, allowing you to enter the information for the new feature. Figure_edit_2 shows setting attributes for a fictitious new river in Alaska. In the *Digitizing* menu under the *Settings* → *Options* menu, you can also activate *Suppress attributes pop-up windows after each created feature* *Reuse last entered attribute values*.

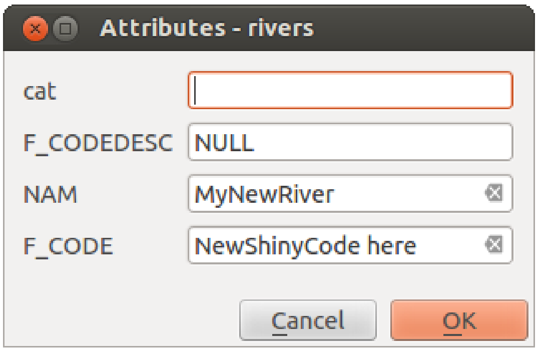


Figure 12.32: Enter Attribute Values Dialog after digitizing a new vector feature

With the Move Feature(s) icon on the toolbar you can move existing features.

Tip: Attribute Value Types

For editing the attribute types are validated during the entry. Because of this, it is not possible to enter a number into the text-column in the dialog *Enter Attribute Values* or vice versa. If you need to do so, you should edit the attributes in a second step within the *Attribute table* dialog.

Current Edits

This new feature allows the digitization of multiple layers. Choose *Save for Selected Layers* to save all changes you made in multiple layers. You also have the opportunity to *Rollback for Selected Layers* so that the digitization is then withdrawn for all selected layers. If you want to stop editing the selected layers the *Cancel for Selected Layer(s)* is an easy way.

The same functions for editing all layers of the project are available.

Node Tool

For shapefile-based layers as well as SpatialLite,PostgreSQL/PostGIS, MSSQL Spatial and Oracle Spatial tables the Node Tool provides manipulation capabilites of feature vertices similar to CAD programs. It is possible to simply select multiple vertices at once and to move, add or delete them alltogether. The node tool also works with 'on the fly' projection turned on and supports the topological editing feature. This tool is, unlike other tools in

QGIS, persistent, so when some operation is done, selection stays active for this feature and tool. If the node tool couldn't find any features, a warning will be displayed.

Important is to set the property *Settings* → *Options* → *Digitizing* → *Search Radius:* 1,00 to a number greater than zero (i.e. 10). Otherwise QGIS will not be able to tell which vertex is being edited.

Tip: Vertex Markers

The current version of QGIS supports three kinds of vertex-markers: Semi transparent circle, Cross and None. To change the marker style, choose *Options* from the *Settings* menu and click on the *Digitizing* tab and select the appropriate entry.

Basic operations

Start by activating the Node Tool and selecting a feature by clicking on it. Red boxes will appear at each vertex of this feature.

- **Selecting vertices**: You can select vertices by clicking on them one at a time, by clicking on an edge to select the vertices at both ends, or by clicking and dragging a rectangle around some vertices. When a vertex is selected its color changes to blue. To add more vertices to the current selection, hold down the `Ctrl` key while clicking. Hold down `Ctrl` or `Shift` when clicking to toggle the selection state of vertices (vertices that are currently unselected will be selected as usual, but also vertices that are already selected will become unselected).
- **Adding vertices**: To add a vertex simply double click near an edge and a new vertex will appear on the edge near to the cursor. Note that the vertex will appear on the edge, not at the cursor position, therefore it has to be moved if necessary.
- **Deleting vertices**: After selecting vertices for deletion, click the `Delete` key. Note that you cannot use the Node Tool to delete a complete feature; QGIS will ensure it retains the minimum number of vertices for the feature type you are working on. To delete a complete feature use the Delete Selected tool.
- **Moving vertices**: Select all the vertices you want to move. Click on a selected vertex or edge and drag in the direction you wish to move. All the selected vertices will move together. If snapping is enabled, the whole selection can jump to the nearest vertex or line.

Each change made with the node tool is stored as a separate entry in the undo dialog. Remember that all operations support topological editing when this is turned on. On the fly projection is also supported, and the node tool provides tooltips to identify a vertex by hovering the pointer over it.

Cutting, Copying and Pasting Features

Selected features can be cut, copied and pasted between layers in the same QGIS project, as long as destination layers are set to Toggle editing beforehand.

Features can also be pasted to external applications as text: That is, the features are represented in CSV format with the geometry data appearing in the OGC Well-Known Text (WKT) format.

However in this version of QGIS, text features from outside QGIS cannot be pasted to a layer within QGIS. When would the copy and paste function come in handy? Well, it turns out that you can edit more than one layer at a time and copy/paste features between layers. Why would we want to do this? Say we need to do some work on a new layer but only need one or two lakes, not the 5,000 on our `big_lakes` layer. We can create a new layer and use copy/paste to plop the needed lakes into it.

As an example we are copying some lakes to a new layer:

1. Load the layer you want to copy from (source layer)
2. Load or create the layer you want to copy to (target layer)

3. Start editing for target layer
4. Make the source layer active by clicking on it in the legend
5. Use the Select Single Feature tool to select the feature(s) on the source layer
6. Click on the Copy Features tool
7. Make the destination layer active by clicking on it in the legend
8. Click on the Paste Features tool
9. Stop editing and save the changes

What happens if the source and target layers have different schemas (field names and types are not the same)? QGIS populates what matches and ignores the rest. If you don't care about the attributes being copied to the target layer, it doesn't matter how you design the fields and data types. If you want to make sure everything - feature and its attributes - gets copied, make sure the schemas match.

Tip: Congruency of Pasted Features

If your source and destination layers use the same projection, then the pasted features will have geometry identical to the source layer. However if the destination layer is a different projection then QGIS cannot guarantee the geometry is identical. This is simply because there are small rounding-off errors involved when converting between projections.

Deleting Selected Features

If we want to delete an entire polygon, we can do that by first selecting the polygon using the regular Select Single Feature tool. You can select multiple features for deletion. Once you have the selection set, use the Delete Selected tool to delete the features.

The Cut Features tool on the digitizing toolbar can also be used to delete features. This effectively deletes the feature but also places it on a "spatial clipboard". So we cut the feature to delete. We could then use the Paste Features tool to put it back, giving us a one-level undo capability. Cut, copy, and paste work on the currently selected features, meaning we can operate on more than one at a time.

Saving Edited Layers

When a layer is in editing mode, any changes remain in the memory of QGIS. Therefore they are not committed/saved immediately to the data source or disk. If you want to save edits to the current layer but want to continue editing without leaving the editing mode, you can click the Save Layer Edits button. When you turn editing mode off with the Toggle editing (or quit QGIS for that matter), you are also asked if you want to save your changes or discard them.

If the changes cannot be saved (e.g. disk full, or the attributes have values that are out of range), the QGIS in-memory state is preserved. This allows you to adjust your edits and try again.

Tip: Data Integrity

It is always a good idea to back up your data source before you start editing. While the authors of QGIS have made every effort to preserve the integrity of your data, we offer no warranty in this regard.

12.3.5 Advanced digitizing

Icon	Purpose	Icon	Purpose
	Undo		Redo
	Rotate Feature(s)		Simplify Feature
	Add Ring		Add Part
	Delete Ring		Delete Part
	Reshape Features		Offset Curve
	Split Features		Merge Selected Features
	Rotate Point Symbols		Merge Attributes of Selected Features

Table Advanced Editing: Vector layer advanced editing toolbar

Undo and Redo

The Undo and Redo tools allow the user to undo or redo vector editing operations. There is also a dockable widget, which shows all operations in the undo/redo history (see Figure_edit_3). This widget is not displayed by default; it can be displayed by right clicking on the toolbar and activating the Undo/Redo check box. Undo/Redo is however active, even if the widget is not displayed.

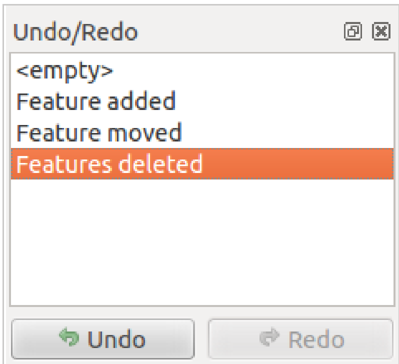


Figure 12.33: Redo and Undo digitizing steps

When Undo is hit, the state of all features and attributes are reverted to the state before the reverted operation happened. Changes other than normal vector editing operations (for example changes done by a plugin), may or may not be reverted, depending on how the changes were performed.

To use the undo/redo history widget simply click to select an operation in the history list; all features will be reverted to the state they were in after the selected operation.

Rotate Feature(s)

Use the Rotate Feature(s) to rotate one or multiple selected features in the map canvas. You first need to select the features and then press the Rotate Feature(s) Icon. Then the centroid of the feature appears and will be the rotation anchor point. If you selected multiple features the rotation anchor point will be the common center of the features. Press and drag the left mouse button in the desired direction to rotate the selected features.

It's also possible to create a user-defined rotation anchor point by which the selected feature will rotate. Select the features to rotate and activate the Rotate Feature(s) Tool. Press and hold the `Ctrl` button and move the mouse pointer (without pressing the mouse button) to the place where you want the rotation anchor to be moved. Release the `Ctrl` button when the desired rotation anchor point is reached. Now press and drag the left mouse button in the desired direction to rotate the selected feature(s).

Simplify Feature

The Simplify Feature tool allows to reduce the number of vertices of a feature, as long as the geometry doesn't change. You need to select a feature, it will be highlighted by a red rubber band and a slider appears. Moving the slider, the red rubber band is changing its shape to show how the feature is being simplified. Clicking **[OK]** the new, simplified geometry will be stored. If a feature cannot be simplified (e.g. MultiPolygons), a message shows up.

Add Ring

You can create ring polygons using the Add Ring icon in the toolbar. This means inside an existing area it is possible to digitize further polygons, that will occur as a 'hole', so only the area in between the boundaries of the outer and inner polygons remain as a ring polygon.

Add Part

You can add part polygons to a selected multipolygon. The new part polygon has to be digitized outside the selected multipolygon.

Delete Ring

The Delete Ring tool allows to delete ring polygons inside an existing area. This tool only works with polygon layers. It doesn't change anything when it is used on the outer ring of the polygon. This tool can be used on polygon and multi-polygon features. Before you select the vertices of a ring, adjust the vertex edit tolerance.

Delete Part

The Delete Part tool allows to delete parts from multifeatures (e.g. to delete polygons from a multipolygon feature). It won't delete the last part of the feature, this last part will stay untouched. This tool works with all multi-part geometries point, line and polygon. Before you select the vertices of a part, adjust the vertex edit tolerance.

Reshape Features

You can reshape line and polygon features using the Reshape Features icon on the toolbar. It replaces the line or polygon part from the first to the last intersection with the original line. With polygons this can sometimes lead to unintended results. It is mainly useful to replace smaller parts of a polygon, not major overhauls and the reshape line is not allowed to cross several polygon rings as this would generate an invalid polygon.

For example, you can edit the boundary of a polygon with this tool. First, click in the inner area of the polygon next to the point where you want to add a new vertex. Then, cross the boundary and add the vertices outside the polygon. To finish, right-click in the inner area of the polygon. The tool will automatically add a node where the new line crosses the border. It is also possible to remove part of the area from the polygon, starting the new line outside the polygon, adding vertices inside, and ending the line outside the polygon with a right click.

Note: The reshape tool may alter the starting position of a polygon ring or a closed line. So the point that is represented 'twice' will not be the same any more. This may not be a problem for most applications, but it is something to consider.

Offset Curves

The Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (creates copies of the lines / rings and adds it to the the edited layer). It is thus ideally suited for the creation of distance line layers. The displacement is shown at the bottom left of the taskbar. To create a shift of a line layer you have to go into editing mode and then select the feature. You can make the Offset Curve tool active and drag the cross to the desired distance. Your changes then can be saved with the Save Layer Edits tool.

Split Features

You can split features using the Split Features icon on the toolbar. Just draw a line across the feature you want to split.

Merge selected features

The Merge Selected Features tool allows to merge features that have common boundaries and the same attributes.

Merge attributes of selected features

The Merge Attributes of Selected Features tool allows to merge attributes of features with common boundaries and attributes without merging their boundaries. You can merge the attributes when selecting several features at once. Then press the Merge Attributes of Selected Features button. Now QGIS offers you which attributes are to be applied to all selected objects. As a result, all objects have the same attribute entries.

Rotate Point Symbols

The Rotate Point Symbols allows to change the rotation of point symbols in the map canvas. You have to define a rotation column from the attribute table of the point layer in the *Advanced* menu of the *Style* menu of the *Layer Properties*. Also you have to go into the 'SVG marker' and choose *Data defined properties ...*. Activate *Angle* and choose 'rotation' as field. Without these settings the tool is inactive.

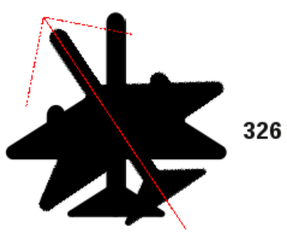


Figure 12.34: Rotate Point Symbols

To change the rotation, select a point feature in the map canvas and rotate it holding the left mouse button pressed. A red arrow with the rotation value will be visualized (see Figure_edit_4). When you release the left mouse button again, the value will be updated in the attribute table.

Note: If you hold the `Ctrl` key pressed, the rotation will be done in 15 degree steps.

12.3.6 Creating new Vector layers

QGIS allows to create new Shapefile layers, new SpatiaLite layers, and new GPX Layers. Creation of a new GRASS layer is supported within the GRASS-plugin. Please refer to section *Creating a new GRASS vector layer* for more information on creating GRASS vector layers.

Creating a new Shapefile layer

To create a new Shape layer for editing, choose *New* → *New Shapefile Layer...* from the *Layer* menu. The *New Vector Layer* dialog will be displayed as shown in Figure_edit_5. Choose the type of layer (point, line or polygon) and the CRS (Coordinate Reference System).

Note that QGIS does not yet support creation of 2.5D features (i.e. features with X,Y,Z coordinates).

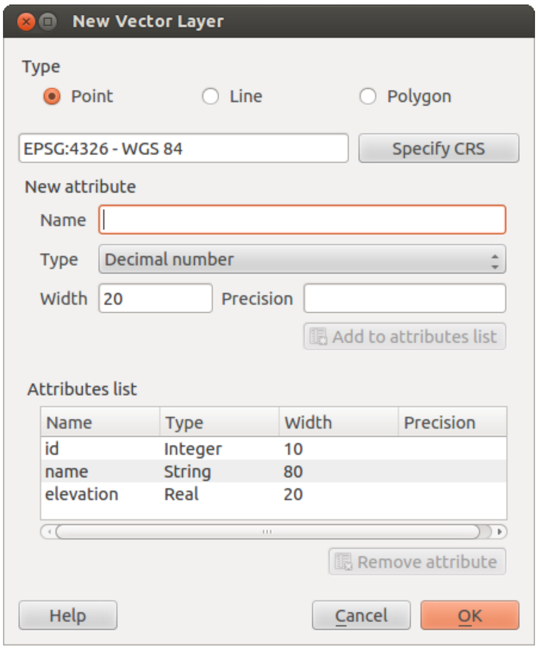


Figure 12.35: Creating a new Shapefile layer Dialog

To complete the creation of the new Shapefile layer, add the desired attributes by clicking on the **[Add to attributes list]** button and specifying a name and type for the attribute. A first 'id' column is added as default but can be removed, if not wanted. Only *Type: real*, *Type: integer*, *Type: string* and *Type:date* attributes are supported. Additionally and according to the attribute type you can also define the width and precision of the new attribute column. Once you are happy with the attributes, click **[OK]** and provide a name for the shapefile. QGIS will automatically add a `.shp` extension to the name you specify. Once the layer has been created, it will be added to the map and you can edit it in the same way as described in Section *Digitizing an existing layer* above.

Creating a new SpatiaLite layer

To create a new SpatiaLite layer for editing, choose *New* → *New SpatiaLite Layer...* from the *Layer* menu. The *New SpatiaLite Layer* dialog will be displayed as shown in Figure_edit_6.

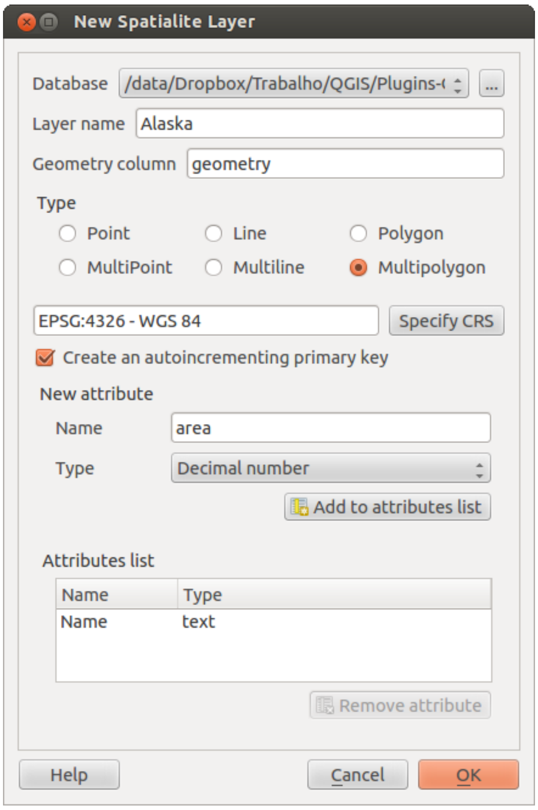


Figure 12.36: Creating a New SpatiaLite layer Dialog

First step is to select an existing SpatiaLite database or to create a new SpatiaLite database. This can be done with the browse button ... to the right of the database field. Then add a name for the new layer and define the layer type and specify the Coordinate Reference System with **[Specify CRS]**. If desired you can select to *Create an autoincrementing primary key*.

To define an attribute table for the new SpatiaLite layer, add the names of the attribute columns you want to create with the according column type and click on the **[Add to attribute list]** button. Once you are happy with the attributes, click **[OK]**. QGIS will automatically add the new layer to the legend and you can edit it in the same way as described in Section *Digitizing an existing layer* above.

Further management of SpatiaLite-Layers can be done with the DB Manager see *DB Manager Plugin*.

Creating a new GPX layer

To create a new GPX file you need to load the GPS plugin first. *Plugins* → *Plugin Manager...* opens the Plugin Manager Dialog. Activate the *GPS Tools* checkbox.

When this plugin is loaded choose *New* → *Create new GPX Layer...* from the *Layer*. In the *Save new GPX file as* dialog you can choose where to save the new GPX layer.

12.3.7 Working with the Attribute Table

The attribute table displays features of a selected layer. Each row in the table represents one map feature and each column contains a particular piece of information about the feature. Features in the table can be searched, selected, moved or even edited.

To open the attribute table for a vector layer, make the layer active by clicking on it in the map legend area. Then from the main menu *Layer* choose *Open Attribute Table*. It is also possible to rightclick on the layer and choose *Open Attribute Table* from the dropdown menu and to click on the *Open Attribute Table* button in the Attributes toolbar.

This will open a new window which displays the feature attributes in the layer (figure_attributes_1). The number of features and the number of selected features are shown in the attribute table title.

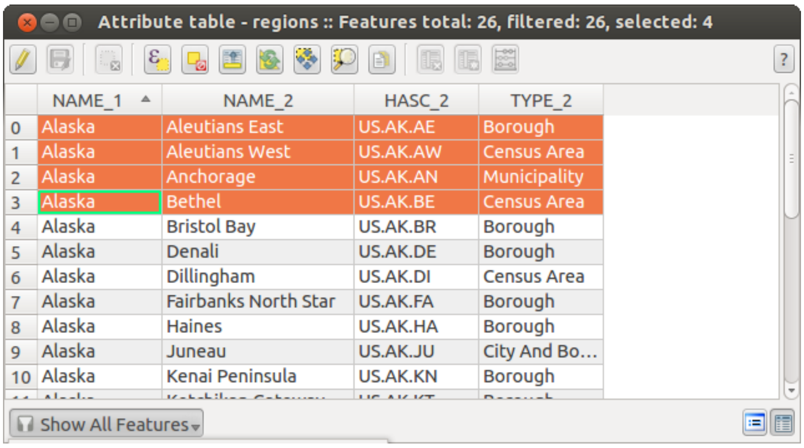


Figure 12.37: Attribute Table for regions layer

Selecting features in an attribute table

Each selected row in the attribute table displays the attributes of a selected feature in the layer. If the set of features selected in the main window is changed, the selection is also updated in the attribute table. Likewise, if the set of rows selected in the attribute table is changed, the set of features selected in the main window will be updated.

Rows can be selected by clicking on the row number on the left side of the row. **Multiple rows** can be marked by holding the `Ctrl` key. A **continuous selection** can be made by holding the `Shift` key and clicking on several row headers on the left side of the rows. All rows between the current cursor position and the clicked row are selected. Moving the cursor position in the attribute table, by clicking a cell in the table, does not change the row selection. Changing the selection in the main canvas does not move the cursor position in the attribute table.

The table can be sorted by any column, by clicking on the column header. A small arrow indicates the sort order (downward pointing means descending values from the top row down, upward pointing means ascending values from the top row down).

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) from which the search should be performed from the dropdown menu and hit the **[Apply]** button. Then only the matching features are shown in the Attribute table.

To make a selection you have to use the Select features using an Expression icon on top of the Attribute table. The Select features using an Expression allows you to define a subset of a table using a *Function List* like in the Field Calculator (see *Field Calculator*). The query result then can be saved as a new vector layer. For example if you want to find regions that are boroughs from the regions.shp of the QGIS sample data you have to open the >*Fields and Values* menu and choose the field that you want to query. Double-klick the field 'TYPE_2' and also **[Load all unique values]** . From list choose and double-klick 'Borough'. In the *Expression* field the following query appears:

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

The matching rows will be selected and the total number of matching rows will appear in the title bar of the attribute table, and in the status bar of the main window. For searches that display only selected features on the map use the Query Builder described in Section *Query Builder*.

To show selected records only, use *Show Selected Features* from the menu at the bottom left.

The other buttons at the top of the attribute table window provide following functionality:

- Toggle editing mode to edit single values and to enable functionalities described below also with `Ctrl+E`
- Save Edits also with `Ctrl+S`
- Unselect all also with `Ctrl+U`
- Move selected to top also with `Ctrl+T`
- Invert selection also with `Ctrl+R`
- Copy selected rows to clipboard also with `Ctrl+C`
- Zoom map to the selected rows also with `Ctrl+J`
- Pan map to the selected rows also with `Ctrl+P`
- Delete selected features also with `Ctrl+D`
- New Column for PostGIS layers and for OGR layers with GDAL version >= 1.6 also with `Ctrl+W`
- Delete Column for PostGIS layers and for OGR layers with GDAL version >= 1.9 also with `Ctrl+L`
- Open field calculator also with `Ctrl+I`

Tip: Skip WKT geometry

If you want to use attribute data in external programs (such as Excel) use the Copy selected rows to clipboard button. You can copy the information without vector geometries if you deactivate *Settings* → *Options* → Data sources menu *Copy geometry in WKT representation from attribute table*.

Save selected features as new layer

The selected features can be saved as any OGR supported vector format and also transformed into another Coordinate Reference System (CRS). Just open the right mouse menu of the layer and click on *Save selection as* → to define the name of the output file, its format and CRS (see Section *Map Legend*). It is also possible to specify OGR creation options within the dialog.

Working with non spatial attribute tables

QGIS allows also to load non spatial tables. This includes currently tables supported by OGR, delimited text as well as the PostgreSQL, MSSQL and Oracle provider. The tables can be used for field lookups or just generally browsed and edited using the table view. When you load the table you will see it in the legend field. It can be opened e.g. with the Open Attribute Table tool and is then editable like any other layer attribute table.

As an example you can use columns of the non spatial table to define attribute values or a range of values that are allowed to be added to a specific vector layer during digitizing. Have a closer look at the edit widget in section *Fields Menu* to find out more.

12.4 Query Builder

The Query Builder allows you to define a subset of a table using a SQL-like WHERE clause and display the result in the main window. The query result then can be saved as a new vector layer.

12.4.1 Query

Open the **Query Builder** by opening the Layer Properties and go to the *General* menu. Under *Feature subset* click on the **[Query Builder]** button to open the *Query builder*. For example, if you have a `regions` layer with a `TYPE_2` field you could select only regions that are `borough` in the *Provider specific filter expression* box of the Query builder. Figure_attributes_2 shows an example of the Query builder populated with the regions.shp layer from the QGIS sample data. The Fields, Values and Operators sections help the user to construct the SQL-like query.

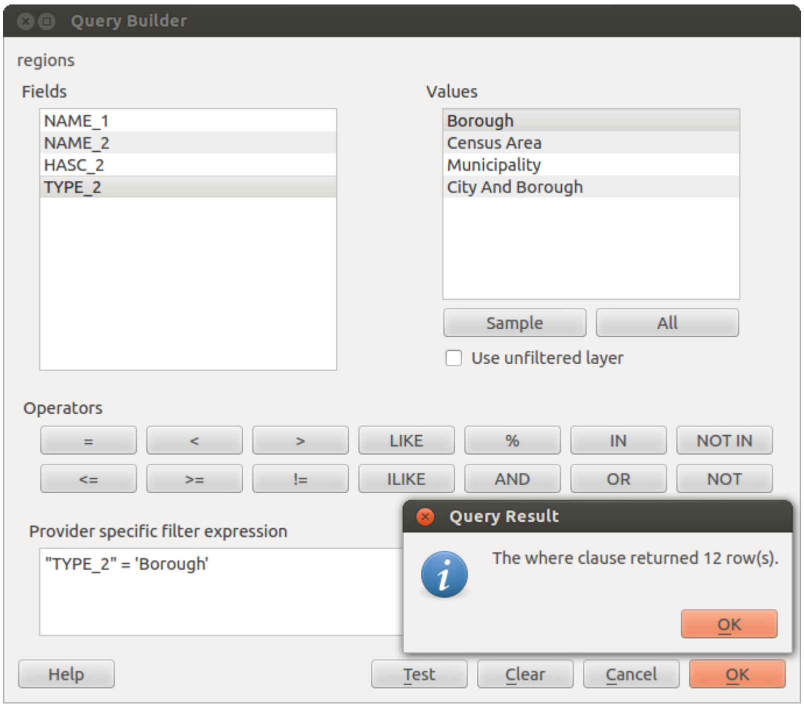


Figure 12.38: Query Builder

The **Fields list** contains all attribute columns of the attribute table to be searched. To add an attribute column to the SQL where clause field, double click its name in the Fields list. Generally you can use the various fields, values and operators to construct the query or you can just type it into the SQL box.

The **Values list** lists the values of an attribute table. To list all possible values of an attribute, select the attribute in the Fields list and click the **[all]** button. To list the first 25 unique values of an attribute column, select the attribute column in the Fields list and click the **[Sample]** button. To add a value to the SQL where clause field, double click its name in the Values list.

The **Operators section** contains all usable operators. To add an operator to the SQL where clause field, click the appropriate button. Relational operators (= , > , ...), string comparison operator (`LIKE`), logical operators (`AND`, `OR`, ...) are available.

The **[Test]** button shows a message box with the number of features satisfying the current query, which is usable in the process of query construction. The **[Clear]** button clears the text in the SQL where clause text field. The **[OK]** button closes the window and selects the features satisfying the query. The **[Cancel]** button closes the window without changing the current selection.

12.4.2 Save selected features as new layer

The selected features can be saved as any OGR supported vector format and also transformed into another Coordinate Reference System (CRS). Just open the right mouse menu of the layer and click on *Save selection as →* to define the name of the output file, its format and CRS (see Section *Map Legend*). It is also possible to specify OGR creation options within the dialog.

12.5 Field Calculator

The Field Calculator button in the attribute table allows to perform calculations on basis of existing attribute values or defined functions, e.g to calculate length or area of geometry features. The results can be written to a new attribute column or it can be used to update values in an already existing column.

You have to bring the vector layer in editing mode, before you can click on the field calculator icon to open the dialog (see figure_attributes_3). In the dialog you first have to select whether you want to only update selected features, create a new attribute field where the results of the calculation will be added or update an existing field.

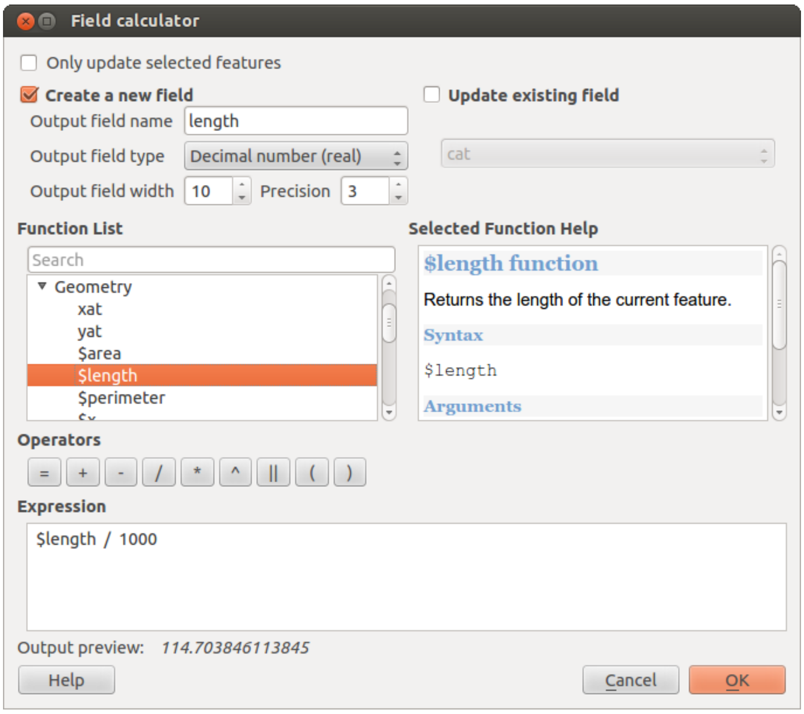


Figure 12.39: Field Calculator

If you choose to add a new field, you need to enter a field name, a field type (integer, real or string), the total field width, and the field precision (see figure_attributes_3). For example, if you choose a field width of 10 and a field precision of 3 it means you have 6 signs before the dot, then the dot and another 3 signs for the precision.

The **Function List** contains functions as well as fields and values. View the help function in the **Selected Function Help**. In **Expression** you see the calculation expressions you create with the **Function List**. The most commonly used operators, see **Operators**.

In the **Function List**, click on *Fields and Values* to view all attributes of the attribute table to be searched. To add an attribute to the Field calculator **Expression** field, double click its name in the *Fields and Values* list. Generally you can use the various fields, values and functions to construct the calculation expression or you can just type it into the box. To display the values of a field, you just right click on the appropriate field. You can choose between *Load top 10 unique values* and *Load all unique values*. On the right side opens the **Field Values** list with the unique values. To add a value to the Field calculator **Expression** box, double click its name in the **Field Values** list.

The *Operators*, *Math*, *Conversions*, *String*, *Geometry* and *Record* groups provides several functions. In *Operators* you find mathematical operators. Find *Math* for mathematical functions. The *Conversions* group contains functions that convert one data type to another. The *String* group provides functions for data strings. In the *Geometry* group you find functions for geometry objects. With *Record* group functions you can add a numeration to your data set. To add a function to the Field calculator **Expression** box, click on the > and then doubleclick the function.

A short example illustrates how the field calculator works. We want to calculate the length in km of the `railroads` layer from the QGIS sample dataset:

1. Load the Shapefile `railroads.shp` in QGIS and press Open Attribute Table.
2. Click on Toggle editing mode and open the Field Calculator dialog.
3. Select the *Create a new field* checkbox to safe the calculations into a new field.
4. Add `length` as Output field name, `real` as Output field type and define Output field width 10 and a Precision of 3.
5. Now doubleclick on function `$length` in the *Geometry* group to add it into the Field calculator expression box.
6. Complete the expression by typing "/ 1000" in the Field calculator expression box and click **[Ok]**.
7. You can now find a new column `length` in the attribute table.

The available functions are listed below.

The field calculator **Function list** with the **Selected Function Help** , **Operators** and **Expression** menu are also available through the rule-based rendering in the Style menu of the Layer properties and the expression based labeling ... in the Labeling core application.

Operators

This group contains operators e.g + - *

```
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 -> 2 fits into 7 three times rest is 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
'string'                     a string value
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
```

Conditionals

This group contains functions to handle conditional checks in expressions.

```
CASE                          evaluates multiple expressions and return a result
CASE ELSE                     evaluates multiple expressions and return 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
```

Mathematical Functions

This group contains math functions e.g square root, sin and cos

```
sqrt(a)                       square root of a
abs                           returns the absolute value of a number.
sin(a)                        sinus of a
cos(a)                        cosinus of a
tan(a)                        tangens of a
asin(a)                       arcussinus of a
acos(a)                       arcuscosinus of a
atan(a)                       arcustangens of a
atan2(y,x)                    arcustangens of y/x using the signs of the two arguments
                              to determine the quadrant of the result
exp                           exponential of an 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                         number 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
```

Conversions

This group contains functions to convert on data type to another e.g string to integer, integer to string.

```
toint                        converts a string to integer number
toreal                       converts a string to real number
tostring                     convert number to string
todatetime                   convert a string into Qt data time type
todate                       convert a string into Qt data type
totime                       convert a string into Qt time type
tointerval                   converts a string to a interval type. Can be used to take days,
                             hours, month, etc off a date
```

Date and Time Functions

This group contains functions for handling date and time data.

```
$now       current date and time
age        difference between two dates
year       extract the year part from a date, or the number of years from a Interval
month      extract the month part from a date, or the number of months from a Interval
week       extract the week number from a date, or the number of weeks from a Interval
day        extract the day from a date, or the number of days from a Interval
hour       extract the hour from a datetime or time, or the number
           of hours from a Interval
minute     extract the minute from a datetime or time, or the number
           of minutes from a Interval
second     extract the second from a datetime or time, or the number
           of minutes from a Interval
```

String Functions

This group contains functions that operate on strings e.g replace, convert to upper case.

```
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 whitespace (spaces, tabs, etc) from a string
length        length of string a
replace       returns a string with the 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
```

Color Functions

This group contains functions for manipulating colors.

```
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
```

Geometry Functions

This group contains functions that operate on geometry objects e.g length, area.

```
xat               retrieves a x coordinate of the current feature
yat               retrieves a y coordinate of the current feature
$area             returns the area size of the current feature
$length           returns the area 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
$geometry         returns the geometry of the current feature. Can be used
                  for processing with other functions.
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 the 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
convexHull        returns the convex hull of a geometry. It 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 a
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
```

Record Functions

This group contains functions that operate on record identifiers.

```
$rownum                      returns the number of the current row
$id                          returns the feature id of the current row
$scale                       returns the current scale of the map canvas
```

Fields and Values

Contains a list of fields from the layer. Sample values can also be accessed via right-click.

Select the field name from the list then right-click to access context menu with options to load sample values from the selected field.

CHAPTER 13

Working with Raster Data

13.1 Working with Raster Data

This Section describes how to visualize and set raster layer properties. QGIS uses the GDAL library to read and write raster data formats, including Arc/Info Binary Grid, Arc/Info ASCII Grid, GeoTIFF, Erdas Imagine and many more. GRASS raster support is supplied by a native QGIS data provider plugin. The raster data can also be loaded in read mode from zip and gzip archives into QGIS.

At the date of this document, more than 100 raster formats are supported by the GDAL library (see GDAL-SOFTWARE-SUITE *Literature and Web References*). A complete list is available at http://www.gdal.org/formats_list.html.

Note: Not all of the listed formats may work in QGIS for various reasons. For example, some require external commercial libraries or the GDAL installation of your OS was not built to support the format you want to use. Only those formats that have been well tested will appear in the list of file types when loading a raster into QGIS. Other untested formats can be loaded by selecting the `[GDAL] All files (*)` filter.

Working with GRASS raster data is described in Section *GRASS GIS Integration*.

13.1.1 What is raster data?

Raster data in GIS are matrices of discrete cells that represent features on, above or below the earth's surface. Each cell in the raster grid is the same size, and cells are usually rectangular (in QGIS they will always be rectangular). Typical raster datasets include remote sensing data such as aerial photography or satellite imagery and modelled data such as an elevation matrix.

Unlike vector data, raster data typically do not have an associated database record for each cell. They are geocoded by its pixel resolution and the x/y coordinate of a corner pixel of the raster layer. This allows QGIS to position the data correctly in the map canvas.

QGIS makes use of georeference information inside the raster layer (e.g. GeoTiff) or in an appropriate world file to properly display the data.

13.1.2 Loading raster data in QGIS

Raster layers are loaded either by clicking on the Add Raster Layer icon or by selecting the *Layer* → *Add Raster Layer* menu option. More than one layer can be loaded at the same time by holding down the `Control` or `Shift` key and clicking on multiple items in the dialog *Open a GDAL Supported Raster Data Source*.

Once a raster layer is loaded in the map legend you can click on the layer name with the right mouse button to select and activate layer specific features or to open a dialog to set raster properties for the layer.

Right mouse button menu for raster layers

- *Zoom to layer extent*
- *Zoom to Best Scale (100%)*
- *Show in Overview*
- *Remove*
- *Duplicate*
- *Set Layer CRS*
- *Set Project CRS from Layer*
- *Save as ...*
- *Properties*
- *Rename*
- *Copy Style*
- *Add New Group*
- *Expand all*
- *Collapse all*
- *Update Drawing Order*

13.2 Raster Properties Dialog

To view and set the properties for a raster layer, double click on the layer name in the map legend or right click on the layer name and choose *Properties* from the context menu:

This will open the *Raster Layer Properties* dialog, (see figure_raster_1).

There are several menus in the dialog:

- *General*
- *Style*
- *Transparency*
- *Pyramids*
- *Histogram*
- *Metadata*

13.2.1 General Menu

Layer Info

The *General* menu displays basic information about the selected raster, including the layer source path, the display name in the legend (which can be modified) and the number of columns, rows and No-Data Values of the raster.

Coordinate reference system

Below you find the coordinate reference system (CRS) information printed as a PROJ.4-string. If this setting is not correct, it can be modified by clicking the **[Specify]** button.

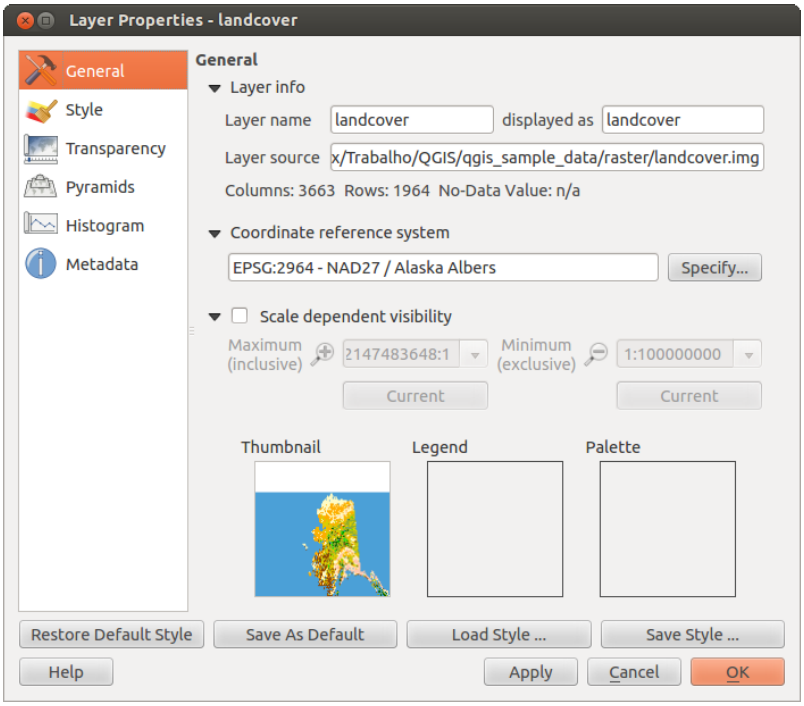


Figure 13.1: Raster Layers Properties Dialog

Scale Dependent visibility

Additionally Scale Dependent visibility can be set in this tab. You need to check the checkbox and set an appropriate scale where your data will be displayed in the map canvas.

At the bottom you can see a thumbnail of the layer, its legend symbol, and the palette.

13.2.2 Style Menu

Band rendering

QGIS offers four different *Render types*. The renderer chosen is dependent on the data type.

1. Multiband color - if the file comes as a multi band with several bands (e.g. used with a satellite image with several bands)
2. Paletted - if a single band file comes with an indexed palette (e.g. used with a digital topographic map)
3. Singleband gray- (one band of) the image will be rendered as gray, QGIS will choose this renderer if the file neither has multi bands, nor has an indexed palette nor has a continous palette (e.g. used with a shaded relief map)
4. Singleband pseudocolor - this renderer is possible for files with a continuous palette, e.g. the file has got a color map (e.g. used with an elevation map)

Multiband color

With the multiband color renderer three selected bands from the image will be rendered, each band representing the red, green or blue component that will be used to create a color image. You can choose several *Contrast enhancement* methods: 'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' and 'Clip to min max'.

This selection offers you a wide range of options to modify the appearance of your rasterlayer. First of all you have to get the data range from your image. This can be done by choosing the *Extent* and pressing **[Load]**. QGIS can *Estimate (faster)* the *Min* and *Max* values of the bands or use the *Actual (slower) Accuracy*.

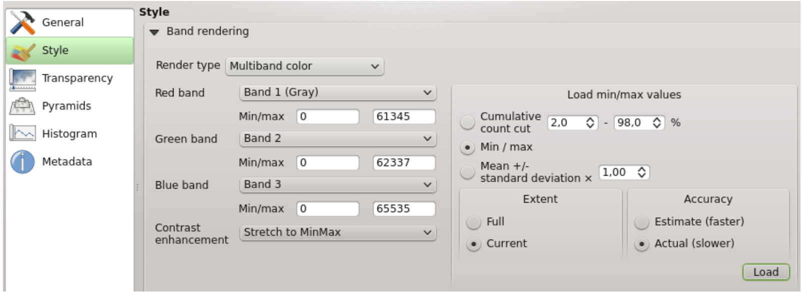


Figure 13.2: Raster Renderer - Multiband color

Now you can scale the colors with the help of the *Load min/max values* section. A lot of images have few very low and high data. These outliers can be eliminated using the *Cumulative count cut* setting. The standard data range is set from 2% until 98% of the data values and can be adapted manually. With this setting the gray character of the image can disappear. With the scaling option *Min/max* QGIS creates a color table with the whole data included in the original image. E.g. QGIS creates a color table with 256 values, given the fact that you have 8bit bands. You can also calculate your color table using the *Mean +/- standard deviation x* 1,00. Then only the values within the standard deviation or within multiple standard deviations are considered for the color table. This is useful when you have one or two cells with abnormally high values in a raster grid that are having a negative impact on the rendering of the raster.

All calculation can also be made for the *Current* extend.

Tip: Viewing a Single Band of a Multiband Raster

If you want to view a single band (for example Red) of a multiband image, you might think you would set the Green and Blue bands to "Not Set". But this is not the correct way. To display the Red band, set the image type to 'Singleband gray', then select Red as the band to use for Gray.

Paletted

This is the standard render option for singleband files that already include a color table, where each pixel value is assigned to a certain color. In that case, the palette is rendered automatically. If you want to change colors assigned to certain values, just double-click on the color and the *Select color* dialog appears.

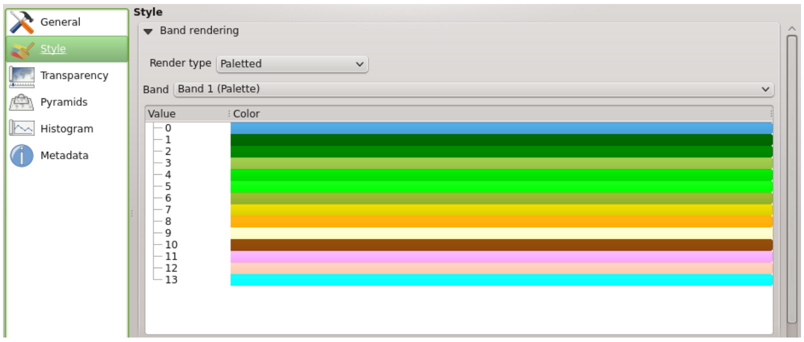


Figure 13.3: Raster Renderer - Paletted

Contrast enhancement

Note: When adding GRASS rasters the option *Contrast enhancement* will be always set to automatically to *stretch to min max* regardless if the QGIS general options this is set to another value.

Singleband gray

This renderer allows you to render a single band layer with a *Color gradient* 'Black to white' or 'White to black'. You can define a *Min* and a *Max* value with choosing the *Extend* first and then pressing **[Load]**. QGIS can

Estimate (faster) the *Min* and *Max* values of the bands or use the *Actual (slower) Accuracy*.

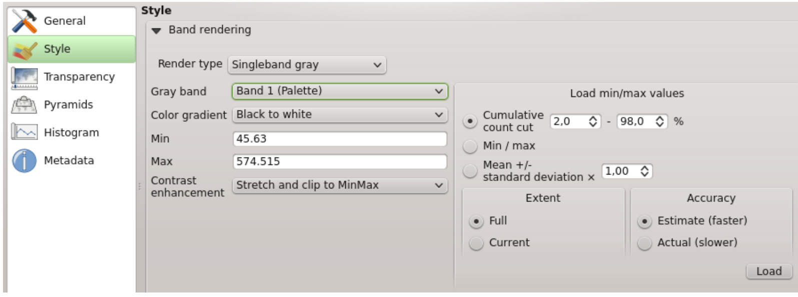


Figure 13.4: Raster Renderer - Singleband gray

With the *Load min/max values* section scaling of the color table is possible. Outliers can be eliminated using the *Cumulative count cut* setting. The standard data range is set from 2% until 98% of the data values and can be adapted manually. With this setting the gray character of the image can disappear. Further settings can be made with *Min/max* and *Mean +/- standard deviation x* 1,00. While the first one creates a color table with the whole data included in the original image the second creates a colortable that only considers values within the standard deviation or within multiple standard deviations. This is useful when you have one or two cells with abnormally high values in a raster grid that are having a negative impact on the rendering of the raster.

Singleband pseudocolor

This is a render option for single band files including a continous palette. You can also create individual color maps for the single bands here. Three types of color interpolation are available:

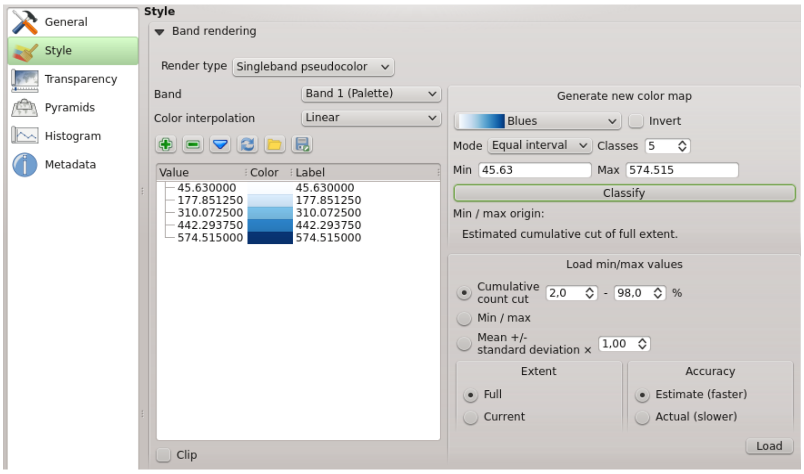


Figure 13.5: Raster Renderer - Singleband pseudocolor

1. Discrete
2. Linear
3. Exact

In the left block the button Add values manually adds a value to the individual color table. Button Remove selected row deletes a value from the individual color table and the Sort colormap items button sorts the color table according to the pixel values in the value column. Double clicking on the value-column lets you insert a specific value. Double clicking on the color-column opens the dialog *Change color* where you can select a color to apply on that value. Further you can also add labels for each color but this value won't be displayed when you use the identify feature tool. You can also click on the button Load color map from band, which tries to load the table from the band (if it has any). And you can use the buttons Load color map from file or Export color map to file to load an existing color table or to save the defined color table for other sessions.

In the right block *Generate new color map* allows you to create newly categorized colormaps. For the *Classification mode* 'Equal interval' you only need to select the *number of classes* 1,00 and press the button *Classify*. You can invert the colors of the the color map by clicking the *Invert* checkbox. In case of the *Mode* 'Continous' QGIS creates classes depending on the *Min* and *Max* automatically. Defining *Min/Max* values can be done with the help of *Load min/max values* section. A lot of images have few very low and high data. These outliers can be eliminated using the *Cumulative count cut* setting. The standard data range is set from 2% until 98% of the data values and can be adapted manually. With this setting the gray character of the image can disappear. With the scaling option *Min/max* QGIS creates a color table with the whole data included in the original image. E.g. QGIS creates a color table with 256 values, given the fact that you have 8bit bands. You can also calculate your color table using the *Mean +/- standard deviation x* 1,00 . Then only the values within the standard deviation or within multiple standard deviations are considered for the color table.

Color rendering

For every *Band rendering* a *Color rendering* is possible.

You can achieve special rendering effects for your raster file(s) using one one of the blending modes (see *blend_modes*).

Further settings can be made in modifiying the *Brightness*, the *Saturation* and the *Contrast*. You can use a *Grayscale* option where you can choose between 'By lightness', 'By luminosity' and 'By average'. For one hue in the color table you can modiy the 'Strength'.

Resampling

The *Resampling* option makes it appearance when you zoom in and out of the image. Resampling modes can optimize the appearance of the map. They calculate a new gray value matrix through a geometric transformation.

While applying the 'Nearest neighbour' method the map can have a pixelated structure when zooming in. This appearance can be improved by using the 'Bilinear' or 'Cubic' method. Sharp features are caused to be blurred now. The effect is a smoother image. The method can be applied e.g. to digital topographic raster maps.

13.2.3 Transparency Menu

QGIS has the ability to display each raster layer at varying transparency levels. Use the transparency slider to indicate to what extent the underlying layers (if any) should be visible though the current raster layer. This is very useful, if you like to overlay more than one rasterlayer, e.g. a shaded relief map overlayed by a classified rastermap. This will make the look of the map more three dimensional.

Additionally you can enter a rastervalue, which should be treated as *NODATA* in the *Additional no data value* menu.

An even more flexible way to customize the transparency can be done in the *Custom transparency options* section. The transparency of every pixel can be set here.

As an example we want to set the water of our example raster file `landcover.tif` to a transparency of 20 %. The following steps are neccessary:

1. Load the rasterfile `landcover`.
2. Open the *Properties* dialog by double-clicking on the raster name in the legend or by right-clicking and choosing *Properties* from the popup menu.
3. Select the *Transparency* menu
4. From the *Transparency band* menu choose 'None'.

5. Click the Add values manually button. A new row will appear in the pixel-list.
6. Enter the raster-value (we use 0 here) in the 'From' and 'To' column and adjust the transparency to 20 %.
7. Press the **[Apply]** button and have a look at the map.

You can repeat the steps 5 and 6 to adjust more values with custom transparency.

As you can see this is quite easy to set custom transparency, but it can be quite a lot of work. Therefore you can use the button Export to file to save your transparency list to a file. The button Import from file loads your transparency settings and applies them to the current raster layer.

13.2.4 Pyramids Menu

Large resolution raster layers can slow navigation in QGIS. By creating lower resolution copies of the data (pyramids), performance can be considerably improved as QGIS selects the most suitable resolution to use depending on the level of zoom.

You must have write access in the directory where the original data is stored to build pyramids.

Several resampling methods can be used to calculate the pyramids:

- Nearest Neighbour
- Average
- Gauss
- Cubic
- Mode
- None

If you choose 'Internal (if possible)' from the *Overview format* menu QGIS tries to build pyramids internally. You can also choose 'External' and 'External (Erdas Imagine)'.

Please note that building pyramids may alter the original data file and once created they cannot be removed. If you wish to preserve a 'non-pyramided' version of your raster, make a backup copy prior to building pyramids.

13.2.5 Histogram Menu

The *Histogram* menu allows you to view the distribution of the bands or colors in your raster. It is generated automatically when you open the *Histogram* menu. All existing bands will be displayed together. You can save the histogram as an image with the button. With the *Visibility* option in the *Prefs/Actions* menu you can display histograms of the individual bands. You will need to select the option *Show selected band*. The *Min/max options* allow you to 'Always show min/max markers', to 'Zoom to min/max' and to 'Update style to min/max'. With the *Actions* option you can 'Reset' and 'Recompute histogram' after you have chosen the *Min/max options*.

13.2.6 Metadata Menu

The *Metadata* menu displays a wealth of information about the raster layer, including statistics about each band in the current raster layer. From this menu entries are made for the *Description*, *Attribution*, *MetadataUrl* and *Properties*. In *Properties* statistics are gathered on a 'need to know' basis, so it may well be that a given layers statistics have not yet been collected.

13.3 Raster Calculator

The *Raster Calculator* in the *Raster* menu (see figure_raster_2) allows to perform calculations on basis of existing raster pixel values. The results are written to a new raster layer with a GDAL supported format.

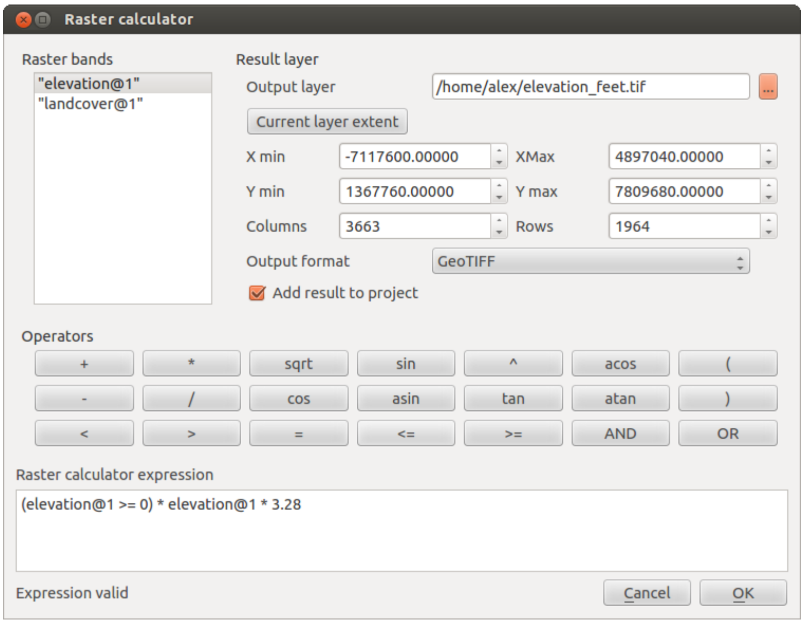


Figure 13.6: Raster Calculator

The **Raster bands** list contains all loaded raster layers that can be used. To add a raster to the raster calculator expression field, double click its name in the Fields list. You can then use the operators to construct calculation expressions or you can just type it into the box.

In the **Result layer** section you have to define an output layer. You can then define the extent of the calculation area based on an input raster layer or based on X,Y coordinates and on Columns and Rows to set the resolution of the output layer. If the input layer has a different resolution, the values will be resampled with nearest neighbor algorithm.

The **Operators** section contains all usable operators. To add an operator to the raster calculator expression box, click the appropriate button. Mathematical calculations (`+`, `-`, `*`, ...) and trigonometric functions (`sin`, `cos`, `tan`, ...) are available. Stay tuned for more operators to come!

With the *Add result to project* checkbox the result layer will automatically added to the legend area and can be visualized.

13.3.1 Examples

Convert elevation values from meter to feet

Creating an elevation raster feet from a raster in meter, you need to use the conversion factor for meters to feet: 3.28. The expression is:

```
elevation@1 * 3.28
```

Using a mask

If you want to mask out parts of a raster, because you are only interested in elevations above 0 meter, you can use following expression to create a mask and apply the result to a raster in one step.

```
(elevation@1 >= 0) * elevation@1
```

For every cell greater than or equal to 0, set its value to 1, otherwise set it to 0. This creates the mask on the fly.

CHAPTER 14

Working with OGC Data

14.1 QGIS as OGC Data Client

The Open Geospatial Consortium (OGC), is an international organization with more than 300 commercial, governmental, nonprofit and research organizations worldwide. Its members develop and implement standards for geospatial content and services, GIS data processing and exchange.

Describing a basic data model for geographic features an increasing number of specifications are developed to serve specific needs for interoperable location and geospatial technology, including GIS. Further information can be found under http://www.opengeospatial.org/.

Important OGC specifications supported by QGIS are:

- **WMS** — Web Map Service (*WMS/WMTS Client*)
- **WMTS** — Web Map Tile Service (*WMS/WMTS Client*)
- **WFS** — Web Feature Service (*WFS and WFS-T Client*)
- **WFS-T** — Web Feature Service - Transactional (*WFS and WFS-T Client*)
- **WCS** — Web Coverage Service (*WCS Client*)
- **SFS** — Simple Features for SQL (*PostGIS Layers*)
- **GML** — Geography Markup Language

OGC services are increasingly being used to exchange geospatial data between different GIS implementations and data stores. QGIS can deal with the above specifications as a client, being **SFS** (through support of the PostgreSQL / PostGIS data provider, see Section *PostGIS Layers*).

14.1.1 WMS/WMTS Client

Overview of WMS Support

QGIS currently can act as a WMS client that understands WMS 1.1, 1.1.1 and 1.3 servers. It has particularly been tested against publicly accessible servers such as DEMIS.

WMS servers act upon requests by the client (e.g. QGIS) for a raster map with a given extent, set of layers, symbolization style, and transparency. The WMS server then consults its local data sources, rasterizes the map, and sends it back to the client in a raster format. For QGIS this would typically be JPEG or PNG.

WMS is generically a REST (Representational State Transfer) service rather than a fully-blown Web Service. As such, you can actually take the URLs generated by QGIS and use them in a web browser to retrieve the same images that QGIS uses internally. This can be useful for troubleshooting, as there are several brands of WMS servers in the market and they all have their own interpretation of the WMS standard.

WMS layers can be added quite simply, as long as you know the URL to access the WMS server, you have a serviceable connection to that server, and the server understands HTTP as the data transport mechanism.

Overview of WMTS Support

QGIS can also act as a WMTS client. WMTS is an OGC standard for distributing tile sets of geospatial data. This is a faster and a more efficient way of distributing data than WMS because with WMTS the tile sets are pre-generated and the client only requests the transmission of the tiles and not their production. A WMS request typically involves both the generation and transmission of the data. A well known example of a non-OGC standard for viewing tiled geospatial data is Google Maps.

In order to display the data at a variety of scales close to what the user might want, the WMTS tile sets are produced at several different scale levels and are made available for the GIS client to request them.

This diagram illustrates the concept of tile sets:

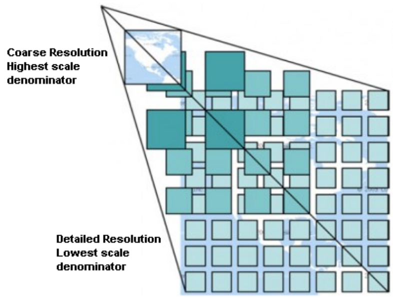


Figure 14.1: Concept of WMTS tile sets

The two types of WMTS interfaces that QGIS supports are via Key-Value-Pairs (KVP) and RESTful. These two interfaces are different and you need to specify them to QGIS differently.

1) In order to access a **WMTS KVP** service, a QGIS user opens the WMS/WMTS interface and adds the following string to the URL of the WMTS tile service:

`"?SERVICE=WMTS&REQUEST=GetCapabilities"`

An example of this type of address is

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?\
  service=WMTS&request=GetCapabilities
```

For testing the topo2 layer in this WMTS works nicely. Adding this string indicates that a WMTS web service is to be used instead of a WMS service

2) The **RESTful WMTS** service takes a different form, it is a straightforward URL, the format recommended by the OGC is:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```

This format helps you to recognize that it is a RESTful address. A RESTful WMTS is accessed in QGIS by simply adding its address in the WMS setup in the URL field of the form. An example for an Austrian basemap of this type of address is http://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml

Note: You can find some old service call WMS-C. Thoses services are quiet similar to WMTS service same purpose but working a little bit differently). You can manage them as the same way you do it for WMTS services. Just add ?tiled=true at the end of the url. See http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification for more information about this specification.

When you read WMTS you can often think WMS-C also.

Selecting WMS/WMTS Servers

The first time you use the WMS feature, there are no servers defined.

Begin by clicking the Add WMS layer button inside the toolbar, or through the *Layer* → *Add WMS Layer..* menu.

The dialog *Add Layer(s) from a Server* for adding layers from the WMS server appears. You can add some servers to play with by clicking the **[Add default servers]** button. This will add two WMS demo servers for you to use, the WMS servers of the DM Solutions Group and Lizardtech. To define a new WMS server in the tab *Layers*, select the **[New]** button. Then enter the parameters to connect to your desired WMS server, as listed in table_OGC_1:

Name	A name for this connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WMS Servers.
URL	URL of the server providing the data. This must be a resolvable host name; the same format as you would use to open a telnet connection or ping a host.
Username	Username to access a secured WMS-server. This parameter is optional.
Password	Password for a basic authenticated WMS-server. This parameter is optional.
Ignore GetMap URI	*Ignore GetMap URI reported in capabilities*, use given URI from URL-field above.
Ignore GetFeatureInfo URI	*Ignore GetFeatureInfo URI reported in capabilities* , use given URI from URL-field above

Table OGC 1: WMS Connection Parameters

If you need to set up a proxy-server to be able to receive WMS-services from the internet, you can add your proxy-server in the options. Choose menu *Settings* → *Options* and click on the tab *Network & Proxy*. There you can add your proxy-settings and enable them by setting the *Use proxy for web access*. Make sure that you select the correct proxy type from the *Proxy type* dropdown menu.

Once the new WMS Server connection has been created, it will be preserved for future QGIS sessions.

Tip: On WMS Server URLs

Be sure, when entering in the WMS server URL, that you have the base URL. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.

Loading WMS/WMTS Layers

Once you have successfully filled in your parameters you can use the **[Connect]** button to retrieve the capabilities of the selected server. This includes the Image encoding, Layers, Layer Styles and Projections. Since this is a network operation, the speed of the response depends on the quality of your network connection to the WMS server. While downloading data from the WMS server, the download progress is visualized in the left bottom of the WMS dialog.

Your screen should now look a bit like figure_OGR_1, which shows the response provided by the DM Solutions Group WMS server.

Image Encoding

The *Image encoding* section now lists the formats that are supported by both the client and server. Choose one depending on your image accuracy requirements.

Tip: Image Encoding

You will typically find that a WMS server offers you the choice of JPEG or PNG image encoding. JPEG is a lossy compression format, whereas PNG faithfully reproduces the raw raster data.

Use JPEG if you expect the WMS data to be photographic in nature and/or you don't mind some loss in picture quality. This trade-off typically reduces by 5 times the data transfer requirement compared to PNG.

Use PNG if you want precise representations of the original data, and you don't mind the increased data transfer requirements.

Options

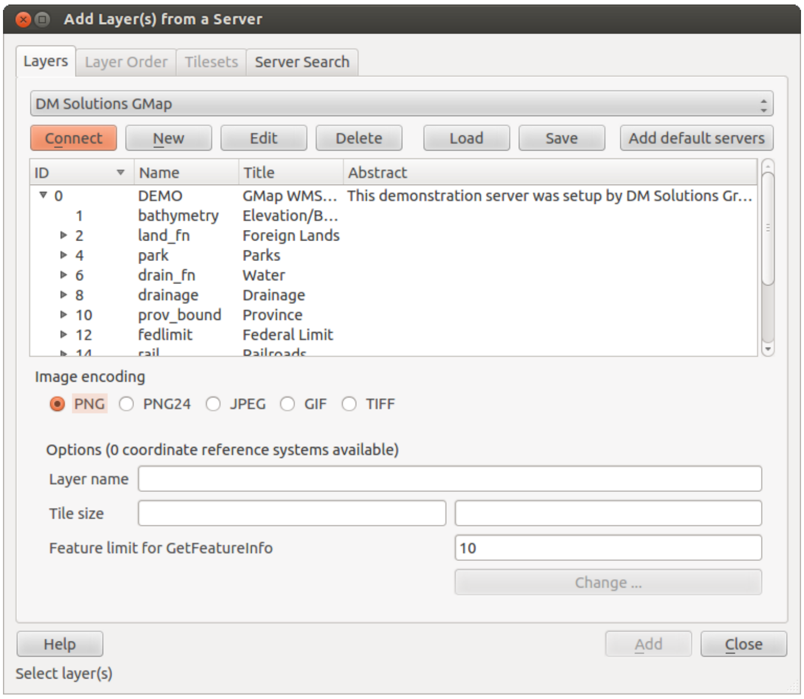


Figure 14.2: Dialog for adding a WMS server, showing its available layers

The Options field provides a text field where you can add a *Layer name* for the WMS-layer. This name will be presented in the legend after loading the layer.

Below the layer name you can define *Tile size*, if you want to set tile sizes (eg. 256x256) to split up the WMS request into multiple requests.

The *Feature limit for GetFeatureInfo* defines what features from the server to query.

If you select a WMS from the list a field with the default projection, provided by the mapserver, appears. If the **[Change...]** button is active, you can click on it and change the default projection of the WMS to another CRS, provided by the WMS server.

Layer Order

The tab *Layer Order* lists the selected layers available from the current connected WMS server. You may notice that some layers are expandable, this means that the layer can be displayed in a choice of image styles.

You can select several layers at once, but only one image style per layer. When several layers are selected, they will be combined at the WMS Server and transmitted to QGIS in one go.

Tip: WMS Layer Ordering

WMS layers rendered by a server are overlaid in the order listed in the Layers section, from top to bottom of the list. If you want to change the overlay order, you can use the tab *Layer Order*.

Transparency

In this version of QGIS, the *Global transparency* setting from the *Layer Properties* is hard-coded to be always on, where available.

Tip: WMS Layer Transparency

The availability of WMS image transparency depends on the image encoding used: PNG and GIF support transparency, whilst JPEG leaves it unsupported.

Coordinate Reference System

A Coordinate Reference System (CRS) is the OGC terminology for a QGIS Projection.

Each WMS Layer can be presented in multiple CRSs, depending on the capability of the WMS server.

To choose a CRS, select **[Change...]** and a dialog similar to Figure Projection 3 in *Working with Projections* will appear. The main difference with the WMS version of the screen is that only those CRSs supported by the WMS Server will be shown.

Server search

Within QGIS you can search for WMS-servers. Figure_OGC_2 shows the tab *Server Search* with the *Add Layer(s) from a Server* dialog.

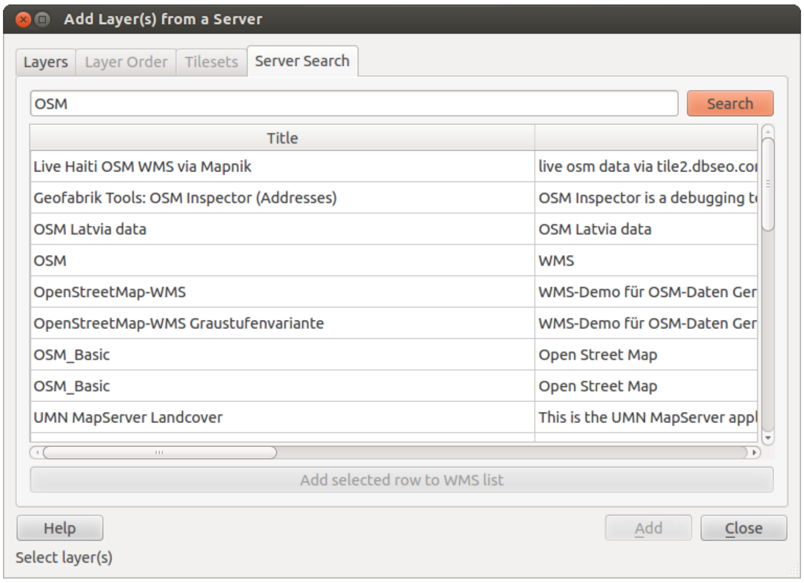


Figure 14.3: Dialog for searching WMS servers after some keywords

As you can see it is possible to enter a search-string in the text field and hit the **[Search]** button. After a short while the search result will be populated into the list below the text field. Browse the result list and inspect your search results within the table. To visualize the results, select a table entry, press the **[Add selected row to WMS-list]** button and change back to the tab *Layers*. QGIS automatically has updated your server list and the selected search result is already enabled in the list of saved WMS-servers in the *Layers* tab. You only need to request the list of layers by clicking the **[Connect]** button. This option is quite handy when you want to search maps by specific keywords.

Basically this option is a frontend to the API of http://geopole.org.

Tilesets

When using WMTS (Cached WMS) Services like

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?\
  service=WMTS&request=GetCapabilities
```

you are able to browse through the tab *Tilesets* given by the server. Additional information like tile size, formats and supported CRS are listed in this table. In combination with this feature you can use the tile scale slider from the *Settings* → *Panels* (KDE and Windows) or *View* → *Panels* (Gnome and MacOSX) then choose *Tile scale*, which gives you the available scales from the tileserver with nice slider docked in.

Using the Identify Tool

Once you have added a WMS server, and if any layer from a WMS server is queryable, you can then use the Identify tool to select a pixel on the map canvas. A query is made to the WMS server for each selection made. The results of the query are returned in plain text. The formatting of this text is dependent on the particular WMS server used. **Format selection**

If multiple output formats are supported by the server, a combo box with supported formats is automatically added to the identify results dialog and the selected format will is stored in project for the layer. **GML format support**

The Identify tool supports WMS server response (GetFeatureInfo) in GML (it is called Feature in QGIS GUI in this context) format. If "Feature" format is supported by the server and selected, results of the Identify tool are vector features like from regular vector layer. When a single feature is selected in the tree, it is highlighted in the map and it can be copied to clipboard and pasted to another vector layer. See example setup of UMN Mapserver below to support GetFeatureInfo GML format.

```
# 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"
```

Viewing Properties

Once you have added a WMS server, you can view its properties by right-clicking on it in the legend, and selecting *Properties*. **Metadata Tab**

The tab *Metadata* displays a wealth of information about the WMS server, generally collected from the Capabilities statement returned from that server. Many definitions can be cleaned by reading the WMS standards (see OPEN-GEOSPATIAL-CONSORTIUM *Literature and Web References*), but here are a few handy definitions:

- **Server Properties**
  - **WMS Version** — The WMS version supported by the server.
  - **Image Formats** — The list of MIME-types the server can respond with when drawing the map. QGIS supports whatever formats the underlying Qt libraries were built with, which is typically at least `image/png` and `image/jpeg`.
  - **Identity Formats** — The list of MIME-types the server can respond with when you use the Identify tool. Currently QGIS supports the `text-plain` type.
- **Layer Properties**
  - **Selected** — Whether or not this layer was selected when its server was added to this project.

- **Visible** — Whether or not this layer is selected as visible in the legend. (Not yet used in this version of QGIS.)
- **Can Identify** — Whether or not this layer will return any results when the Identify tool is used on it.
- **Can be Transparent** — Whether or not this layer can be rendered with transparency. This version of QGIS will always use transparency if this is `Yes` and the image encoding supports transparency
- **Can Zoom In** — Whether or not this layer can be zoomed in by the server. This version of QGIS assumes all WMS layers have this set to `Yes`. Deficient layers may be rendered strangely.
- **Cascade Count** — WMS servers can act as a proxy to other WMS servers to get the raster data for a layer. This entry shows how many times the request for this layer is forwarded to peer WMS servers for a result.
- **Fixed Width, Fixed Height** — Whether or not this layer has fixed source pixel dimensions. This version of QGIS assumes all WMS layers have this set to nothing. Deficient layers may be rendered strangely.
- **WGS 84 Bounding Box** — The bounding box of the layer, in WGS 84 coordinates. Some WMS servers do not set this correctly (e.g. UTM coordinates are used instead). If this is the case, then the initial view of this layer may be rendered with a very 'zoomed-out' appearance by QGIS. The WMS webmaster should be informed of this error, which they may know as the WMS XML elements `LatLonBoundingBox`, `EX_GeographicBoundingBox` or the CRS:84 `BoundingBox`.
- **Available in CRS** — The projections that this layer can be rendered in by the WMS server. These are listed in the WMS-native format.
- **Available in style** — The image styles that this layer can be rendered in by the WMS server.

WMS Client Limitations

Not all possible WMS Client functionality had been included in this version of QGIS. Some of the more notable exceptions follow.

Editing WMS Layer Settings

Once you've completed the Add WMS layer procedure, there is no ability to change the settings. A workaround is to delete the layer completely and start again.

WMS Servers Requiring Authentication

Currently public accessible and secured WMS-services are supported. The secured WMS-servers can be accessed by public authentication. You can add the (optional) credentials when you add a WMS-server. See section *Selecting WMS/WMTS Servers* for details.

Tip: Accessing secured OGC-layers

If you need to access secured layers with other secured methods than basic authentication, you could use InteProxy as a transparent proxy, which does support several authentication methods. More information can be found at the InteProxy manual found on the website http://inteproxy.wald.intevation.org.

Tip: |qg| WMS Mapserver

From Version 1.7.0 QGIS has its own implementation of a WMS 1.3.0 Mapserver. Read more about this at chapter *QGIS as OGC Data Server*.

14.1.2 WCS Client

A Web Coverage Service (WCS) provides access to raster data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The WCS may be compared to the WFS and the WMS.

As WMS and WFS service instances, a WCS allows clients to choose portions of a server's information holdings based on spatial constraints and other query criteria.

QGIS has a native WCS provider and supports both version 1.0 and 1.1 (which are significantly different), but currently it prefers 1.0, because 1.1 has many issues, each server implements it in different way with various particularities.

The native WCS provider handles all network requests and uses all standard QGIS network settings (especially proxy). It is also possible select cache mode (always cache, prefer cache, prefer network, always network) and the provider also supports selection of time position if temporal domain is offered by server.

14.1.3 WFS and WFS-T Client

In QGIS, a WFS layer behaves pretty much like any other vector layer. You can identify and select features and view the attribute table. Since QGIS 1.6 editing (WFS-T) is also supported.

In general adding a WFS layer is very similar to the procedure used with WMS. The difference is there are no default servers defined, so we have to add our own.

Loading a WFS Layer

As an example we use the DM Solutions WFS server and display a layer. The URL is: http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap

1. Click on the Add WFS Layer tool on the Layers toolbar, the dialog *Add WFS Layer from a Server* appears
2. Click on **[New]**
3. Enter 'DM Solutions' as name
4. Enter the URL (see above)
5. Click **[OK]**
6. Choose 'DM Solutions' from the dropdown list *Server Connections*
7. Click **[Connect]**
8. Wait for the list of layers to be populated
9. Select the *Parks* layer in the list
10. Click **[Apply]** to add the layer to the map

Note that proxy settings you have set in your preferences are also recognized.

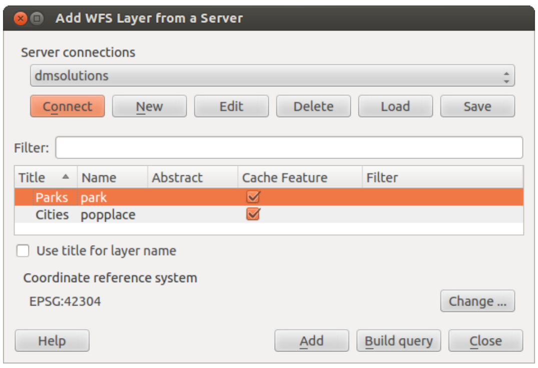


Figure 14.4: Adding a WFS layer

You'll notice the download progress is visualized in the left bottom of the QGIS main window. Once the layer is loaded, you can identify and select a province or two and view the attribute table.

Only WFS 1.0.0 is supported. At this point there have not been many tests against WFS versions implemented in other WFS-servers. If you encounter problems with any other WFS-server, please do not hesitate to contact the development team. Please refer to Section *Help and Support* for further information about the mailinglists.

Tip: Finding WFS Servers

You can find additional WFS servers by using Google or your favorite search engine. There are a number of lists with public URLs, some of them maintained and some not.

14.2 QGIS as OGC Data Server

QGIS Server is an open source WMS 1.3 and WFS 1.0.0 implementation which, in addition, implements advanced cartographic features for thematic mapping. The QGIS Server is a FastCGI/CGI (Common Gateway Interface) application written in C++ that works together with a webserver (e.g. Apache, Lighttpd). It is funded by the EU projects Orchestra, Sany and the city of Uster in Switzerland.

It uses QGIS as backend for the GIS logic and for map rendering. Furthermore the Qt library is used for graphics and for platform independent C++ programming. In contrast to other WMS software, the QGIS Server uses cartographic rules as a configuration language, both for the server configuration and for the user-defined cartographic rules.

Moreover, the QGIS Server project provides the 'Publish to Web' plugin, a plugin for QGIS desktop which exports the current layers and symbology as a web project for QGIS Server (containing cartographic visualization rules expressed in SLD).

As QGIS desktop and QGIS Server use the same visualization libraries, the maps that are published on the web look the same as in desktop GIS. The 'Publish to Web' plugin currently supports basic symbolization, with more complex cartographic visualization rules introduced manually. As the configuration is performed with the SLD standard and its documented extensions, there is only one standardised language to learn, which greatly simplifies the complexity of creating maps for the Web.

In one of the following manuals we will provide a sample configuration to set up a QGIS Server. But for now we recommend to read one of the following URLs to get more information:

- http://karlinapp.ethz.ch/qgis_wms/
- 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 Sample installation on Debian Squeeze

At this point we will give a short and simple sample installation howto for Debian Squeeze. Many other OS provide packages for QGIS Server, too. If you have to build it all from source, please refer to the URLs above.

Apart from QGIS and QGIS Server you need a webserver, in our case apache2. You can install all packages with aptitude or apt-get install together with other necessary dependency packages. After installation you should test, if the webserver and QGIS Server works as expected. Make sure the apache server is running with `/etc/init.d/apache2 start`. Open a web browser and type URL: `http://localhost`. If apache is up, you should see the message 'It works!'.

Now we test the QGIS Server installation. The `qgis_mapserv.fcgi` is available at `/usr/lib/cgi-bin/qgis_mapserv.fcgi` and provides a standard wms that shows the state boundaries of Alaska. Add the WMS with the URL `http://localhost/cgi-bin/qgis_mapserv.fcgi` as described in *Selecting WMS/WMTS Servers*.

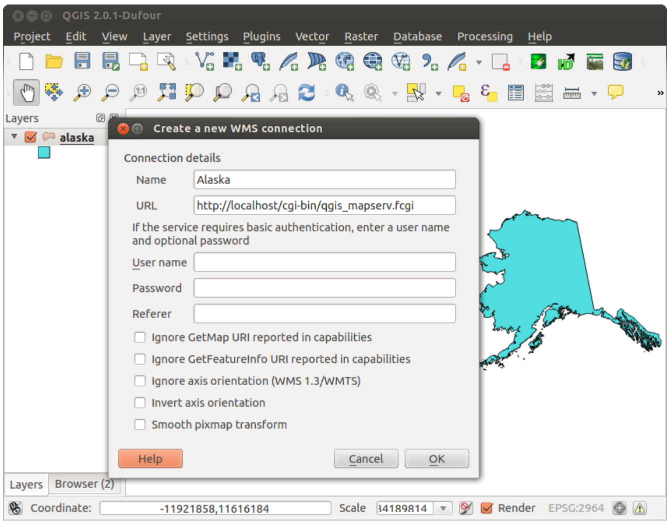


Figure 14.5: Standard WMS with USA boundaries included in the QGIS Server (KDE)

14.2.2 Creating a WMS/WFS from a QGIS project

To provide a new QGIS Server WMS or WFS we have to create a QGIS project file with some data. Here we use the 'alaska' shapefile from the QGIS sample dataset. Define the colors and styles of the layers in QGIS and define the project CRS, if not already done.

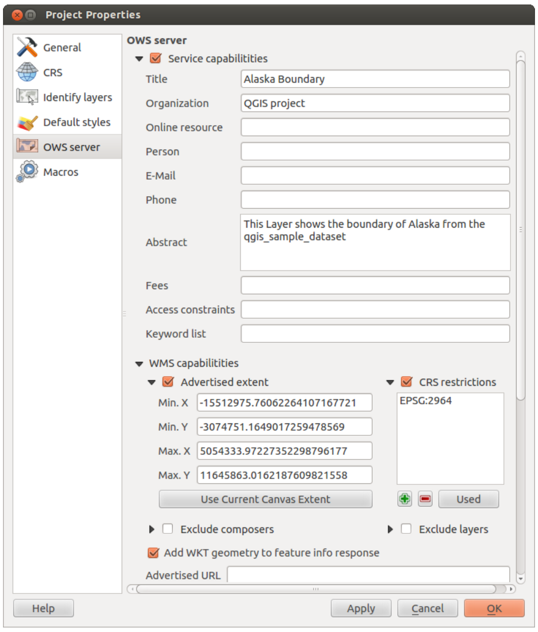


Figure 14.6: Definitions for a QGIS Server WMS/WFS project (KDE)

Then go to the *OWS Server* menu of the *Project* → *Project Properties* dialog and give some information about the OWS in the fields under *Service Capabilities*. This will appear in the GetCapabilities response of the WMS or WFS. If you don't check *Service capabilities* QGIS Server will use the information given in the `wms_metadata.xml` file located in the `cgi-bin` folder.

In the *WMS capabilities* section you can define the extent advertised in the WMS GetCapabilities response by entering the minimum and maximum X and Y values in the fields under *Advertised extent*. Clicking *Use Current Canvas Extent* sets these values to the extent currently displayed in the QGIS map canvas. By checking *CRS restrictions* you can restrict in which coordinate reference systems (CRS) QGIS Server will offer to render maps.

Use the button below to select those CRS from the Coordinate Reference System Selector, or click *Used* to add the CRS used in the QGIS project to the list.

If you have print composers defined in your project they will be listed in the GetCapabilities response, and they can be used by the GetPrint request to create prints, using one of the print composer layouts as a template. This is a QGIS specific extension to the WMS 1.3.0 specification. If you want to exclude any print composer from being published by the WMS, check *Exclude composers* and click the button below. Then select a print composer from the *Select print composer* dialog in order to add it to the excluded composers list.

If you want to exclude any layer or layer group from being published by the WMS, check *Exclude Layers* and click the button below. This opens the *Select restricted layers and groups* dialog which allows you to choose the layers and groups that you don't want to be published. Use the shift or control key if you want to select multiple entries at once.

If you wish you can check *Add WKT geometry to feature info response*. This will include in the GetFeatureInfo response the geometries of the features in a text format. If you want QGIS Server to advertise specific request URLs in the WMS GetCapabilities response, enter the corresponding URL in the *Advertised URL* field. Furthermore you can restrict the maximum size of the maps returned by the GetMap request by entering the maximum width and height into the respective fields under *Maximums for GetMap request*.

In the *WFS capabilities* area you can select the layers that you want to provide as WFS, and specify if they will allow the update, insert and delete operations. If you enter a URL in the *Advertised URL* field of the *WFS capabilities* section, QGIS Server will advertise this specific URL in the WFS GetCapabilities response.

Now save the session in a project file `alaska.qgs`. To provide the project as a WMS/WFS, we create a new folder `/usr/lib/cgi-bin/project` with admin privileges and add the project file `alaska.qgs` and a copy of the `qgis_mapserv.fcgi` file - that's all.

Now we test our project WMS and WFS, add the WMS and WFS as described in *Loading WMS/WMTS Layers* and *WFS and WFS-T Client* to QGIS and load the WMS. The URL is:

```
http://localhost/cgi-bin/project/qgis_mapserv.fcgi
```

Fine tuning your OWS

For vector layers, the *Fields* menu of the *Layer* → *Properties* dialog allows you to define for each attribute if it will be published or not. By default all the attributes are published by your WMS and WFS. If you want a specific attribute not to be published, uncheck the corresponding check box in the *WMS* or *WFS* column.

You can overlay watermarks over the maps produced by your WMS by adding text annotations or SVG annotations to the project file. See *sec_annotations* for instructions on creating annotations. For annotations to be displayed as watermarks on the WMS output, the *Fixed map position* check box in the *Annotation text* dialog must be unchecked. This can be accessed by double clicking the annotation while one of the annotation tools is active. For SVG annotations you will either need to set the project to save absolute paths (in the *General* menu of the *Project* → *Project Properties* dialog) or to manually modify the path to the SVG image in a way that it represents a valid relative path.

Extra parameters supported by the WMS GetMap request

In the WMS GetMap request QGIS Server accepts a couple of extra parameters in addition to the standard parameters according to the OCG WMS 1.3.0 specification:

- **MAP** parameter: Similar to MapServer, the `MAP` parameter can be used to specify the path to the QGIS project file. You can specify an absolute path or a path relative to the location of the server executable (`qgis_mapserv.fcgi`). If not specified, QGIS Server searches for .qgs files in the directory where the server executable is located.

  Example:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\
  REQUEST=GetMap&MAP=/home/qgis/mymap.qgs&...
```

- **DPI** parameter: The `DPI` parameter can be used to specify the requested output resolution.

  Example:

  ```
  http://localhost/cgi-bin/qgis_mapserv.fcgi?REQUEST=GetMap&DPI=300&...
  ```

- **OPACITIES** parameter: Opacity can be set on layer or group level. Allowed values range from 0 (fully transparent) to 255 (fully opaque).

  Example:

  ```
  http://localhost/cgi-bin/qgis_mapserv.fcgi?\
    REQUEST=GetMap&LAYERS=mylayer1,mylayer2&OPACITIES=125,200&...
  ```

Working with GPS Data

15.1 GPS Plugin

15.1.1 What is GPS?

GPS, the Global Positioning System, is a satellite-based system that allows anyone with a GPS receiver to find their exact position anywhere in the world. It is used as an aid in navigation, for example in airplanes, in boats and by hikers. The GPS receiver uses the signals from the satellites to calculate its latitude, longitude and (sometimes) elevation. Most receivers also have the capability to store locations (known as **waypoints**), sequences of locations that make up a planned **route** and a tracklog or **track** of the receivers movement over time. Waypoints, routes and tracks are the three basic feature types in GPS data. QGIS displays waypoints in point layers while routes and tracks are displayed in linestring layers.

15.1.2 Loading GPS data from a file

There are dozens of different file formats for storing GPS data. The format that QGIS uses is called GPX (GPS eXchange format), which is a standard interchange format that can contain any number of waypoints, routes and tracks in the same file.

To load a GPX file you first need to load the plugin. *Plugins* → *Plugin Manager...* opens the Plugin Manager Dialog. Activate the *GPS Tools* checkbox. When this plugin is loaded two buttons with a small handheld GPS device will show up in the toolbar:

- Create new GPX Layer
- GPS Tools

For working with GPS data we provide an example GPX file available in the QGIS sample dataset: `qgis_sample_data/gps/national_monuments.gpx`. See Section *Sample Data* for more information about the sample data.

1. Select *Vector* → *GPS* → *GPS Tools* or click the GPS Tools icon in the toolbar and open the *Load GPX file* tab (see figure_GPS_1).
2. Browse to the folder `qgis_sample_data/gps/`, select the GPX file `national_monuments.gpx` and click **[Open]**.

Use the **[Browse...]** button to select the GPX file, then use the checkboxes to select the feature types you want to load from that GPX file. Each feature type will be loaded in a separate layer when you click **[OK]**. The file `national_monuments.gpx` only includes waypoints.

Note: GPS units allow to store data in different coordinate systems. When downloading a GPX file (from your GPS unit or a web site) and then loading it in QGIS, be sure that the data stored in the GPX

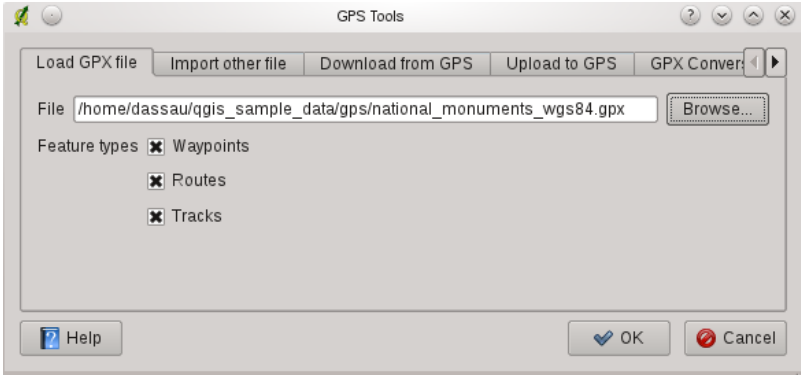


Figure 15.1: The *GPS Tools* dialog window

file uses WGS84 (latitude/longitude). QGIS expects this and it is the official GPX specification. See http://www.topografix.com/GPX/1/1/

15.1.3 GPSBabel

Since QGIS uses GPX files you need a way to convert other GPS file formats to GPX. This can be done for many formats using the free program GPSBabel, which is available at http://www.gpsbabel.org. This program can also transfer GPS data between your computer and a GPS device. QGIS uses GPSBabel to do these things, so it is recommended that you install it. However, if you just want to load GPS data from GPX files you will not need it. Version 1.2.3 of GPSBabel is known to work with QGIS, but you should be able to use later versions without any problems.

15.1.4 Importing GPS data

To import GPS data from a file that is not a GPX file, you use the tool *Import other file* in the GPS Tools dialog. Here you select the file that you want to import (and the file type), which feature type you want to import from it, where you want to store the converted GPX file and what the name of the new layer should be. Note that not all GPS data formats will support all three feature types, so for many formats you will only be able to choose between one or two types.

15.1.5 Downloading GPS data from a device

QGIS can use GPSBabel to download data from a GPS device directly as new vector layers. For this we use the *Download from GPS* tab of the GPS Tools dialog (see Figure_GPS_2). Here, we select the type of GPS device, the port that it is connected to (or usb if your GPS supports this), the feature type that you want to download, the GPX file where the data should be stored, and the name of the new layer.

The device type you select in the GPS device menu determines how GPSBabel tries to communicate with your GPS device. If none of the available types work with your GPS device you can create a new type (see section *Defining new device types*).

The port may be a file name or some other name that your operating system uses as a reference to the physical port in your computer that the GPS device is connected to. It may also be simply usb, for usb enabled GPS units.

- On Linux this is something like `/dev/ttyS0` or `/dev/ttyS1`
- On Windows it is `COM1` or `COM2`

When you click **[OK]** the data will be downloaded from the device and appear as a layer in QGIS.

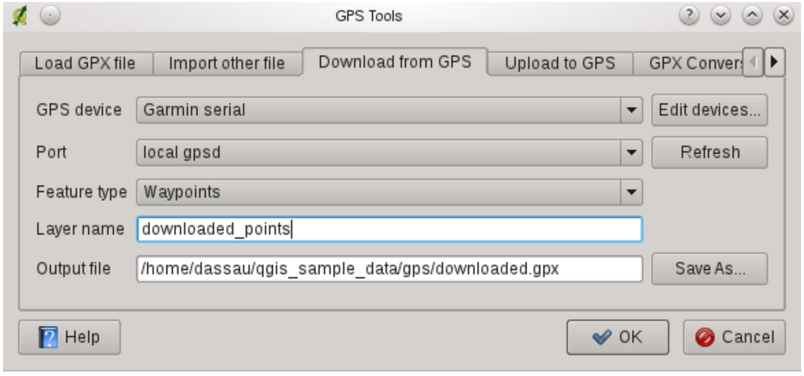


Figure 15.2: The download tool

15.1.6 Uploading GPS data to a device

You can also upload data directly from a vector layer in QGIS to a GPS device using the *Upload to GPS* tab of the GPS Tools dialog. To do this you simply select the layer that you want to upload (which must be a GPX layer), your GPS device type, and the port (or usb) that it is connected to. Just as with the download tool you can specify new device types if your device isn't in the list.

This tool is very useful in combination with the vector editing capabilities of QGIS. It allows you to load a map, create waypoints and routes, and then upload them and use them on your GPS device.

15.1.7 Defining new device types

There are lots of different types of GPS devices. The QGIS developers can't test all of them, so if you have one that does not work with any of the device types listed in the *Download from GPS* and *Upload to GPS* tools you can define your own device type for it. You do this by using the GPS device editor, which you start by clicking the **[Edit devices]** button in the download or the upload tabs.

To define a new device you simply click the **[New device]** button, enter a name, a download command and an upload command for your device, and click the **[Update device]** button. The name will be listed in the device menus in the upload and download windows, and can be any string. The download command is the command that is used to download data from the device to a GPX file. This will probably be a GPSBabel command, but you can use any other command line program that can create a GPX file. QGIS will replace the keywords `%type`, `%in`, and `%out` when it runs the command.

`%type` will be replaced by `-w` if you are downloading waypoints, `-r` if you are downloading routes and `-t` if you are downloading tracks. These are command line options that tell GPSBabel which feature type to download.

`%in` will be replaced by the port name that you choose in the download window and `%out` will be replaced by the name you choose for the GPX file that the downloaded data should be stored in. So if you create a device type with the download command `gpsbabel %type -i garmin -o gpx %in %out` (this is actually the download command for the predefined device type 'Garmin serial') and then use it to download waypoints from port `/dev/ttyS0` to the file `output.gpx`, QGIS will replace the keywords and run the command `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

The upload command is the command that is used to upload data to the device. The same keywords are used, but `%in` is now replaced by the name of the GPX file for the layer that is being uploaded, and `%out` is replaced by the port name.

You can learn more about GPSBabel and it's available command line options at http://www.gpsbabel.org.

Once you have created a new device type it will appear in the device lists for the download and upload tools.

15.2 Live GPS tracking

To activate Live GPS tracking in QGIS you need to select *Settings* → *Panels* *GPS information*. You will get a new docked window on the left side of the canvas.

There are 4 possible screens in this GPS tracking window:

- GPS position coordinates and for manually entering Vertices and Features.
- GPS signal strength of satellite connections.
- GPS polar screen showing number and polar position of satellites.
- GPS options screen (see figure_gps_options).

With a plugged in GPS receiver (has to be supported by your operating system) a simple click on **[Connect]** connects the GPS to QGIS. A second click (now on **[Disconnect]**) disconnects the GPS-receiver from your computer. For GNU/Linux gpsd support is integrated to support connection to most GPS receivers. Therefore you first have to configure gpsd properly to connect QGIS to it.

> **Warning:** If you want to record your position to the canvas you have to create a new vector layer first and switch it to editable status to be able to record your track.

15.2.1 Position and additional attributes

If the GPS is receiving signals from satellites you will see your position in latitude, longitude and altitude together with additional attributes.

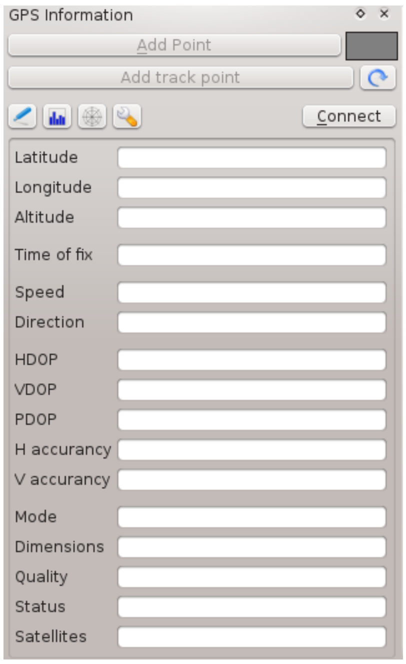


Figure 15.3: GPS tracking position and additional attributes

15.2.2 GPS signal strength

Here you can see the signal strenght of the satellites you are receiving signals from.

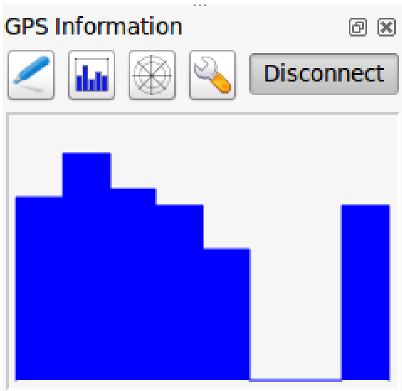


Figure 15.4: GPS tracking signal strength

15.2.3 GPS polar window

If you want to know where in the sky all the connected satellites are, you have to switch to the polar screen. You can also see the ID numbers of the satellites you are receiving signals from.

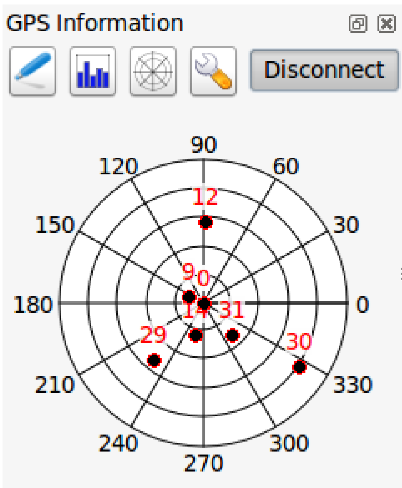


Figure 15.5: GPS tracking polar window

15.2.4 GPS options

In case of connection problems you can switch between:

- *Autodetect*
- *Internal*
- *Serial device*
- *gpsd* (selecting Host, Port and Device your GPS is connected to)

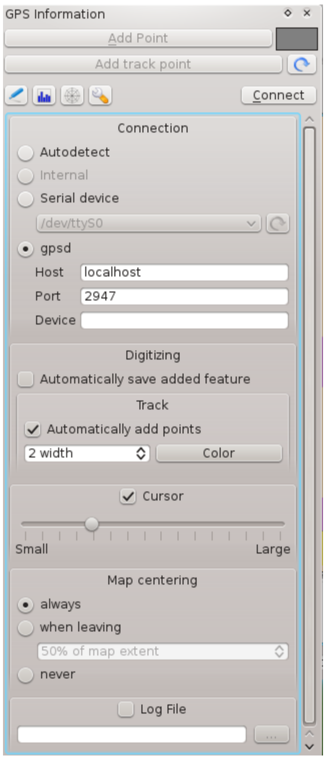


Figure 15.6: GPS tracking options window

A click on **[Connect]** again initiates the connection to the GPS receiver.

You can activate *Automatically save added features* when you are in editing mode. Or you can can activate *Automatically add points* to the map canvas with a certain width and color.

Activating *Cursor* you can use a slider to shrink and grow the position cursor on the canvas.

Activating *Map centering* allows to decide in which way the canvas will be updated. This includes 'always', 'when leaving' if your recorded coordinates start either to move out of canvas or 'never' to keep map extent.

Finally you can activate *Log file* and define a path and a file where log messages about the gps tracking a logged.

If you want to set a feature manually you have to go back to Position and click on **[Add Point]** or **[Add track point]**.

GRASS GIS Integration

The GRASS plugin provides access to GRASS GIS (see GRASS-PROJECT *Literature and Web References*) databases and functionalities. This includes visualization of GRASS raster and vector layers, digitizing vector layers, editing vector attributes, creating new vector layers and analysing GRASS 2D and 3D data with more than 400 GRASS modules.

In this Section we'll introduce the plugin functionalities and give some examples on managing and working with GRASS data. Following main features are provided with the toolbar menu, when you start the GRASS plugin, as described in section *sec_starting_grass* :

- Open mapset
- New mapset
- Close mapset
- Add GRASS vector layer
- Add GRASS raster layer
- Create new GRASS vector
- Edit GRASS vector layer
- Open GRASS tools
- Display current GRASS region
- Edit current GRASS region

16.1 Starting the GRASS plugin

To use GRASS functionalities and/or visualize GRASS vector and raster layers in QGIS, you must select and load the GRASS plugin with the Plugin Manager. Therefore go to the menu *Plugins* → *Manage Plugins*, select *GRASS* and click **[OK]**.

You can now start loading raster and vector layers from an existing GRASS `LOCATION` (see section *sec_load_grassdata*). Or you create a new GRASS `LOCATION` with QGIS (see section *Creating a new GRASS LOCATION*) and import some raster and vector data (see Section *Importing data into a GRASS LOCATION*) for further analysis with the GRASS Toolbox (see section *The GRASS toolbox*).

16.2 Loading GRASS raster and vector layers

With the GRASS plugin, you can load vector or raster layers using the appropriate button on the toolbar menu. As an example we use the QGIS alaska dataset (see Section *Sample Data*). It includes a small sample GRASS `LOCATION` with 3 vector layers and 1 raster elevation map.

1. Create a new folder `grassdata`, download the QGIS 'Alaska' dataset `qgis_sample_data.zip` from http://download.osgeo.org/qgis/data/ and unzip the file into `grassdata`.
2. Start QGIS.
3. If not already done in a previous QGIS session, load the GRASS plugin clicking on *Plugins* → *Manage Plugins* and activate *GRASS*. The GRASS toolbar appears in the QGIS main window.
4. In the GRASS toolbar, click the Open mapset icon to bring up the *MAPSET* wizard.
5. For `Gisdbase` browse and select or enter the path to the newly created folder `grassdata`.
6. You should now be able to select the *LOCATION* `alaska` and the *MAPSET* `demo`.
7. Click **[OK]**. Notice that some previously disabled tools in the GRASS toolbar are now enabled.
8. Click on Add GRASS raster layer, choose the map name `gtopo30` and click **[OK]**. The elevation layer will be visualized.
9. Click on Add GRASS vector layer, choose the map name `alaska` and click **[OK]**. The Alaska boundary vector layer will be overlayed on top of the `gtopo30` map. You can now adapt the layer properties as described in chapter *The Vector Properties Dialog*, e.g. change opacity, fill and outline color.
10. Also load the other two vector layers `rivers` and `airports` and adapt their properties.

As you see, it is very simple to load GRASS raster and vector layers in QGIS. See following sections for editing GRASS data and creating a new `LOCATION`. More sample GRASS `LOCATIONs` are available at the GRASS website at http://grass.osgeo.org/download/sample-data/.

Tip: GRASS Data Loading

If you have problems loading data or QGIS terminates abnormally, check to make sure you have loaded the GRASS plugin properly as described in section *sec_starting_grass*.

16.3 GRASS LOCATION and MAPSET

GRASS data are stored in a directory referred to as GISDBASE. This directory often called `grassdata`, must be created before you start working with the GRASS plugin in QGIS. Within this directory, the GRASS GIS data are organized by projects stored in subdirectories called `LOCATION`. Each `LOCATION` is defined by its coordinate system, map projection and geographical boundaries. Each `LOCATION` can have several `MAPSETs` (subdirectories of the `LOCATION`) that are used to subdivide the project into different topics, subregions, or as workspaces for individual team members (Neteler & Mitasova 2008 *Literature and Web References*). In order to analyze vector and raster layers with GRASS modules, you must import them into a GRASS `LOCATION` (This is not strictly true - with the GRASS modules `r.external` and `v.external` you can create read-only links to external GDAL/OGR-supported data sets without importing them. But because this is not the usual way for beginners to work with GRASS, this functionality will not be described here.).

16.3.1 Creating a new GRASS LOCATION

As an example here is how the sample GRASS `LOCATION alaska`, which is projected in Albers Equal Area projection with unit feet was created for the QGIS sample dataset. This sample GRASS `LOCATION alaska`

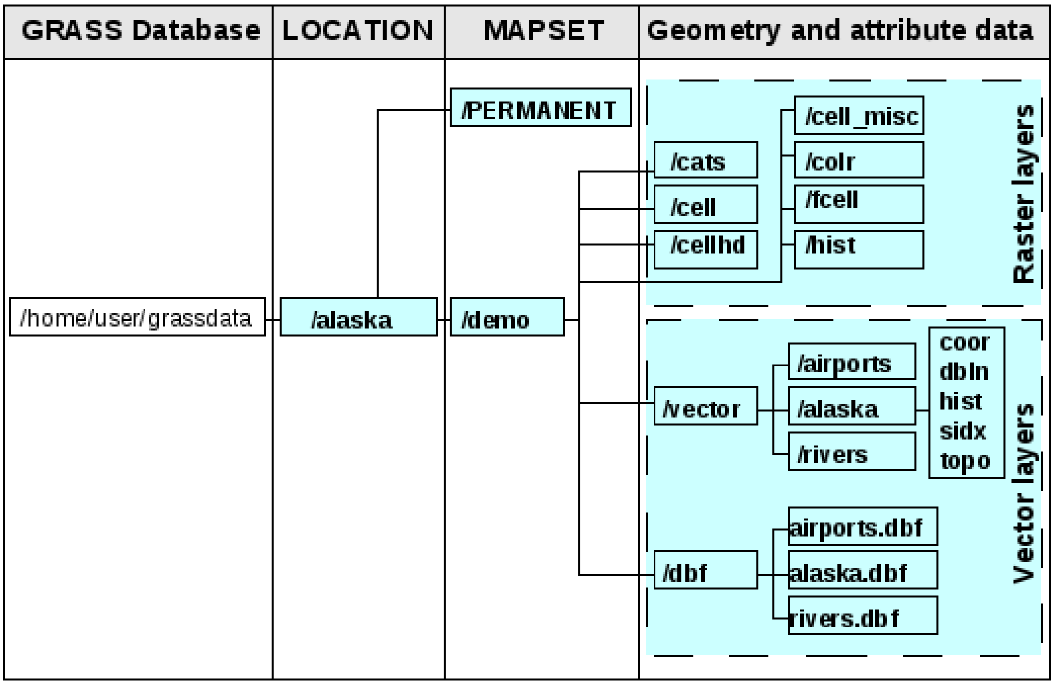


Figure 16.1: GRASS data in the alaska LOCATION

will be used for all examples and exercises in the following GRASS GIS related chapters. It is useful to download and install the dataset on your computer *Sample Data*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Visualize the `alaska.shp` shapefile (see Section *vector_load_shapefile*) from the QGIS alaska dataset *Sample Data*.
3. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
4. Select an existing GRASS database (GISDBASE) folder `grassdata` or create one for the new `LOCATION` using a file manager on your computer. Then click **[Next]**.
5. We can use this wizard to create a new `MAPSET` within an existing `LOCATION` (see section *Adding a new MAPSET*) or to create a new `LOCATION` altogether. Select *Create new location* (see figure_grass_location_2).
6. Enter a name for the `LOCATION` - we used 'alaska' and click **[Next]**.
7. Define the projection by clicking on the radio button *Projection* to enable the projection list.
8. We are using Albers Equal Area Alaska (feet) projection. Since we happen to know that it is represented by the EPSG ID 2964, we enter it in the search box. (Note: If you want to repeat this process for another `LOCATION` and projection and haven't memorized the EPSG ID, click on the CRS Status icon in the lower right-hand corner of the status bar (see Section *Working with Projections*)).
9. In *Filter* insert 2964 to select the projection.
10. Click **[Next]**.
11. To define the default region, we have to enter the `LOCATION` bounds in north, south, east, and west direction. Here we simply click on the button **[Set current QGIS extent]**, to apply the extend of the loaded layer `alaska.shp` as the GRASS default region extend.
12. Click **[Next]**.
13. We also need to define a `MAPSET` within our new `LOCATION`. You can name it whatever you like - we used 'demo' (when creating a new `LOCATION`). GRASS automatically creates a special `MAPSET` called

PERMANENT designed to store the core data for the project, its default spatial extend and coordinate system definitions (Neteler & Mitasova 2008 *Literature and Web References*).

14. Check out the summary to make sure it's correct and click **[Finish]**.
15. The new LOCATION 'alaska' and two MAPSETs 'demo' and 'PERMANENT' are created. The currently opened working set is 'demo', as you defined.
16. Notice that some of the tools in the GRASS toolbar that were disabled are now enabled.

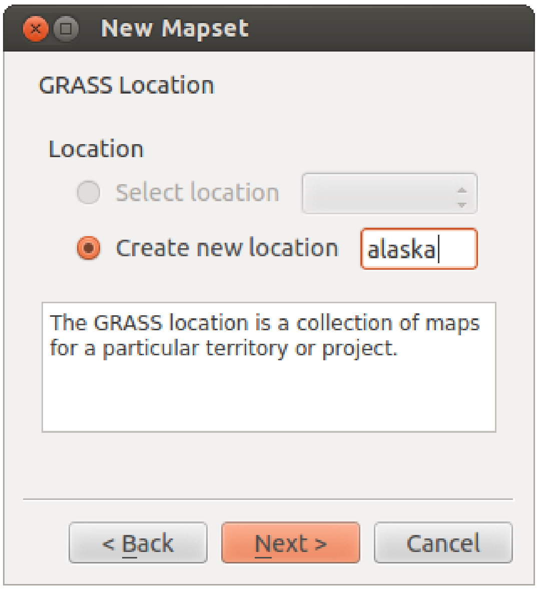


Figure 16.2: Creating a new GRASS LOCATION or a new MAPSET in QGIS

If that seemed like a lot of steps, it's really not all that bad and a very quick way to create a LOCATION. The LOCATION 'alaska' is now ready for data import (see section *Importing data into a GRASS LOCATION*). You can also use the already existing vector and raster data in the sample GRASS LOCATION 'alaska' included in the QGIS 'Alaska' dataset *Sample Data* and move on to Section *The GRASS vector data model*.

16.3.2 Adding a new MAPSET

A user has only write access to a GRASS MAPSET he created. This means that besides access to his own MAPSET, each user can read maps in other user's MAPSETs, but he can modify or remove only the maps in his own MAPSET.

All MAPSETs include a WIND file that stores the current boundary coordinate values and the currently selected raster resolution (Neteler & Mitasova 2008 *Literature and Web References*, see Section *The GRASS region tool*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
3. Select the GRASS database (GISDBASE) folder grassdata with the LOCATION 'alaska', where we want to add a further MAPSET, called 'test'.
4. Click **[Next]**.
5. We can use this wizard to create a new MAPSET within an existing LOCATION or to create a new LOCATION altogether. Click on the radio button *Select location* (see figure_grass_location_2) and click **[Next]**.
6. Enter the name text for the new MAPSET. Below in the wizard you see a list of existing MAPSETs and its owners.
7. Click **[Next]**, check out the summary to make sure it's all correct and click **[Finish]**.

16.4 Importing data into a GRASS LOCATION

This Section gives an example how to import raster and vector data into the 'alaska' GRASS `LOCATION` provided by the QGIS 'Alaska' dataset. Therefore we use a landcover raster map `landcover.img` and a vector GML file `lakes.gml` from the QGIS 'Alaska' dataset *Sample Data*.

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click the Open MAPSET icon to bring up the *MAPSET* wizard.
3. Select as GRASS database the folder `grassdata` in the QGIS alaska dataset, as `LOCATION` 'alaska', as `MAPSET` 'demo' and click **[OK]**.
4. Now click the Open GRASS tools icon. The GRASS Toolbox (see section *The GRASS toolbox*) dialog appears.
5. To import the raster map `landcover.img`, click the module `r.in.gdal` in the *Modules Tree* tab. This GRASS module allows to import GDAL supported raster files into a GRASS `LOCATION`. The module dialog for `r.in.gdal` appears.
6. Browse to the folder `raster` in the QGIS 'Alaska' dataset and select the file `landcover.img`.
7. As raster output name define `landcover_grass` and click **[Run]**. In the *Output* tab you see the currently running GRASS command `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. When it says **Succesfully finished** click **[View output]**. The `landcover_grass` raster layer is now imported into GRASS and will be visualized in the QGIS canvas.
9. To import the vector GML file `lakes.gml`, click the module `v.in.ogr` in the *Modules Tree* tab. This GRASS module allows to import OGR supported vector files into a GRASS `LOCATION`. The module dialog for `v.in.ogr` appears.
10. Browse to the folder `gml` in the QGIS 'Alaska' dataset and select the file `lakes.gml` as OGR file.
11. As vector output name define `lakes_grass` and click **[Run]**. You don't have to care about the other options in this example. In the *Output* tab you see the currently running GRASS command `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes\_grass`.
12. When it says **Succesfully finished** click **[View output]**. The `lakes_grass` vector layer is now imported into GRASS and will be visualized in the QGIS canvas.

16.5 The GRASS vector data model

It is important to understand the GRASS vector data model prior to digitizing.

In general, GRASS uses a topological vector model.

This means that areas are not represented as closed polygons, but by one or more boundaries. A boundary between two adjacent areas is digitized only once, and it is shared by both areas. Boundaries must be connected and closed without gaps. An area is identified (and labeled) by the **centroid** of the area.

Besides boundaries and centroids, a vector map can also contain points and lines. All these geometry elements can be mixed in one vector and will be represented in different so called 'layers' inside one GRASS vector map. So in GRASS a layer is not a vector or raster map but a level inside a vector layer. This is important to distinguish carefully (Although it is possible to mix geometry elements, it is unusual and even in GRASS only used in special cases such as vector network analysis. Normally you should prefere to store different geometry elements in different layers.).

It is possible to store several 'layers' in one vector dataset. For example, fields, forests and lakes can be stored in one vector. Adjacent forest and lake can share the same boundary, but they have separate attribute tables. It is also possible to attach attributes to boundaries. For example, the boundary between lake and forest is a road, so it can have a different attribute table.

The 'layer' of the feature is defined by 'layer' inside GRASS. 'Layer' is the number which defines if there are more than one layer inside the dataset, e.g. if the geometry is forest or lake. For now, it can be only a number, in the future GRASS will also support names as fields in the user interface.

Attributes can be stored inside the GRASS `LOCATION` as DBase or SQLITE3 or in external database tables, for example PostgreSQL, MySQL, Oracle, etc.

Attributes in database tables are linked to geometry elements using a 'category' value.

'Category' (key, ID) is an integer attached to geometry primitives, and it is used as the link to one key column in the database table.

Tip: Learning the GRASS Vector Model

The best way to learn the GRASS vector model and its capabilities is to download one of the many GRASS tutorials where the vector model is described more deeply. See http://grass.osgeo.org/documentation/manuals/ for more information, books and tutorials in several languages.

16.6 Creating a new GRASS vector layer

To create a new GRASS vector layer with the GRASS plugin click the Create new GRASS vector toolbar icon. Enter a name in the text box and you can start digitizing point, line or polygon geometries, following the procedure described in Section *Digitizing and editing a GRASS vector layer*.

In GRASS it is possible to organize all sort of geometry types (point, line and area) in one layer, because GRASS uses a topological vector model, so you don't need to select the geometry type when creating a new GRASS vector. This is different from Shapefile creation with QGIS, because Shapefiles use the Simple Feature vector model (see Section *Creating new Vector layers*).

Tip: Creating an attribute table for a new GRASS vector layer

If you want to assign attributes to your digitized geometry features, make sure to create an attribute table with columns before you start digitizing (see figure_grass_digitizing_5).

16.7 Digitizing and editing a GRASS vector layer

The digitizing tools for GRASS vector layers are accessed using the Edit GRASS vector layer icon on the toolbar. Make sure you have loaded a GRASS vector and it is the selected layer in the legend before clicking on the edit tool. Figure figure_grass_digitizing_2 shows the GRASS edit dialog that is displayed when you click on the edit tool. The tools and settings are discussed in the following sections.

Tip: Digitizing polygons in GRASS

If you want to create a polygon in GRASS, you first digitize the boundary of the polygon, setting the mode to 'No category'. Then you add a centroid (label point) into the closed boundary, setting the mode to 'Next not used'. The reason is, that a topological vector model links attribute information of a polygon always to the centroid and not to the boundary.

Toolbar

In figure_grass_digitizing_1 you see the GRASS digitizing toolbar icons provided by the GRASS plugin. Table table_grass_digitizing_1 explains the available functionalities.

Figure 16.3: GRASS Digitizing Toolbar

Icon	Tool	Purpose
	New Point	Digitize new point
	New Line	Digitize new line
	New Boundary	Digitize new boundary (finish by selecting new tool)
	New Centroid	Digitize new centroid (label existing area)
	Move vertex	Move one vertex of existing line or boundary and identify new position
	Add vertex	Add a new vertex to existing line
	Delete vertex	Delete vertex from existing line (confirm selected vertex by another click)
	Move element	Move selected boundary, line, point or centroid and click on new position
	Split line	Split an existing line to 2 parts
	Delete element	Delete existing boundary, line, point or centroid (confirm selected element by another click)
	Edit attributes	Edit attributes of selected element (note that one element can represent more features, see above)
	Close	Close session and save current status (rebuilds topology afterwards)

Table GRASS Digitizing 1: GRASS Digitizing Tools

Category Tab

The *Category* tab allows you to define the way in which the category values will be assigned to a new geometry element.

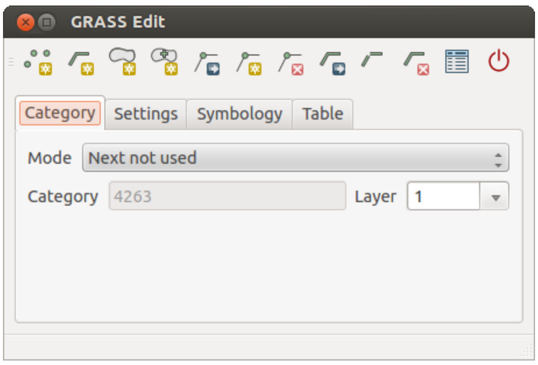


Figure 16.4: GRASS Digitizing Category Tab

- **Mode**: what category value shall be applied to new geometry elements.
  - Next not used - apply next not yet used category value to geometry element.
  - Manual entry - manually define the category value for the geometry element in the 'Category'-entry field.
  - No category - Do not apply a category value to the geometry element. This is e.g. used for area boundaries, because the category values are connected via the centroid.

- **Category** - A number (ID) is attached to each digitized geometry element. It is used to connect each geometry element with its attributes.
- **Field (layer)** - Each geometry element can be connected with several attribute tables using different GRASS geometry layers. Default layer number is 1.

Tip: Creating an additional GRASS 'layer' with QGIS

If you would like to add more layers to your dataset, just add a new number in the 'Field (layer)' entry box and press return. In the Table tab you can create your new table connected to your new layer.

Settings Tab

The *Settings* tab allows you to set the snapping in screen pixels. The threshold defines at what distance new points or line ends are snapped to existing nodes. This helps to prevent gaps or dangles between boundaries. The default is set to 10 pixels.

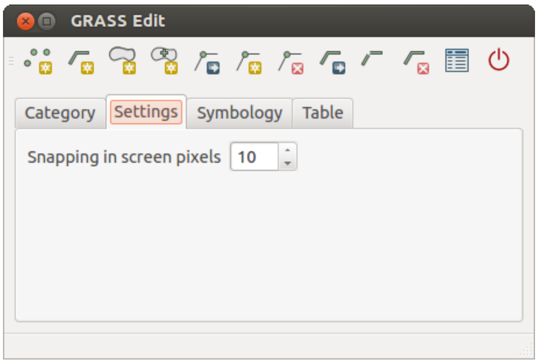


Figure 16.5: GRASS Digitizing Settings Tab

Symbology Tab

The *Symbology* tab allows you to view and set symbology and color settings for various geometry types and their topological status (e.g. closed / opened boundary).

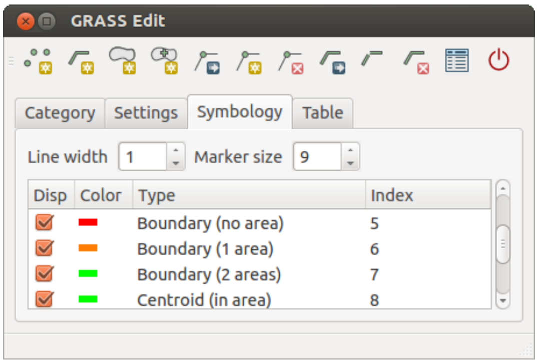


Figure 16.6: GRASS Digitizing Symbolog Tab

Table Tab

The *Table* tab provides information about the database table for a given 'layer'. Here you can add new columns to an existing attribute table, or create a new database table for a new GRASS vector layer (see Section *Creating a new GRASS vector layer*).

Tip: GRASS Edit Permissions

You must be the owner of the GRASS `MAPSET` you want to edit. It is impossible to edit data layers in a `MAPSET` that is not yours, even if you have write permissions.

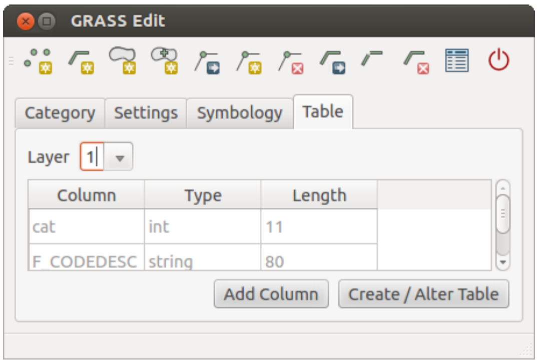


Figure 16.7: GRASS Digitizing Table Tab

16.8 The GRASS region tool

The region definition (setting a spatial working window) in GRASS is important for working with raster layers. Vector analysis is by default not limited to any defined region definitions. But all newly-created rasters will have the spatial extension and resolution of the currently defined GRASS region, regardless of their original extension and resolution. The current GRASS region is stored in the `$LOCATION/$MAPSET/WIND` file, and it defines north, south, east and west bounds, number of columns and rows, horizontal and vertical spatial resolution.

It is possible to switch on/off the visualization of the GRASS region in the QGIS canvas using the Display current GRASS region button.

With the Edit current GRASS region icon you can open a dialog to change the current region and the symbology of the GRASS region rectangle in the QGIS canvas. Type in the new region bounds and resolution and click **[OK]**. It also allows to select a new region interactively with your mouse on the QGIS canvas. Therefore click with the left mouse button in the QGIS canvas, open a rectangle, close it using the left mouse button again and click **[OK]**.

The GRASS module `g.region` provide a lot more parameters to define an appropriate region extend and resolution for your raster analysis. You can use these parameters with the GRASS Toolbox, described in Section *The GRASS toolbox*.

16.9 The GRASS toolbox

The Open GRASS Tools box provides GRASS module functionalities to work with data inside a selected GRASS `LOCATION` and `MAPSET`. To use the GRASS toolbox you need to open a `LOCATION` and `MAPSET` where you have write-permission (usually granted, if you created the `MAPSET`). This is necessary, because new raster or vector layers created during analysis need to be written to the currently selected `LOCATION` and `MAPSET`.

The GRASS Shell inside the GRASS Toolbox provides access to almost all (more than 330) GRASS modules through a command line interface. To offer a more user friendly working environment, about 200 of the available GRASS modules and functionalities are also provided by graphical dialogs within the GRASS plugin Toolbox.

16.9.1 Working with GRASS modules

The GRASS Shell inside the GRASS Toolbox provides access to almost all (more than 300) GRASS modules in a command line interface. To offer a more user friendly working environment, about 200 of the available GRASS modules and functionalities are also provided by graphical dialogs.

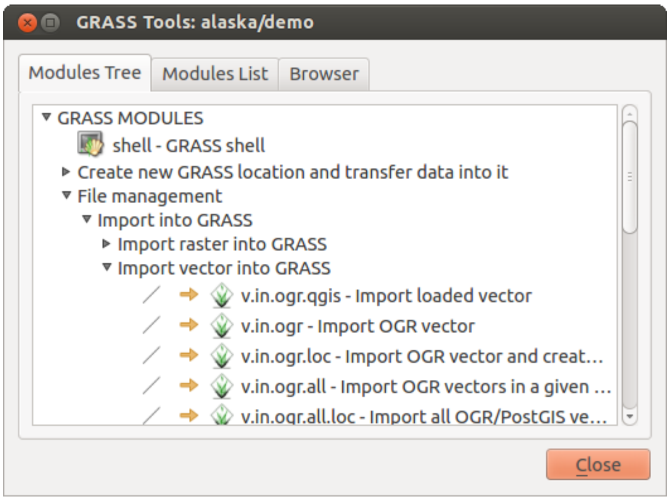


Figure 16.8: GRASS Toolbox and Module Tree

A complete list of GRASS modules available in the graphical Toolbox in QGIS version 2.0 is available in the GRASS wiki (http://grass.osgeo.org/wiki/GRASS-QGIS_relevant_module_list).

It is also possible to customize the GRASS Toolbox content. This procedure is described in Section *Customizing the GRASS Toolbox*.

As shown in figure_grass_toolbox_1 , you can look for the appropriate GRASS module using the thematically grouped *Modules Tree* or the searchable *Modules List* tab.

Clicking on a graphical module icon a new tab will be added to the toolbox dialog providing three new sub-tabs *Options*, *Output* and *Manual*.

Options

The *Options* tab provides a simplified module dialog where you can usually select a raster or vector layer visualized in the QGIS canvas and enter further module specific parameters to run the module.

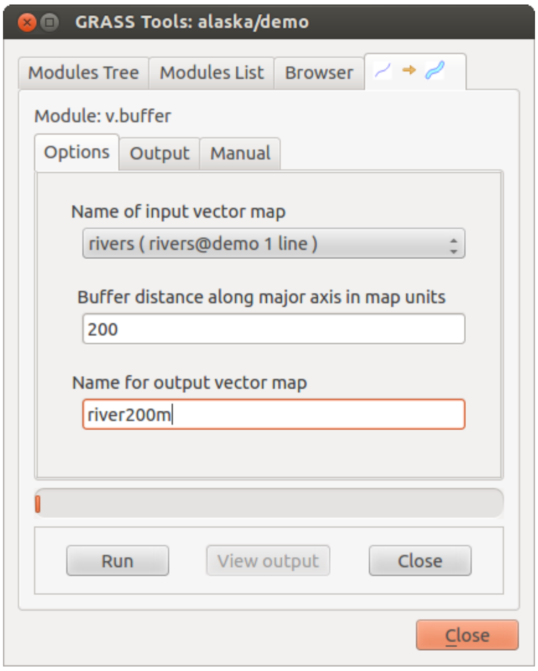


Figure 16.9: GRASS Toolbox Module Options

The provided module parameters are often not complete to keep the dialog clear. If you want to use further module parameters and flags, you need to start the GRASS Shell and run the module in the command line.

A new feature since QGIS 1.8 is the support for a *show advanced options* button below the simplified module dialog in the *Options* tab. At the moment it is only added to the module v.in.ascii as an example use, but will probably be part of more / all modules in the GRASS toolbox in future versions of QGIS. This allows to use the complete GRASS module options without the need to switch to the GRASS Shell.

Output

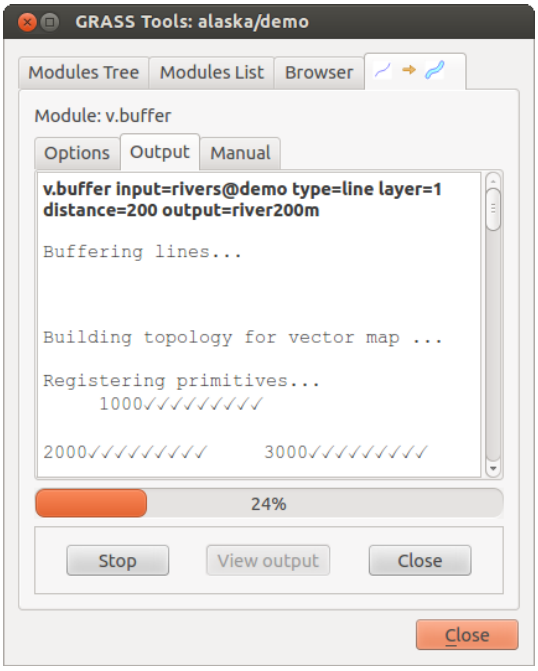


Figure 16.10: GRASS Toolbox Module Output

The *Output* tab provides information about the output status of the module. When you click the **[Run]** button, the module switches to the *Output* tab and you see information about the analysis process. If all works well, you will finally see a `Successfully finished` message.

Manual

The *Manual* tab shows the HTML help page of the GRASS module. You can use it to check further module parameters and flags or to get a deeper knowledge about the purpose of the module. At the end of each module manual page you see further links to the `Main Help index`, the `Thematic index` and the `Full index`. These links provide the same information as if you use the module `g.manual`.

Tip: Display results immediately

If you want to display your calculation results immediately in your map canvas, you can use the 'View Output' button at the bottom of the module tab.

16.9.2 GRASS module examples

The following examples will demonstrate the power of some of the GRASS modules.

Creating contour lines

The first example creates a vector contour map from an elevation raster (DEM). Assuming you have the Alaska `LOCATION` set up as explained in Section *Importing data into a GRASS LOCATION*.

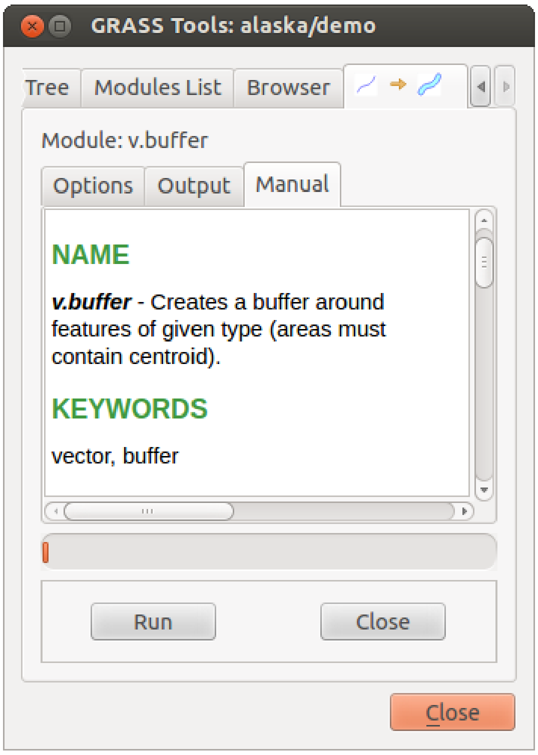


Figure 16.11: GRASS Toolbox Module Manual

- First open the location by clicking the Open mapset button and choosing the Alaska location.
- Now load the `gtopo30` elevation raster by clicking Add GRASS raster layer and selecting the `gtopo30` raster from the demo location.
- Now open the Toolbox with the Open GRASS tools button.
- In the list of tool categories double click *Raster* → *Surface Management* → *Generate vector contour lines*.
- Now a single click on the tool **r.contour** will open the tool dialog as explained above *Working with GRASS modules*. The `gtopo30` raster should appear as the *Name of input raster*.
- Type into the *Increment between Contour levels* 1,00 the value 100. (This will create contour lines at intervals of 100 meters.)
- Type into the *Name for output vector map* the name `ctour_100`.
- Click **[Run]** to start the process. Wait for several moments until the message `Successfully finished` appears in the output window. Then click **[View Output]** and **[Close]**.

Since this is a large region, it will take a while to display. After it finishes rendering, you can open the layer properties window to change the line color so that the contours appear clearly over the elevation raster, as in *The Vector Properties Dialog*.

Next zoom in to a small mountainous area in the center of Alaska. Zooming in close you will notice that the contours have sharp corners. GRASS offers the **v.generalize** tool to slightly alter vector maps while keeping their overall shape. The tool uses several different algorithms with different purposes. Some of the algorithms (i.e. Douglas Peuker and Vertex reduction) simplify the line by removing some of the vertices. The resulting vector will load faster. This process will be used when you have a highly detailed vector, but you are creating a very small scale map, so the detail is unnecessary.

Tip: The simplify tool

Note that the QGIS fTools plugin has a *Simplify geometries* → tool that works just like the GRASS **v.generalize** Douglas-Peuker algorithm.

However, the purpose of this example is different. The contour lines created by `r.contour` have sharp angles that should be smoothed. Among the **v.generalize** algorithms there is Chaikens which does just that (also Hermite splines). Be aware that these algorithms can **add** additional vertices to the vector, causing it to load even more slowly.

- Open the GRASS toolbox and double click the categories *Vector* → *Develop map* → *Generalization*, then click on the **v.generalize** module to open its options window.
- Check that the 'ctour_100' vector appears as the *Name of input vector*.
- From the list of algorithms choose Chaiken's. Leave all other options at their default, and scroll down to the last row to enter in the field *Name for output vector map* 'ctour_100_smooth', and click **[Run]**.
- The process takes several moments. Once `Successfully finished` appears in the output windows, click **[View output]** and then **[close]**.
- You may change the color of the vector to display it clearly on the raster background and to contrast with the original contour lines. You will notice that the new contour lines have smoother corners than the original while staying faithful to the original overall shape.

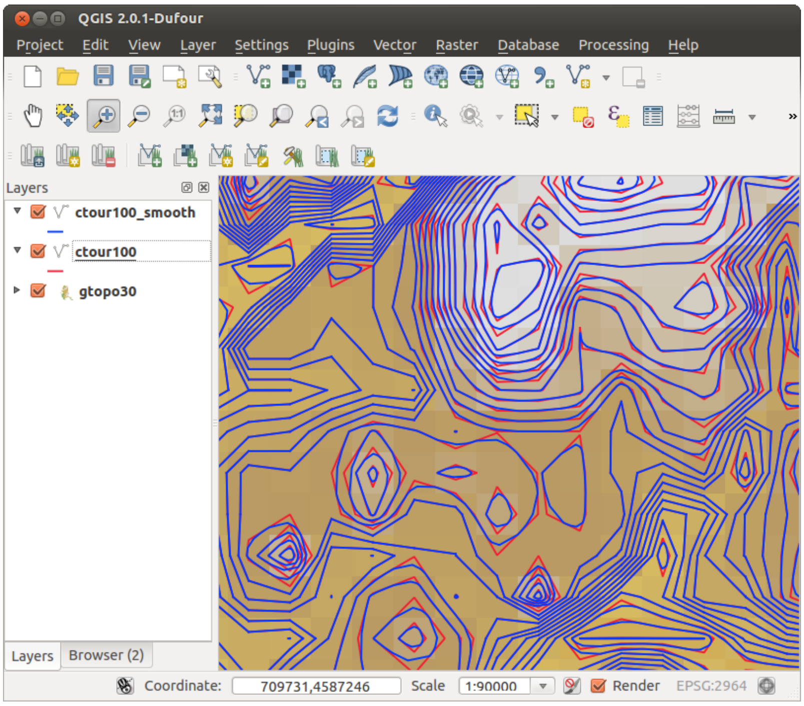


Figure 16.12: GRASS module v.generalize to smooth a vector map

Tip: Other uses for r.contour

The procedure described above can be used in other equivalent situations. If you have a raster map of precipitation data, for example, then the same method will be used to create a vector map of isohyetal (constant rainfall) lines.

Creating a Hillshade 3D effect

Several methods are used to display elevation layers and give a 3D effect to maps. The use of contour lines as shown above is one popular method often chosen to produce topographic maps. Another way to display a 3D effect is by hillshading. The hillshade effect is created from a DEM (elevation) raster by first calculating the slope and aspect of each cell, then simulating the sun's position in the sky and giving a reflectance value to each cell. Thus you get sun facing slopes lighted and the slopes facing away from the sun (in shadow) are darkened.

- Begin this example by loading the `gtopo30` elevation raster. Start the GRASS toolbox and under the Raster category double click to open *Spatial analysis* → *Terrain analysis*.
- Then click **r.shaded.relief** to open the module.
- Change the *azimuth angle* 1,00 270 to 315.
- Enter `gtopo30_shade` for the new hillshade raster, and click **[Run]**.
- When the process completes, add the hillshade raster to the map. You should see it displayed in grayscale.
- To view both the hill shading and the colors of the `gtopo30` together shift the hillshade map below the `gtopo30` map in the table of contents, then open the *Properties* window of `gtopo30`, switch to the *transparency* tab and set its transparency level to about 25%.

You should now have the `gtopo30` elevation with its colormap and transparency setting displayed **above** the grayscale hillshade map. In order to see the visual effects of the hillshading, turn off the `gtopo30_shade` map, then turn it back on.

Using the GRASS shell

The GRASS plugin in QGIS is designed for users who are new to GRASS, and not familiar with all the modules and options. As such, some modules in the toolbox do not show all the options available, and some modules do not appear at all. The GRASS shell (or console) gives the user access to those additional GRASS modules that do not appear in the toolbox tree, and also to some additional options to the modules that are in the toolbox with the simplest default parameters. This example demonstrates the use of an additional option in the **r.shaded.relief** module that was shown above.

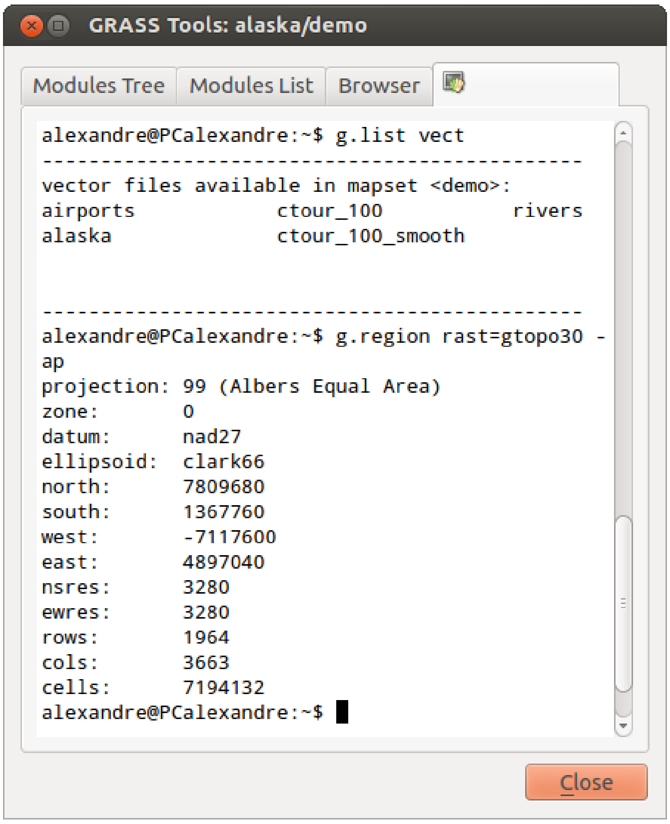


Figure 16.13: The GRASS shell, r.shaded.relief module

The module **r.shaded.relief** can take a parameter `zmult` which multiplies the elevation values relative to the X-Y coordinate units so that the hillshade effect is even more pronounced.

- Load the `gtopo30` elevation raster as above, then start the GRASS toolbox and click on the GRASS shell. In the shell window type the command `r.shaded.relief map=gtopo30 shade=gtopo30_shade2 azimuth=315 zmult=3` and press **[Enter]**.

- After the process finishes shift to the *Browse* tab and double click on the new `gtopo30_shade2` raster to display in QGIS.
- As explained above, shift the shaded relief raster below the gtopo30 raster in the Table of Contents, then check transparency of the colored gtopo30 layer. You should see that the 3D effect stands out more strongly compared to the first shaded relief map.

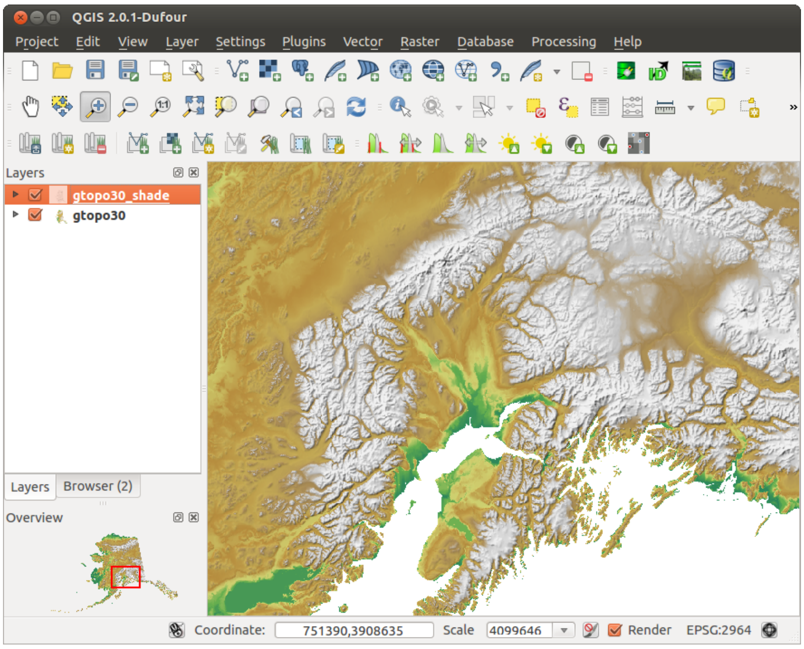


Figure 16.14: Displaying shaded relief created with the GRASS module r.shaded.relief

Raster statistics in a vector map

The next example shows how a GRASS module can aggregate raster data and add columns of statistics for each polygon in a vector map.

- Again using the Alaska data, refer to *Importing data into a GRASS LOCATION* to import the trees shapefile from the `shapefiles` directory into GRASS.
- Now an intermediary step is required: centroids must be added to the imported trees map to make it a complete GRASS area vector (including both boundaries and centroids).
- From the toolbox choose *Vector* → *Manage features*, and open the module **v.centroids**.
- Enter as the *output vector map* 'forest_areas' and run the module.
- Now load the `forest_areas` vector and display the types of forests - deciduous, evergreen, mixed - in different colors: In the layer *Properties* window, *Symbology* tab, choose from *Legend type* 'Unique value' and set the *Classification field* to 'VEGDESC'. (Refer to the explanation of the symbology tab :ref:sec_symbology in the vector section).
- Next reopen the GRASS toolbox and open *Vector* → *Vector update* by other maps.
- Click on the **v.rast.stats** module. Enter `gtopo30`, and `forest_areas`.
- Only one additional parameter is needed: Enter *column prefix* `elev`, and click **[run]**. This is a computationally heavy operation which will run for a long time (probably up to two hours).
- Finally open the `forest_areas` attribute table, and verify that several new columns have been added including `elev_min`, `elev_max`, `elev_mean` etc. for each forest polygon.

16.9.3 Working with the GRASS LOCATION browser

Another useful feature inside the GRASS Toolbox is the GRASS `LOCATION` browser. In figure_grass_module_7 you can see the current working `LOCATION` with its `MAPSETs`.

In the left browser windows you can browse through all `MAPSETs` inside the current `LOCATION`. The right browser window shows some meta information for selected raster or vector layers, e.g. resolution, bounding box, data source, connected attribute table for vector data and a command history.

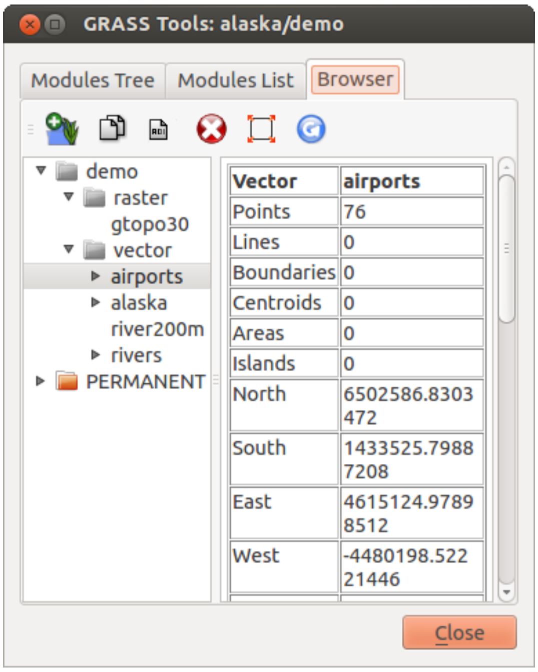


Figure 16.15: GRASS LOCATION browser

The toolbar inside the *Browser* tab offers following tools to manage the selected `LOCATION`:

- *Add selected map to canvas*
- *Copy selected map*
- *Rename selected map*
- *Delete selected map*
- *Set current region to selected map*
- *Refresh browser window*

The *Rename selected map* and *Delete selected map* only work with maps inside your currently selected `MAPSET`. All other tools also work with raster and vector layers in another `MAPSET`.

16.9.4 Customizing the GRASS Toolbox

Nearly all GRASS modules can be added to the GRASS toolbox. A XML interface is provided to parse the pretty simple XML files which configures the modules appearance and parameters inside the toolbox.

A sample XML file for generating the module `v.buffer` (v.buffer.qgm) looks like this:

```
<?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>
```

The parser reads this definition and creates a new tab inside the toolbox when you select the module. A more detailed description for adding new modules, changing the modules group, etc. can be found on the QGIS wiki at http://hub.qgis.org/projects/quantum-gis/wiki/Adding_New_Tools_to_the_GRASS_Toolbox

CHAPTER 17

QGIS processing framework

17.1 Introduction

This chapter introduces the QGIS processing framework, a geoprocessing environment that can be used to call native and third party algorithms from QGIS, making your spatial analysis tasks more productive and easy to accomplish.

In the following sections we will review how to use the graphical elements of this framework and take the most out of each one of them.

There are four basic elements in the framework GUI, which are used to run algorithms for different purposes. Choosing one tool or another will depend on the kind of analysis that is to be performed and the particular characteristics of each user and project. All of them (except for the batch processing interface, which is called from the toolbox, as we will see) can be accessed from the *Processing* menu item (you will see more than four entries. The remaining ones are not used to execute algorithms and will be explained later in this chapter).

- The toolbox. The main element of the GUI, it is used to execute a single algorithm or run a batch process based on that algorithm.

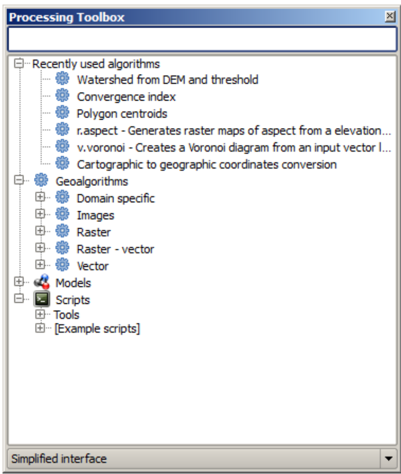


Figure 17.1: Processing Toolbox

- The graphical modeler. Several algorithms can be combined graphically using the modeler to define a workflow, creating a single process that involves several sub-processes
- The history manager. All actions performed using any of the aforementioned elements are stored in a history file and can be later easily reproduced using the history manager
- The batch processing interface. This interface allows you to execute batch processes and automate the execution of a single algorithm on multiple datasets.

Along the following sections we will review each one of this elements in detail.

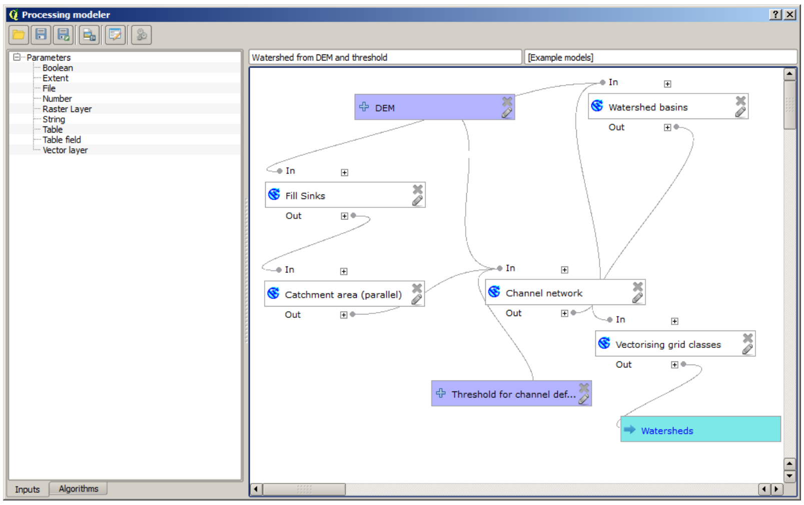


Figure 17.2: Processing Modeler

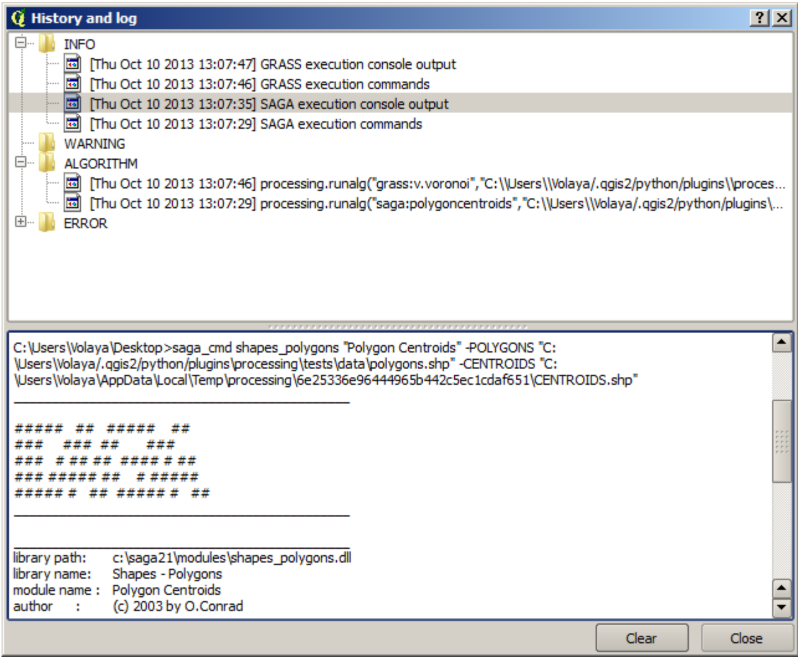


Figure 17.3: Processing History

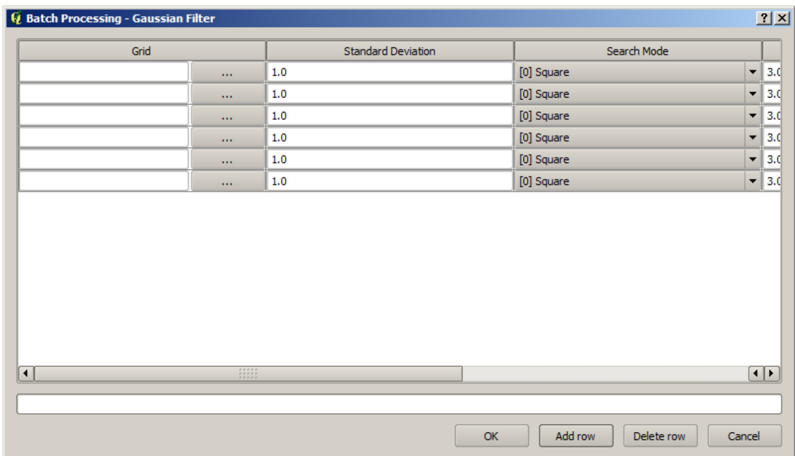


Figure 17.4: Batch Processing interface

17.2 The toolbox

The *Toolbox* is the main element of the processing GUI, and the one that you are more likely to use in your daily work. It shows the list of all available algorithms grouped in different blocks, and is the access point to run them whether as a single process or as a batch process involving several executions of a same algorithm on different sets of inputs.

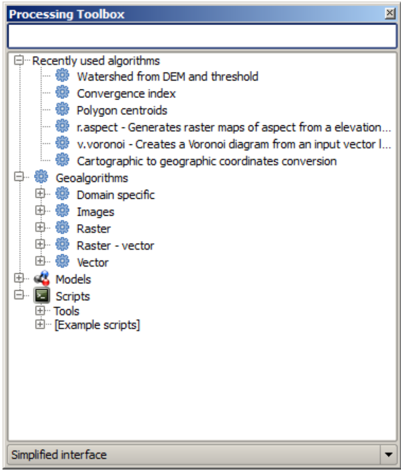


Figure 17.5: Processing Toolbox

The toolbox contains all the available algorithms, divided into predefined groups. All these groups are found under a single tree entry named *Geoalgorithms*.

Additionally, two more entries are found, namely *Models* and *Scripts*. These include user-created algorithms, and allow you to define your own workflows and processing tasks. We will devote a full section to them a bit later.

In the upper part of the toolbox you can find a text box. To reduce the number of algorithms shown in the toolbox and make it easier to find the one you need, you can enter any word or phrase on the text box. Notice that, as you type, the number of algorithms in the toolbox is reduced to just those which contain the text you have entered in their names.

In the lower part you will find a box that allows you to switch between the simplified algorithm list (the one explained above), and the advanced list. If you change to the advanced mode, the toolbox will look like this:

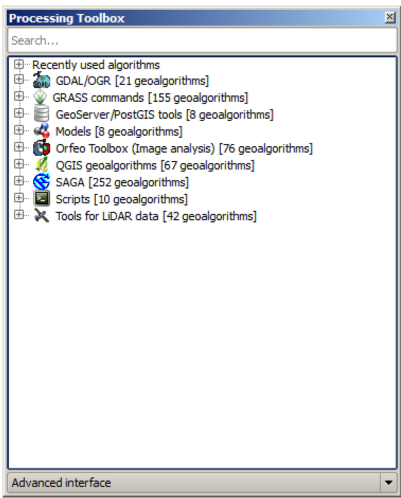


Figure 17.6: Processing Toolbox (advanced mode)

In the advanced view, each group represents a so-called 'algorithm provider', which is a set of algorithms coming from the same source, for instance, from a third-party application with geoprocessing capabilities. Some of this groups represent algorithms from one of such third-party applications (like SAGA, GRASS or R), while other contain algorithms directly coded as part of the processing plugin, not relying on any additional software.

This view is recommended to those users that have a certain knowledge of the applications that are backing those algorithms, since they will be shown with their original names and groups.

Also, some additional algorithms are available only in the advanced view, such as LiDAR tools or scripts based on the R statistical computing software, among others. Independent QGIS plugins that add new algorithms to the toolbox will only be shown in the advanced view.

In particular, the simplified view contains algorithms from the following providers:

- GRASS
- SAGA
- OTB
- Native QGIS algorithms

In the particular case of running QGIS under Windows, these algorithms are fully-functional in a fresh installation of QGIS and they can be run without requiring any additional installation. Also running them requires no prior knowledge of the external applications they use, making them more accesible for first-time users.

If you want to use an algorithm not provided by the any of above providers, switch to the advanced mode by selecting the corresponding option at the bottom of the toolbox.

To execute an algorithm, just double-click on its name in the toolbox.

17.2.1 The algorithm dialog

Once you double-click on the name of the algorithm that you want to execute, a dialog similar to the next one is shown (in this case, the dialog corresponds to the SAGA 'Convergence index' algorithm).

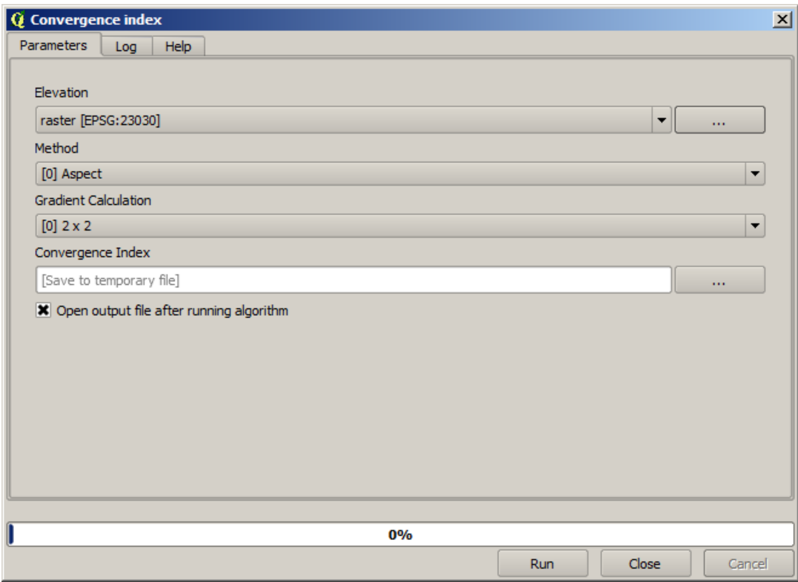


Figure 17.7: Parameters Dialog

This dialog is used to set the input values that the algorithm needs to be executed. It shows a table where input values and configuration parameters are to be set. It, of course, has a different content depending on the requirements of the algorithm to be executed, and is created automatically based on those requirements. On the left side, the name of the parameter is shown. On the right side the value of the parameter can be set.

Although the number and type of parameters depend on the characteristics of the algorithm, the structure is similar for all of them. The parameters found on the table can be of one of the following types.

- A raster layer, to select from a list of all the ones available (currently opened) in QGIS. The selector contains as well a button on its right-hand side, to let you select filenames that represent layers currently not loaded in QGIS.
- A vector layer, to select from a list of all the ones available in the QGIS. Layers not loaded in QGIS can be selected as well, as in the case of raster layers, but only if the algorithm does not require a table field selected from the attributes table of the layer. In that case, only opened layers can be selected, since they need to be open so as to retrieve the list of field names available.

  You will see a button by each vector layer selector, as shown in the figure below.

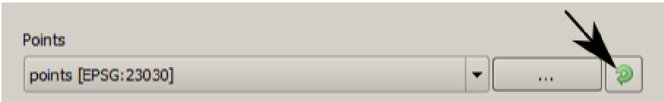


Figure 17.8: Vector iterator button

If the algorithm contains several of them, you will be able to toggle just one of them. If the button corresponding to a vector input is toggled, the algorithm will be executed iteratively on each one of its features instead of just once for the whole layer, producing as many outputs as times the algorithm is executed. This allows for automating the process when all features in a layer have to be processed separately.

- A table, to select from a list of all the ones available in QGIS. Non-spatial tables are loaded into QGIS like vector layers, and in fact they are treated as such by the program. Currently, the list of available tables that you will see when executing an algorithm that needs one of them is restricted to tables coming from files in DBase (`.dbf`) or Comma-Separated Values (`.csv`) formats.
- An option, to choose from a selection list of possible options.
- A numerical value, to be introduced in a text box. You will find a button by its side. Clicking on it you will see a dialog that allows you to enter a mathematical expression, so you can use it as a handy calculator. Some useful variables related to data loaded into QGIS can be added to your expression, so you can select a value derived from any of this variables such as the cellsize of a layer or the northern most coordinate of another one.

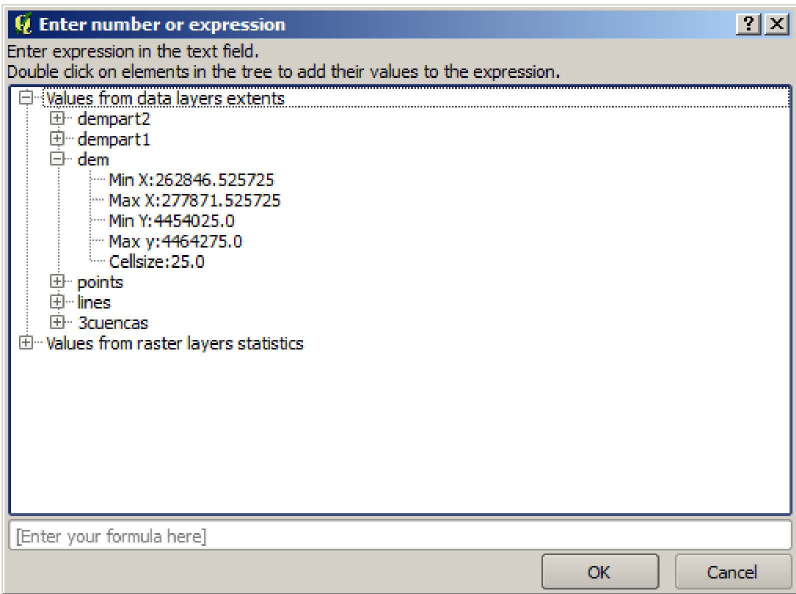


Figure 17.9: Number Selector

- A range, with min and max values to be introduced in two text boxes.

- A text string, to be introduced in a text box.
- A field, to choose from the attributes table of a vector layer or a single table selected in another parameter.
- A Coordinate Reference System. You can type the EPSG code directly in the text box, or select it from the CRS selection dialog that appear when you click on the button on the right-hand size
- A extent, to be entered by four number representing its `xmin`, `xmax`, `ymin`, `ymax` limits. Clicking on the button on the right-hand side of the value selector, a pop-up menu will appear, giving you two option: to select the value from a layer or the current canvas extent, or to define it by dragging directly onto the map canvas.

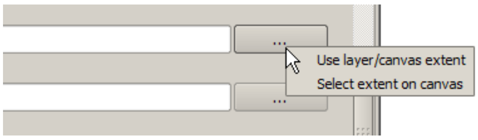


Figure 17.10: Extent selector

If you select the first option, you will see a window like the next one.

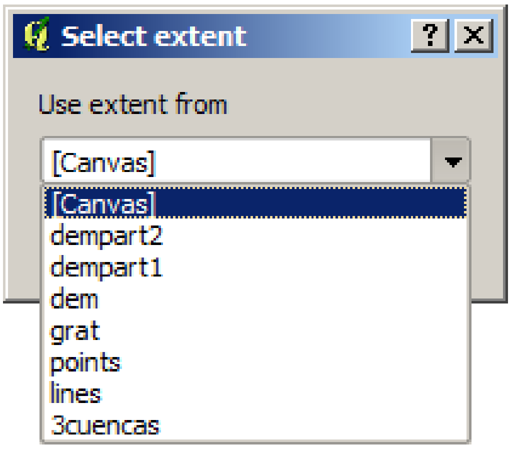


Figure 17.11: Extent List

If you select the second one, the parameters window will hide itself, so you can click and drag onto the canvas. Once you have defined the selected rectangle, the dialog will reappear, containing the values in the extent text box.

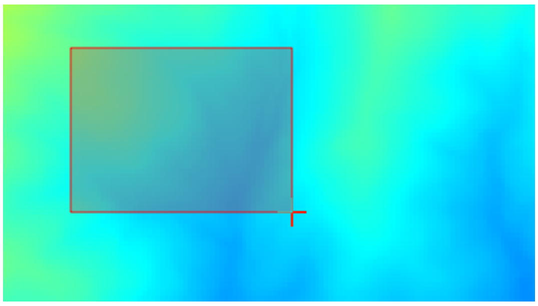


Figure 17.12: Extent Drag

- A list of elements (whether raster layers, vector ones or tables), to select from the list of the ones available in QGIS. To make the selection, click on the small button on the left side of the corresponding row to see a dialog like the following one.

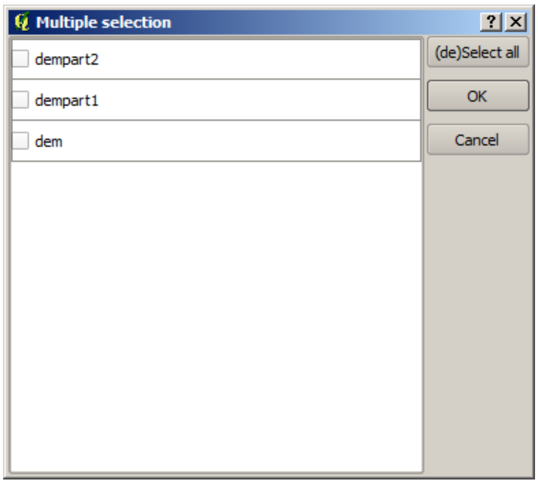


Figure 17.13: Multiple Selection

- A small table to be edited by the user. These are used to define parameters like lookup tables or convolution kernels, among others.

  Click on the button on the right side to see the table and edit its values.

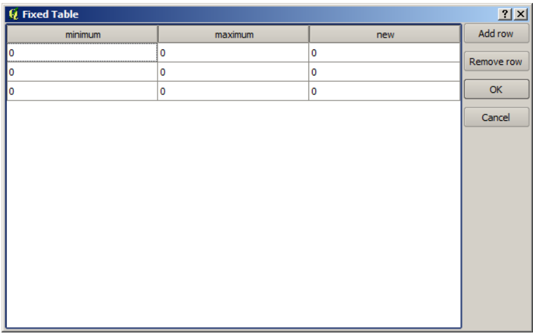


Figure 17.14: Fixed Table

  Depending on the algorithm, the number of rows can be modified or not, using the buttons on the right side of the window.

You will find a **[Help]** tab in the the parameters dialog. If a help file is available, it will be shown, giving you more information about the algorithms and detailed descriptions of what each parameter does. Unfortunately, most algorithms lack good documentation, but if you feel like contributing to the project, this would be a good place to start.

A note on projections

Algorithms run from the processing framework — and also most of the external applications whose algorithms are exposed through it — do not perform any reprojection on input layers and assumes that all of them are already in a common coordinate system and ready to be analized. Whenever you use more than one layer as input to an algorithm, whether vector or raster, it is up to you to make sure that they are all in the same coordinate system.

Note that, due to QGIS's on-the-fly reprojecting capabilities, although two layers might seem to overlap and match, that might not be true if their original coordinates are used without reprojecting them onto a common coordinate system. That reprojection should be done manually and then use the resulting files as input to the algorithm. Also note that the reprojection process can be performed with the algorithms that are available in the processing framework itself.

By default, the parameters dialog will show a description of the CRS of each layer along with its name, making it easy to select layers that share the same CRS to be used as input layers. If you do not want to see this additional information, you can disable this functionality in the processing config dialog, unchecking the *Show CRS* option.

If you try to execute an algorithm using as input two or more layers with unmatching CRS's, a warning dialog will be shown.

You still can execute the algorithm, but be aware that in most cases that will produce wrong results, such as empty layers due to input layers not overlapping.

17.2.2 Data objects generated by algorithms

Data objects generated by an algorithm can be of any of the following types:

- A raster layer
- A vector layer
- A table
- An HTML file (used for text and graphical outputs)

They are all saved to disk, and the parameters table will contain a text box corresponding to each one of these outputs, where you can type the output channel to use for saving it. An output channel contains the information needed to save the resulting object somewhere. In the most usual case, you will save it to a file, but the architecture allows for any other way of storing it. For instance, a vector layer can be stored in a database or even uploaded to a remote server using a WFS-T service. Although solutions like these are not yet implemented, the processing framework is prepared to handle them, and we expect to add new kinds of output channels in a near feature.

To select an output channel, just click on the button on the right side of the text box. That will open a save file dialog, where you can select the desired filepath. Supported file extensions are shown in the file format selector of the dialog, depending on the kind of output and the algorithm.

The format of the output is defined by the filename extension. The supported formats depend on the ones supported by the algorithm itself. To select a format, just select the corresponding file extension (or add it if you are directly typing the filepath instead). If the extension of the filepath you entered does not match any of the supported ones, a default extension (usually `.dbf`' for tables, `.tif` for raster layers and `.shp` for vector ones) will be appended to the filepath and the file format corresponding to that extension will be used to save the layer or table.

If you do not enter any filename, the result will be saved as a temporary file and in the corresponding default file format, and will be deleted once you exit QGIS (take care with that in case you save your project and it contains temporary layers).

You can set a default folder for output data objects. Go to the configuration dialog (you can open it from the *Processing* menu), and in the *General* group you will find a parameter named *Output folder*. This output folder is used as the default path in case you type just a filename with no path (i.e. `myfile.shp`) when executing an algorithm.

When running an algorithm that uses vector layer in iterative mode, the entered file path is used as the base path for all generated files, which are named using the base name and appending a number representing the index of the iteration. The file extension (and format) is used for all those generated files.

Apart from raster layers and tables, algorithms also generates graphics and texts as HTML files. These results are shown at the end of the algorithm execution in a new dialog. This dialog will keep the results produced by any algorithm during the current session, and can be shown at any time by selecting the *Processing → Results viewer* from QGIS main menu.

Some external applications might have files (with no particular extension restrictions) as output, but they do not belong to any of the categories above. Those output files will not be processed by QGIS (opened or included into the current QGIS project), since most of the times correspond to file formats or elements not supported by QGIS. This is, for instance, the case with LAS files used for LiDAR data. The files get created, but you won't see anything new in your QGIS working session.

For all the other types of outputs, you will find a check box that you can use to tell the algorithm whether to load the file once it is generated by the algorithm or not. By default, all files are opened.

Optional outputs are not supported, so all outputs are created, but you can uncheck the corresponding check box if you are not interested in a given output, which virtually makes it behave like an optional output (although the layer is created anyway, but if you leave the text box empty, it will be saved to a temporary file and deleted once you exit QGIS)

17.2.3 Configuring the processing framework

As it has been mentioned, the configuration menu gives access to a new dialog where you can configure how algorithms work. Configuration parameters are structured in separate blocks that you can select on the left-hand side of the dialog.

Along with the aforementioned *Output folder* entry, the *General* block contains parameters for setting the default rendering style for output layers (that is, layers generated by using algorithms from any of the framework GUI components). Just create the style you want using QGIS, save it to a file, and then enter the path to that file in the settings so the algorithms can use it. Whenever a layer is loaded by SEXTANTE and added to the QGIS canvas, it will be rendered with that style.

Rendering styles can be configured individually for each algorithm and each one of its outputs. Just right-click on the name of the algorithm in the toolbox and select *Edit rendering styles*. You will see a dialog like the one shown next.

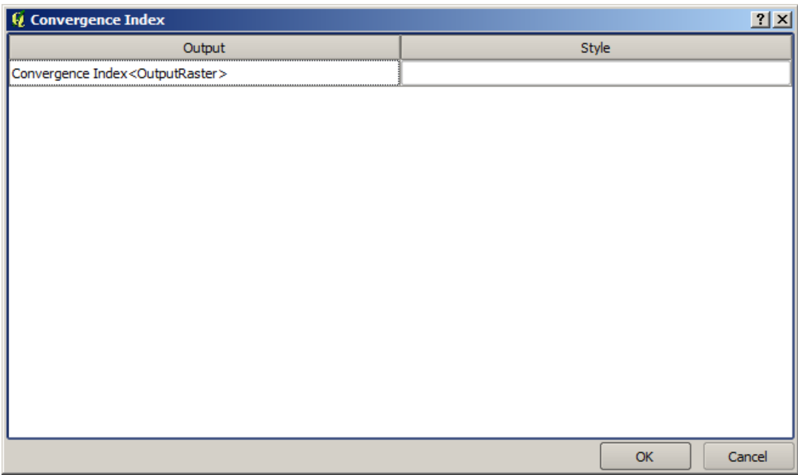


Figure 17.15: Rendering Styles

Select the style file (`.qml`) that you want for each output and press **[OK]**.

Other configuration parameters in the *General* group are the following ones:

- *Use filename as layer name*. The name of each resulting layer created by an algorithm is defined by the algorithm itself. In some cases, a fixed name might be used, that meaning that the same name will be used, no matter which input layer is used. In other cases, the name might depend on the name of the input layer or some of the parameters used to run the algorithm. If this checkbox is checked, the name will be taken from the output filename instead. Notice, that, if the output is saved to a temporary file, the filename of this temporary file is usually long and meaningless one intended to avoid collision with other already existing filenames.
- *Use only selected features*. If this option is selected, whenever a vector layer is used as input for an algorithm, only its selected features will be used. If the layer has no selected features, all of them will be used.
- *Pre-execution script file* and *Post-execution script file*. This parameters refer to scripts written using the processing scripting functionality, and are explained in the section covering scripting and the console.

Apart from the *General* block in the settings dialog, you will also find one for each algorithm provider. They contain an *Activate* item that you can use to make algorithms appear or not in the toolbox. Also, some algorithm providers have their own configuration items, that we will explain later when covering particular algorithm providers.

17.3 The graphical modeler

The *graphical modeler* allows to create complex models using a simple and easy-to-use interface. When working with a GIS, most analysis operations are not isolated, but part of a chain of operations instead. Using the graphical modeler, that chain of processes can be wrapped into a single process, so it is easier and more convenient to execute than a single process later on a different set on inputs. No matter how many steps and different algorithms it involves, a model is executed as a single algorithm, thus saving time and effort, specially for larger models.

The modeler can be opened from the processing menu.

The modeler has a working canvas where the structure of the model and the workflow it represents are shown. On the left part of the window, a panel with two tabs can be used to add new elements to the model.

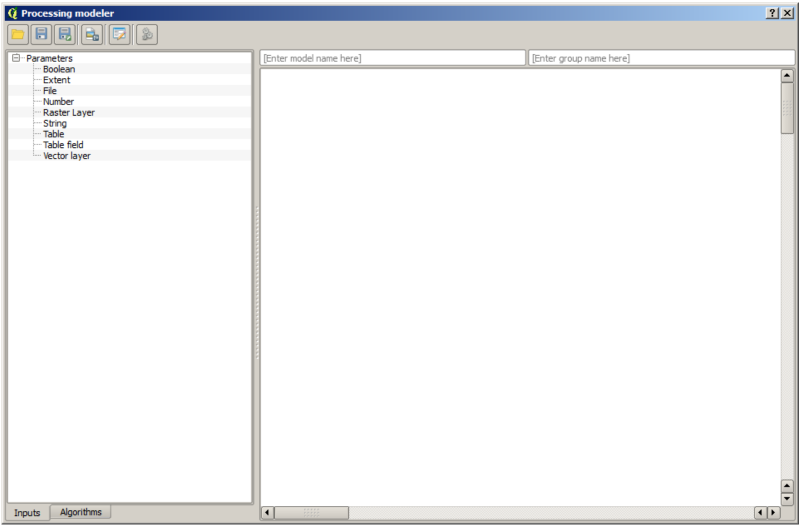


Figure 17.16: Modeler

Creating a model involves two steps:

1. *Definition of necessary inputs*. These inputs will be added to the parameters window, so the user can set their values when executing the model. The model itself is an algorithm, so the parameters window is generated automatically as it happens with all the algorithms available in the processign framework.
2. *Definition of the workflow*. Using the input data of the model, the workflow is defined adding algorithms and selecting how they use those inputs or the outputs generated by other algorithms already in the model

17.3.1 Definition of inputs

The first step to create a model is to define the inputs it needs. The following elements are found in the *Inputs* tab on the left side of the modeler window:

- Raster layer
- Vector layer
- String
- Table field
- Table
- Extent

- Number
- Boolean
- File

Double-clicking on any of them, a dialog is shown to define its characteristics. Depending on the parameter itself, the dialog will contain just one basic element (the description, which is what the user will see when executing the model) or more of them. For instance, when adding a numerical value, as it can be seen in the next figure, apart from the description of the parameter you have to set a default value and a range of valid values.

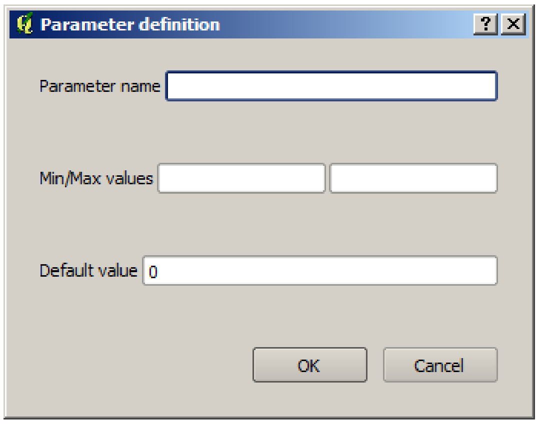


Figure 17.17: Model Parameters

For each added input, a new element is added to the modeler canvas.

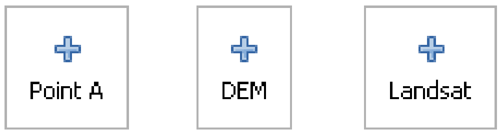


Figure 17.18: Model Parameters

17.3.2 Definition of the workflow

Once the inputs have been defined, it is time to define the algorithms to apply on them. Algorithms can be found in the *Algorithms* tab, grouped much in the same way as they are in the toolbox.

The appearance of the toolbox has two modes here as well: simplified and advanced. However, there is no element to switch between views in the modeler, and you have to do it in the toolbox. The mode that is selected in the toolbox is the one that will be used for the list of algorithms in the modeler.

To add an algorithm to a model, double-click on its name. 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 correspond to the SAGA 'Convergence index' algorithm, the same one we saw in the section dedicated to the toolbox.

As you can see, some differences exist. Instead of the file output box that was used to set the filepath for output layers and tables, a simple text box is. If the layer generated by the algorithm is just a temporary result that will be used as the input of another algorithm and should not be kept as a final result, just do not edit that text box. Typing anything on it means that the result is a final one, and the text that you supply will be the description for the output, which will be the one the user will see when executing the model.

Selecting the value of each parameter is also a bit different, since there are important differences between the context of the modeler and the toolbox one. Let's see how to introduce the values for each type of parameter.

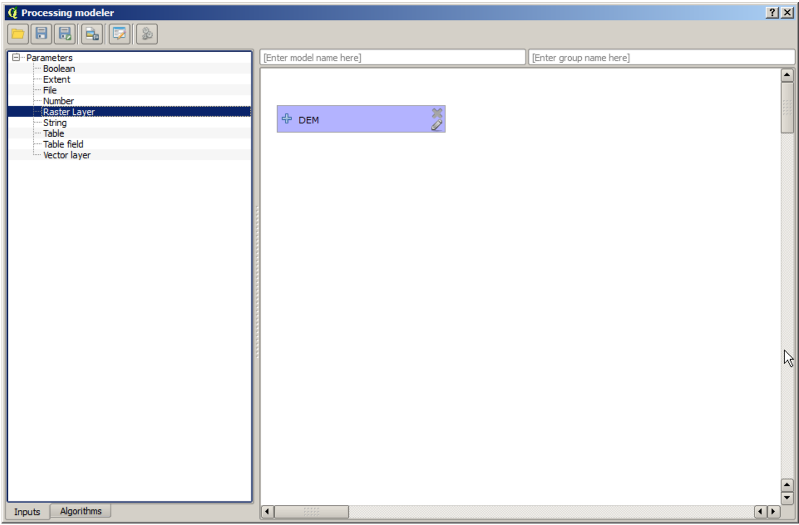


Figure 17.19: Model Parameters

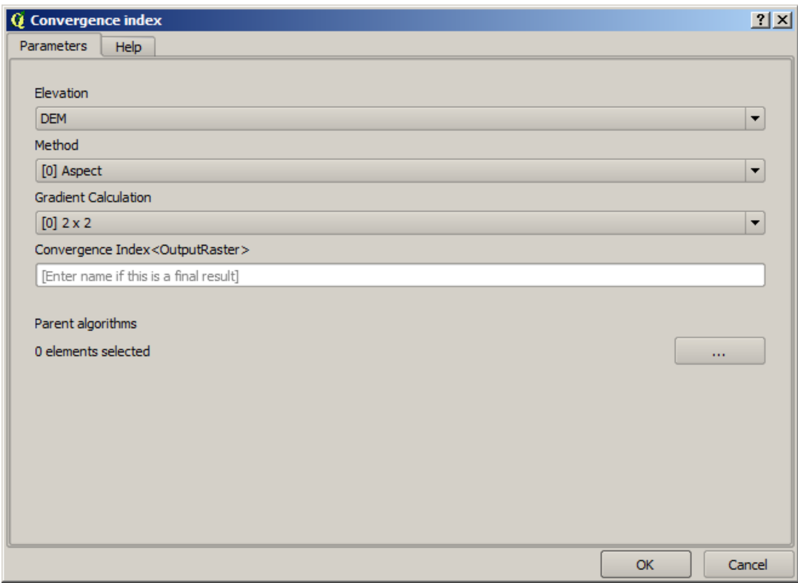


Figure 17.20: Model Parameters

- Layers (raster and vector) and tables. They are selected from a list, but in this case the possible values are not the layers or tables currently loaded in QGIS, but the list of model inputs of the corresponding type, or other layers or tables generated by algorithms already added to the model.
- Numerical values. Literal values can be introduced directly on the text box. But this text box is also a list that can be used to select any of the numerical value inputs of the model. In this case, the parameter will take the value introduced by the user when executing the model.
- String. Like in the case of numerical values, literal strings can be typed, or an input string can be selected.
- Table field. The fields of the parent table or layer cannot be known at design-time, since they depend of the selection of the user each time the model is executed. To set the value for this parameter, type the name of a field directly in the text box, or use the list to select a table field input already added to the model. The validity of the selected field will be checked at run-time.

In all cases, you will find an additional parameter named *Parent algorithms* that is not available when calling the algorithm from the toolbox. This parameter allows you to define the order in which algorithms are executed, by explicitly defining one algorithm as a parent of the current one, which will force it to be executed before it.

When you use the output of a previous algorithm as the input of your algorithm, that implicitly sets the former as parent of the current one (and places the corresponding arrow in the modeler canvas). However, in some cases an algorithm might depend on another one even if it does not use any output object from it (for instance, and algorithm that executes an SQL sentence on a PostGIS database and another one which imports a layer into that same database) In that case, just select it in the *Parent algorithms* parameter and they will be executed in the correct order.

Once all the parameter have been assigned valid values, click on **[OK]** and the algorithm will be added to the canvas. It will be linked to all the other elements in the canvas, whether algorithms or inputs, which provide objects that are used as inputs for that algorithm.

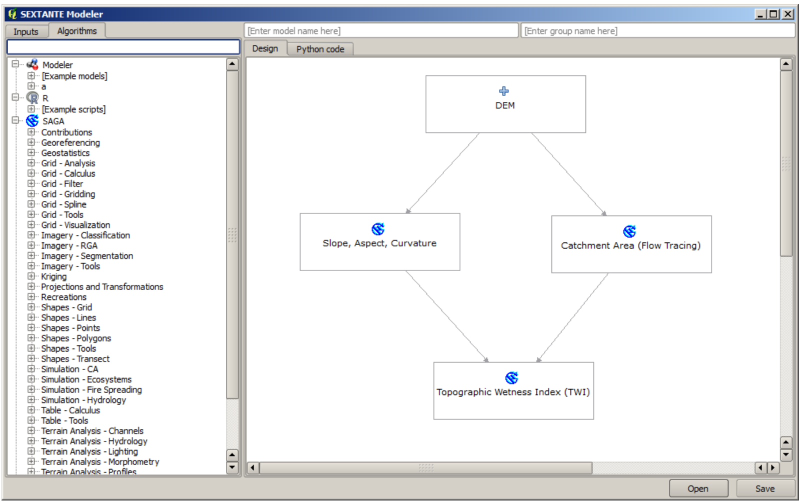


Figure 17.21: Model Parameters

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 update automatically.

You can run your algorithm anytime clicking on the **[Run]** button. However, in order to use it from the toolbox, it has to be saved and the modeler dialog closed, to allow the toolbox to refresh its contents.

17.3.3 Saving and loading models

Use the **[Save]** button to save the current model and the **[Open]** one to open any model previously saved. Model are saved with the `.model` extension. If the model has been previously saved from the modeler window, you will not be prompted for a filename, since there is already a file associated with that model, and it will be used.

Before saving a model, you have to enter a name and a group for it, using the text boxes in the upper part of the window.

Models saved on the `models` folder (the default folder when you are prompted for a filename to save the model) will appear in the toolbox in the corresponding branch. When the toolbox is invoked, it searches the `models` folder for files with `.model` extension and loads the models they contain. Since a model is itself an algorithm, it can be added to the toolbox just like any other algorithm.

The models folder can be set from the processing configuration dialog, under the *Modeler* group.

Models loaded from the `models` folder appear not only in the toolbox, but also in the algorithms tree in the *Algorithms* tab of the modeler window. That means that you can incorporate a model as a part of a bigger model, just as you add any other algorithm.

In some cases, a model might not be loaded because not all the algorithms included in its workflow are available. If you have used a given algorithm as part of your model, it should be available (that is, it should appear on the toolbox) in order to load that model. Deactivating an algorithm provider in the processing configuration window renders all the algorithms in that provider unusable by the modeler, which might cause problems when loading models. Keep that in mind when you have trouble loading or executing models.

17.3.4 Editing a model

You can edit the model you are currently creating, redefining the workflow and the relationships between the algorithms and inputs that define the model itself.

If you right-click on an algorithm in the canvas representing the model, you will see a context menu like the one shown next:

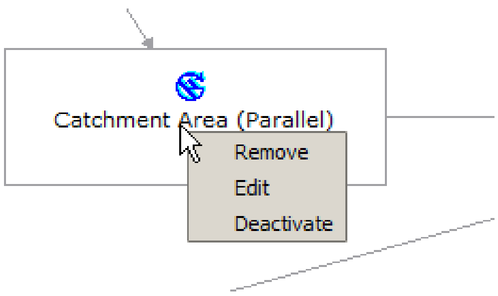


Figure 17.22: Modeler Right Click

Selecting the *Remove* option will cause the selected algorithm to be removed. An algorithm can be removed only if there are no other algorithms depending on it. That is, if no output from the algorithm is used in a different one as input. If you try to remove an algorithm that has others depending on it, a warning message like the one you can see below will be shown:

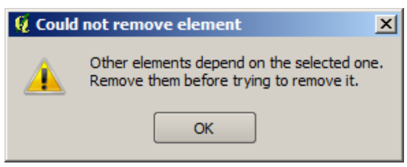


Figure 17.23: Cannot Delete Algorithm

Selecting the *Edit* option or simply double-clicking on the algorithm icon will show the parameters dialog of the algorithm, so you can change the inputs and parameter values. Not all input elements available in the model will appear in this case as available inputs. Layers or values generated at a more advanced step in the workflow defined by the model will not be available if they cause circular dependencies.

Select the new values and then click on the **[OK]** button as usual. The connections between the model elements will change accordingly in the modeler canvas.

17.3.5 Activating and deactivating algorithms

Algorithms can be deactivated in the modeler, so they will not be executed once the model is run. This can be used to test just a given part of the model, or when you do not need all the outputs it generates.

To deactivate an algorithm, right-click on its icon in the model canvas and select the *Deactivate* option. You will see that the algorithm is represented now with a red label under its name indicating that is not active.

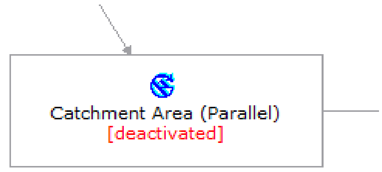


Figure 17.24: Deactivate

All algorithms depending (directly or undirectly) on that algorithm will also appear as inactive, since they cannot be executed now.

To activate an algorithm, just right–click on its icon and select the *Activate* option.

17.3.6 Editing model help files and meta-information

You can document your models from the modeler itself. Just click on the **[Edit model help]** button and a dialog like the one shown next will appear.

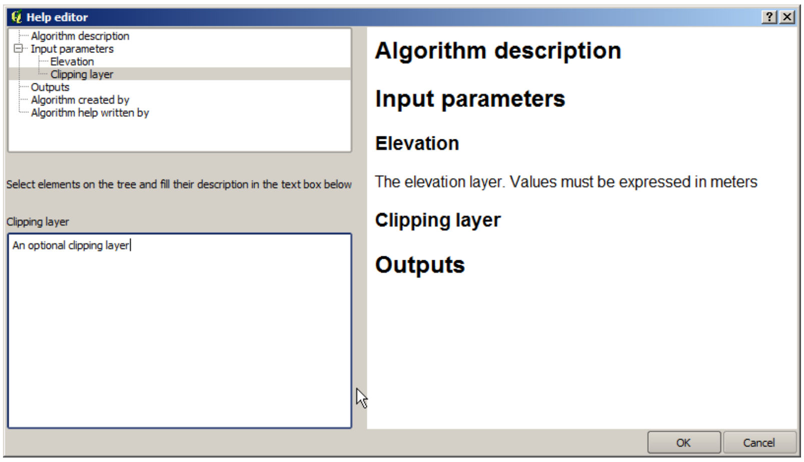


Figure 17.25: Help Edition

On the right-hand side you will see a simple HTML page, created using the description of the input parameters and outputs of the algorithm, along with some additional items like a general description of the model or its author. The first time you open the help editor all those descriptions are empty, but you can edit them using the elements on the left-hand side of the dialog. Select an element on the upper part and the write its description in the text box below.

Model help is saved in a file in the same folder as the model itself. You do not have to worry about saving it, since it is done automatically.

17.3.7 About available algorithms

You might notice that some algorithms that can be be executed from the toolbox do not appear in the list of available ones when you are designing a model. To be included in a model, and algorithm must have a correct semantic, so as to be properly linked to other in the workflow. If an algorithm does not have such well-defined semantic (for instance, if the number of output layers cannot be know in advance), then it is not possible to use it within a model, and thus does not appear in the list of them that you can find in the modeler dialog.

Additionally, you will see some algorithms in the modeler that are not found in the toolbox. This algorithms are meant to be used exclusively as part of a model, and they are of no interest in a different context. The 'Calculator' algorithm is an example of that. It is just a simple arithmetic calculator that you can use to modify numerical values (entered by the user or generated by some other algorithm). This tools is really useful within a model, but outside of that context, it doesn't make too much sense.

17.3.8 Saving models as Python code

Given a model, it is possible to automatically create Python code that performs the same task as the model itself. This code is used to create a console script (we will explain them later in this manual) and you can modify that script to incorporate actions and methods not available in the graphical modeler, such as loops or conditional sentences.

This feature is also a very practical way of learning how to use processign algorithms from the console and how to create new algorithms using Python code, so you can use it as a learning tool when you start creating your own scripts.

Save your model in the models folder and go to the toolbox, where it should appear now, ready to be run. Right click on the model name and select *Save as Python script* in the context menu that will pop-up. A dialog will prompt you to introduce the file where you want to save the script.

17.4 The batch processing interface

17.4.1 Introduction

All algorithms (including models) can be executed as a batch process. That is, they can be executed using not a single set of inputs, but several of them, executing the algorithm as many times as needed. This is useful when processing large amounts of data, since it is not necessary to launch the algorithm many times from the toolbox.

To execute an algorithm as a batch process, right-click on its name in the toolbox and select the *Execute as batch process* option in the pop-up menu that will appear.

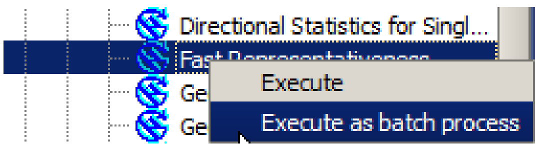


Figure 17.26: Batch Processing Right Click

17.4.2 The parameters table

Executing a batch process is similar to performing a single execution of an algorithm. Parameter values have to be defined, but in this case we need not just a single value for each parameter, but a set of them instead, one for each time the algorithm has to be executed. Values are introduced using a table like the one shown next.

Each line of this table represents a single execution of the algorithm, and each cell contains the value of one of the parameters. It is similar to the parameters dialog that you see when executing an algorithm from the toolbox, but with a different arrangement.

By default, the table contains just two rows. You can add or remove rows using the buttons on the lower part of the window.

Once the size of the table has been set, it has to be filled with the desired values.

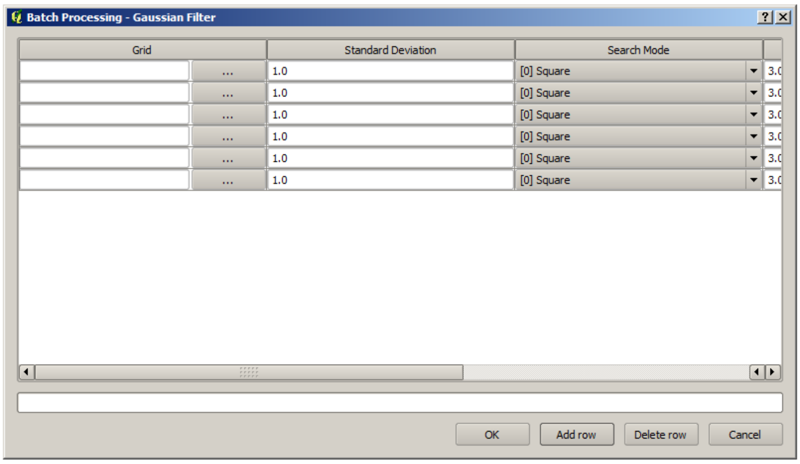


Figure 17.27: Batch Processing

17.4.3 Filling the parameters table

For most parameters, setting its value is trivial. Just type the value or select it from the list of available options, depending on the parameter type.

The main differences are found for parameters representing layers or tables, and for output filepaths. Regarding input layers and tables, when an algorithm is executed as part of a batch process those input data objects are taken directly from files, and not from the set of them already opened in QGIS. For this reason, any algorithm can be executed as a batch process even if no data objects at all are opened and the algorithm cannot be run from the toolbox.

Filenames for input data objects are introduced directly typing or, more conveniently, clicking on the ... button on the right hand of the cell, which shows a typical file chooser dialog. Multiple files can be selected at once. If the input parameter represents a single data object and several files are selected, each one of them will be put in a separate row, adding new ones if needed. If it represents a multiple input, all the selected files will be added to a single cell, separated by semicolons (`;`).

Output data objects are always saved to a file and, unlike when executing an algorithm from the toolbox, saving to a temporary one is not permitted. You can type the name directly or use the file chooser dialog that appears when clicking on the accompanying button.

Once you select the file, a new dialog is shown to allow for autocompletion of other cells in the same column (same parameter).

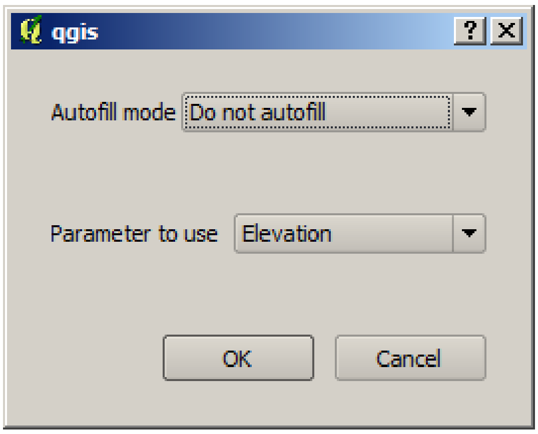
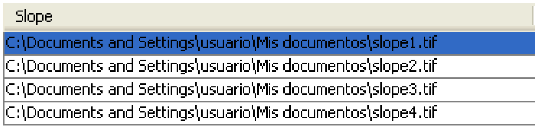


Figure 17.28: Batch Processing Save

If the default value ('Do not autocomplete') is selected, it will just put the selected filename in the selected cell from the parameters table. If any of the other options is selected, all the cells below the selected one will be automatically filled based on a defined criteria. This way, it is much easier to fill the table, and the batch process can be defined with less effort.

Automatic filling can be done simply adding correlative numbers to the selected filepath, or appending the value of another field at the same row. This is particularly useful for naming output data object according to input ones.



Slope
C:\Documents and Settings\usuario\Mis documentos\slope1.tif
C:\Documents and Settings\usuario\Mis documentos\slope2.tif
C:\Documents and Settings\usuario\Mis documentos\slope3.tif
C:\Documents and Settings\usuario\Mis documentos\slope4.tif

Figure 17.29: Batch Processing File Path

17.4.4 Executing the batch process

To execute the batch process once you have introduced all the necessary values, just click on **[OK]**. Progress of the global batch task will be shown in the progress bar in the lower part of the dialog.

17.5 Using processing algorithms from the console

The console allows advanced users to increase their productivity and perform complex operations that cannot be performed using any of the other GUI elements of the processing framework GUI. Models involving several algorithms can be defined using the command-line interface, and additional operations such as loops and conditional sentences can be added to create more flexible and powerful workflows.

There is not a proccesing console in QGIS, but all processing commands are available instead from the QGIS built-in Python console. That means that you can incorporate those command to your console work and connect processing algorithms to all the other features (including methods from the QGIS API) available from there.

The code that you can execute from the Python console, even if it does not call any specific processing method, can be converted into a new algorithm that you can later call from the toolbox, the graphical modeler or any other component, just like you do with any other algorithm. In fact, some algorithms that you can find in the toolbox are simple scripts.

In this chapter we will see how to use processing algorithms from the QGIS Python console, and also how to write your own algorithms using Python.

17.5.1 Calling algorithms from the Python console

The first thing you have to do is to import the processing functions with the following line:

```
>>> import processing
```

Now, there is basically just one (interesting) thing you can do with that from the console: to execute an algorithm. That is done using the `runalg()` method, which takes the name of the algorithm to execute as its first parameter, and then a variable number of additional parameter depending on the requirements of the algorithm. So the first thing you need to know is the name of the algorithm to execute. That is not the name you see in the toolbox, but rather a unique command–line name. To find the right name for your algorithm, you can use the `algslist()` method. Type the following line in you console:

```
>>> processing.alglist()
```

You will see something like this.

```
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 Thereshold 1--------------------->saga:averagewiththereshold1
Average With Thereshold 2--------------------->saga:averagewiththereshold2
Average With Thereshold 3--------------------->saga:averagewiththereshold3
B-Spline Approximation------------------------>saga:b-splineapproximation
...
```

That's a list of all the available algorithms, alphabetically ordered, along with their corresponding command-line names.

You can use a string as a parameter for this method. Instead of returning the full list of algorithm, it will only display those that include that string. If, for instance, you are looking for an algorithm to calculate slope from a DEM, type `alglist("slope")` to get the following result:

```
DTM Filter (slope-based)---------------------->saga:dtmfilter(slope-based)
Downslope Distance Gradient------------------->saga:downslopedistancegradient
Relative Heights and Slope Positions---------->saga:relativeheightsandslopepositions
Slope Length---------------------------------->saga:slopelength
Slope, Aspect, Curvature---------------------->saga:slopeaspectcurvature
Upslope Area---------------------------------->saga:upslopearea
Vegetation Index[slope based]----------------->saga:vegetationindex[slopebased]
```

This result might change depending on the algorithms you have available.

It is easier now to find the algorithm you are looking for and its command-line name, in this case `saga:slopeaspectcurvature`.

Once you know the command-line name of the algorithm, the next thing to do is to know the right syntax to execute it. That means knowing which parameters are needed and the order in which they have to be passed when calling the `runalg()` method. There is a method to describe an algorithm in detail, which can be used to get a list of the parameters that an algorithms require and the outputs that it will generate. To do it, you can use the `alghelp(name_of_the_algorithm)` method. Use the command-line name of the algorithm, not the full descriptive name.

Calling the method with `saga:slopeaspectcurvature` as parameter, you get the following description.

```
>>> processing.alghelp("saga:slopeaspectcurvature")
ALGORITHM: Slope, Aspect, Curvature
   ELEVATION <ParameterRaster>
   METHOD <ParameterSelection>
   SLOPE <OutputRaster>
   ASPECT <OutputRaster>
   CURV <OutputRaster>
   HCURV <OutputRaster>
   VCURV <OutputRaster>
```

Now you have everything you need to run any algorithm. As we have already mentioned, there is only one single command to execute algorithms: `runalg()`. Its syntax is as follows:

```
>>> processing.runalg(name_of_the_algorithm, param1, param2, ..., paramN,
         Output1, Output2, ..., OutputN)
```

The list of parameters and outputs to add depends on the algorithm you want to run, and is exactly the list that the `alghelp()` method gives you, in the same order as shown.

Depending on the type of parameter, values are introduced differently. The next one is a quick review of how to introduce values for each type of input parameter:

- Raster Layer, Vector Layer or Table. Simply use a string with the name that identifies the data object to use (the name it has in the QGIS Table of Contents) or a filename (if the corresponding layer is not opened, it will be opened, but not added to the map canvas). If you have an instance of a QGIS object representing the layer, you can also pass it as parameter. If the input is optional and you do not want to use any data object, use `None`.
- Selection. If an algorithm has a selection parameter, the value of that parameter should be entered using an integer value. To know the available options, you can use the `algoptions()` command, as shown in the following example:

  ```
  >>> processing.algoptions("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 this case, the algorithm has one of such such parameters, with 7 options. Notice that ordering is zero-based.
- Multiple input. The value is a string with input descriptors separated by semicolons (`;`). As in the case of single layers or tables, each input descriptor can be the data object name, or its filepath.
- Table Field from XXX. Use a string with the name of the field to use. This parameter is case-sensitive.
- Fixed Table. Type the list of all table values separated by commas (`,`) and enclosed between quotes (`"`). Values start on the upper row and go from left to right. You can also use a 2D array of values representing the table.
- CRS. Enter the EPSG code number of the desired CRS.
- Extent. You must use a string with `xmin`, `xmax`, `ymin` and `ymax` values separated by commas (`,`).

Boolean, file, string and numerical parameters do not need any additional explanations.

Input parameters such as strings booleans or numerical values have default values. To use them, use `None` in the corresponding parameter entry.

For output data objects, type the filepath to be used to save it, just as it is done from the toolbox. If you want to save the result to a temporary file, use `None`. The extension of the file determines the file format. If you enter a file extension not included in the ones supported by the algorithm, the default file format for that output type will be used, and its corresponding extension appended to the given filepath.

Unlike when an algorithm is executed from the toolbox, outputs are not added to the map canvas if you execute that same algorithm from the Python Console. If you want to add an output to it, you have to do it yourself after running the algorithm. To do so, you can use QGIS API commands, or, even easier, use one of the handy methods provided for such task.

The `runalg` method returns a dictionary with the output names (the ones shown in the algorithm description) as keys and the filepaths of those outputs as values. You can load those layers by passing its filepath to the `load()` method.

17.5.2 Additional functions for handling data

Apart from the functions used to call algorithms, importing the `processing` package will also import some additional functions that make it easier to work with data, particularly vector data. They are just convenience functions that wrap some functionality from the QGIS API, usually with a less complex syntax. These functions should be used when developing new algorithms, as they make it easier to operate with input data.

Below is a list of some of this commands. More information can be found in the classes under the `processing/tools` package, and aso in the example scripts provided with QGIS.

- `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)`: Returns the values in the attributes table of a vector layer, for the passed fields. Fields can be passed as field names or as zero-based field indices. Returns a dict of lists, with the passed field identifiers as keys. It considers the existing selection
- `getfeatures(layer)`: Returns an iterator over the features of a vector layer, considering the existing selection.
- `uniquelabels(layer, field)`: Returns a list of unique values for a given attribute. Attribute can be passed as a field name or a zero-based field index. It considers the existing selection

17.5.3 Creating scripts and running them from the toolbox

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
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, False, False, None, None, None, None, None)
processing.runalg("saga:topographicwetnessindex(twi), 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 sections in this 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`
- `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

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 it 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 object, 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`

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.

When you declare an output, 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
```

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.

If you 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.

Several examples are provided. 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.

17.5.4 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.

17.5.5 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.

17.6 The history manager

17.6.1 The processing history

Every time you execute an algorithm, information about the process is stored in the history manager. Along with the parameters used, the date and time of the execution are also saved.

This way, it is easy to track the and control all the work that has been developed using the processing framework, and easily reproduce it.

The history manager is a set of registry entries grouped according to their date of execution, making it easier to find information about an algorithm executed at any particular moment.

Process information is kept as a command-line expression, even if the algorithm was launched from the toolbox. This makes it also useful for those learning how to use the command-line interface, since they can call an algorithm using the toolbox and then check the history manager to see how that same algorithm could be called from the command line.

Apart from browsing the entries in the registry, processes can be re-executed, simply double-clicking on the corresponding entry.

Along with algorithm executions, the processing framework communicates with the user using the other groups of the registry, namely *Errors*, *Warnings* and *Information*. In case something is not working properly, having a look at the *Errors* might help you to see what is happening. If you get in contact with a developer to report a bug or error, the information in that group will be very useful for him to find out what is going wrong.

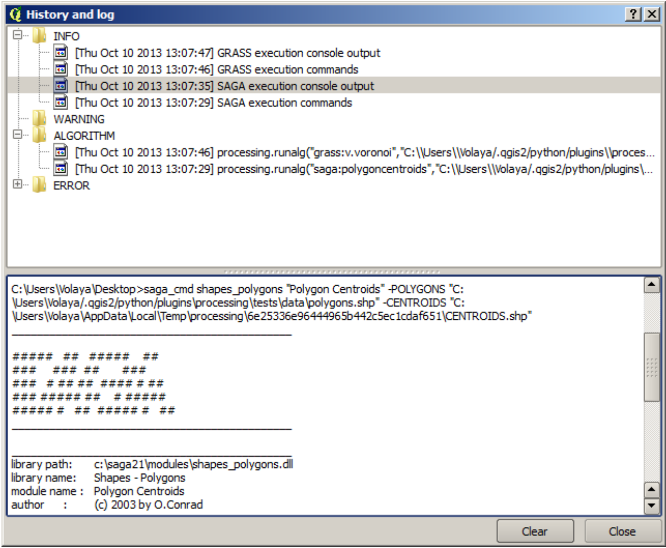


Figure 17.30: History

When executing third party algorithms, this is usually done calling their command-line interfaces, which communicate with the user using the console. Although that console is not shown, a full dump of it is stored in the *Information* group each time you run one of those algorithms. If, for instance, you are having problems executing a SAGA algorithm, look for an entry name 'SAGA execution console output' to check all the messages generated by SAGA and try to find out where the problem is.

Some algorithms, even if they can produce a result with the given input data, might add comments or additional information to *Warning* block in case they detect potential problems from that data, in order to warn you about them. Make sure you check those messages in case you are having unexpected results.

17.7 Configuring external applications

The processign framework can be extended using additional applications. Currently, SAGA, GRASS, OTB(Orfeo Toolbox) and R are supported, along with some other command-line applications that provide spatial data analysis functionalities. Algorithms relying on an external application are managed by their own algorithm provider.

This chapter will show you how to configure the processing framework to include these additional applications, and will explain some particular features of the algorithm based on them. Once you have correctly configured the system, you will be able to execute external algorithms from any component like the toolbox or the graphical modeler, just like you do with any other geoalgorithm.

By default, all algorithms that rely on an external appplication not shipped with QGIS are not enabled. You can enable them in the configuration dialog. Make sure that the corresponding application is already installed in your system. Enabling an algorithm provider without installing the application it needs will cause the algorithms to appear in the toolbox, but an error will be thrown when you try to execute them.

This is because the algorithm descriptions (needed to create the parameters dialog and provide the information needed about the algorithm) are not included with each application, but with QGIS instead. That is,they are part of QGIS, so you have them in your installation even if you have not installed any other software. Running the algorithm, however, needs the application binaries to be installed in your system.

17.7.1 A note for Windows users

If you are not an advanced user and you are running QGIS on windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the OSGeo4W application. That will automatically install SAGA, GRASS and OTB in your system, and configure them so they can be run from QGIS.

All the algorithms in the simplified view of the toolbox will be ready to be run, without needing any further configuration.

If you want to know more about how these providers work, or want to use some algorithms not included in the simplified toolbox (such as R scripts), keep on reading.

17.7.2 A note on file formats

When using an external software, opening a file in QGIS does not mean that it can be opened and processed as well on that other software. In most cases, it can read what you have opened in QGIS, but in some cases, that might not be the case. When using databases or uncommon file formats, whether for raster of vector layers, problems might arise. If that happens, try to use well known file formats that you are sure that are understood by both programs, and check to console output (in the history and log dialog) for knowing more about what is going wrong.

Using GRASS raster layers is, for instance, one case in which you might have trouble and not be able to complete your work if you call an external algorithm using such a layer as input. For this reason, these layers will not appear as available to algorithms.

You should, however, find no problems at all with vector layers, since QGIS automatically converts from the original file format to one accepted by the external application before passing the layer to it. This adds an extra processing time, which might be significant if the layer has a large size, so do not be surprised if it takes more to process a layer from a DB connection that one of a similar size stored in a shapefile.

Providers not using external applications can process any layer that you can open in QGIS, since they open it for analysis through QGIS.

Regarding output formats, all formats supported by QGIS as output can be used, both for raster and vector layers. Some provider do not support certain formats, but all can export to common formats raster layers that can be later transformed by QGIS automatically. As in the case of input layers, if this conversion is needed, that might increase the processing time.

If the extension of the filename specified when calling an algorithm does not match the extension of any of the formats supported by QGIS, then a suffix will be added to set a default format. In the case of raster layers, the `tif` extension is used, while `shp` is used for vector layer.

17.7.3 A note on vector layer selections

External applications are also aware of the selection that exist in vector layers within QGIS. However, that requires rewritting all input vector layers, just as if they were originally in a format not supported by the external application. Only when no selection exist, or the *Use only selected features* option is not enabled in the processing general configuration, a layer can be directly passed to an external application.

In other cases, exporting only selected features is needed, which causes execution times to be longer.

SAGA

SAGA algorithms can be run from QGIS if you have SAGA installed in your system and you configure the processing framework properly so it can find SAGA executables. In particular, the SAGA command-line executable is needed to run SAGA algorithms.

In case of running Windows, the standalone installer or the OSGeo4W installer, both install SAGA along with QGIS, and the path is automatically configured, so there is no need to do anything else.

If you have installed SAGA yourself (remember, you need version 2.1), the path to the SAGA executable must be configured. To do it, open the configuration dialog. In the *SAGA* block you will find a setting named *SAGA Folder*. Enter the path to the folder where SAGA is installed. Close the configuration dialog and now you are ready to run SAGA algorithms from QGIS.

In case you are running linux, SAGA binaries are not included with SEXTANTE, so you have to download and install the software yourself. Please check the SAGA website for more information. SAGA 2.1 is needed.

In this case there is no need to configure that, and you will not see those folders. Instead, you must make sure that SAGA is properly installed and its folder is added to the PATH environment variable. Just open a console and type `saga_cmd` to check that the system can find where SAGA binaries are located.

17.7.4 About SAGA grid system limitations

Most of SAGA algorithms that require several input raster layers, require them to have the same grid system. That is, to cover the same geographic area and have the same cellsize, so their corresponding grids match. When calling SAGA algorithms from QGIS, you can use any layer, regardless of its cellsize and extent. When multiple raster layers are used as input for a SAGA algorithm, QGIS resamples them to a common grid system and then passes them to SAGA (unless the SAGA algorithm can operate with layers from different grid systems).

The definition of that common grid system is controlled by the user, and you will find several parameters in the SAGA group of the setting window to do so. There are two ways of setting the target grid system:

- Setting it manually. You define the extent setting the values of the following parameters:
  - *Resampling min X*
  - *Resampling max X*
  - *Resampling min Y*
  - *Resampling max Y*
  - *Resampling cellsize*

  Notice that QGIS will resample input layers to that extent, even if they do not overlap with it.
- Setting it automatically from input layers. To select this option, just check the *Use min covering grid system for resampling* option. All the other settings will be ignored and the minimum extent that covers all the input layers will be used. The cellsize of the target layer is the maximum of all cellsizes of the input layers.

For algorithms that do not use multiple raster layers, or for those that do not need a unique input grid system, no resampling is performed before calling SAGA, and those parameters are not used.

17.7.5 Limitations for multi-band layers

Unlike QGIS, SAGA has no support for multi-band layers. If you want to use a multiband layer (such as an RGB or multispectral image), you first have to split it into single-banded images. To do so, you can use the 'SAGA/Grid - Tools/Split RGB image' algorithm (which creates 3 images from an RGB image) or the 'SAGA/Grid - Tools/Extract band' algorithm (to extract a single band).

17.7.6 Limitations in cellsize

SAGA assumes that raster layers have the same cellsize in the X and Y axis. If you are working with a layer with different values for its horizontal and vertical cellsizes, you might get unexcepted results. In this case, a warning will be added to the processing log, indicating that an input layer might not be suitable to be processed by SAGA.

17.7.7 Logging

When QGIS calls SAGA, it does it using its command-line interface, thus passing a set of commands to perform all the required operation. SAGA show its progress by writing information to the console, which includes the percentage of processing already done, along with additional content. This output is filtered and used to update the progress bar while the algorithm is running.

Both the commands sent by QGIS and the additional information printed by SAGA can be logged along with other processing log messages, and you might find them useful to track in detailed what is going on when QGIS runs a SAGA algorithm. you will find two settings, namely *Log console output* and *Log execution commands* to activate that logging mechanism.

Most other providers that use an external application and call it through the command-line have similar options, so you will find them as well in other places in the processing settings list.

R. Creating R scripts

R integration in QGIS is different from that of SAGA in that there is not a predefined set of algorithms you can run (except for a few examples). Instead, you should write your scripts and call R commands, much like you would do from R, and in a very similar manner to what we saw in the chapter dedicated to processing scripts. This chapter shows you the syntax to use to call those R commands from QGIS and how to use QGIS objects (layers, tables) in them.

The first thing you have to do, as we saw in the case of SAGA, is to tell QGIS where you R binaries are located. You can do so using the *R folder* entry in the processing configuration dialog. Once you have set that parameter, you can start creating your own R scripts and executing them.

Once again, this is different in Linux, and you just have to make sure that the R folder is included in the PATH environment variable. If you can start R just typing `R` in a console, then you are ready to go.

To add a new algorithm that calls an R function (or a more complex R script that you have developed and you would like to have available from QGIS), you have to create a script file that tells the processing framework how to perform that operation and the corresponding R commands to do so.

Script files have the extension `.rsx` and creating them is pretty easy if you just have a basic knowledge of R syntax and R scripting. They should be stored in the R-scripts folder. You can set this folder in the *R* settings group (available from the processing settings dialog), just like you do with the folder for regular processing scripts.

Let's have a look at a very simple file script file, which calls the R method `spsample` to create a random grid within the boundary of the polygons in a given polygon layer. This method belong to the `maptools` package. Since almost all the algorithms that you might like to incorporate into QGIS will use or generate spatial data, knowledge of spatial packages like `maptools` and, specially, `sp`, is mandatory.

```
##polyg=vector
##numpoints=number 10
##output=output vector
##sp=group
pts=spsample(polyg,numpoints,type="random")
output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

The first lines, which start with a double Python comment sign (`##`), tell QGIS the inputs of the algorithm described in the file and the outputs that it will generate. They work exactly with the same syntax as the SEXTANTE scripts that we have already seen, so they will not be described here again. Check the *processing_scripts* section for more information.

When you declare an input parameter, QGIS uses that information for two things: creating the user interface to ask the user for the value of that parameter and creating a corresponding R variable that can be later used as input for R commands.

In the above example, we are declaring an input of type `vector` named `polyg`. When executing the algorithm, QGIS will open in R the layer selected by the user and store it in a variable also named `polyg`. So the name of a parameter is also the name of the variable that we can use in R for accesing the value of that parameter (thus, you should avoid using reserved R words as parameter names).

Spatial elements such as vector and raster layers are read using the `readOGR()` and `brick()` commands (you do not have to worry about adding those commands to your description file, QGIS will do it) and stored as `Spatial*DataFrame` objects. Table fields are stored as strings containing the name of the selected field.

Tables are opened using the `read.csv()` command. If a table entered by the user is not in CSV format, it will be converted prior to importing it in R.

Additionally, raster files can be read using the `readGDAL()` command instead of `brick()`, by using the `##usereadgdal`.

If you are an advanced user and do not want QGIS to create the object representing the layer, you can use the ##passfilename tag to indicate that you prefer a string with the filename instead. In this case, it is up to you to open the file before performing any operation on the data it contains.

With the above information, we can now understand the first line of our first example script (the first line not starting with a Python comment).

```
pts=spsample(polyg,numpoints,type="random")
```

The variable polygon already contains a SpatialPolygonsDataFrame object, so it can be used to call the spsample method, just like the numpoints one, which indicates the number of points to add to the created sample grid.

Since we have declared an output of type vector named out, we have to create a variable named out and store a Spatial*DataFrame object in it (in this case, a SpatialPointsDataFrame). You can use any name for your intermediate variables. Just make sure that the variable storing your final result has the same name that you used to declare it, and contains a suitable value.

In this case, the result obtained from the spsample method has to be converted explicitly into a SpatialPointsDataFrame object, since it is itself an object of class ppp, which is not a suitable class to be returned to QGIS.

If your algorithm generates raster layers, the way they are saved will depend on whether you have used or not the #dontuserasterpackage option. In you have used it, layers are saved using the writeGDAL() method. If not, the writeRaster() method from the raster package will be used.

If you have used the #passfilename option, outputs are generated using the raster package (with writeRaster()), even though it is not used for the inputs.

If you algorithm does not generate any layer, but a text result in the console instead, you have to indicate that you want the console to be shown once the execution is finished. To do so, just start the command lines that produce the results you want to print with the > ('greater') sign. The output of all other lines will not be shown. For instance, here is the description file of an algorithm that performs a normality test on a given field (column) of the attributes of a vector layer:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

The output ot the last line is printed, but the output of the first is not (and neither are the outputs from other command lines added automatically by QGIS).

If your algorithm creates any kind of graphics (using the plot() method), add the following line:

```
##showplots
```

This will cause QGIS to redirect all R graphical outputs to a temporary file, which will be later opened once R execution has finished.

Both graphics and console results will be shown in the processing results manager.

For more information, please check the script files provided with SEXTANTE. Most of them are rather simple and will greatly help you understand how to create your own ones.

Note: rgdal and maptools libraries are loaded by default so you do not have to add the corresponding library() commands (you have to make sure, however, that those two packages are installed in your R distribution). However, other additional libraries that you might need have to be explicitly loaded. Just add the necessary commands at the beginning of your script. You also have to make sure that the corresponding packages are installed in the R distribution used by QGIS. The processing framework will not take care of any package installation. If you run a script that requires an uninstalled package, the execution will fail, and SEXTANTE will try to detect which packages are missing. You must install those missing libraries manually before you can run the algorithm.

GRASS

Configuring GRASS is not much different from configuring SAGA. First, the path to the GRASS folder has to be defined, but only if you are running Windows. Additionaly, a shell interpreter (usually `msys.exe`, which can be found in most GRASS for Windows distributions) has to be defined and its path set up as well.

By default, the processign framework tries to configure its GRASS connector to use the GRASS distribution that ships along with QGIS. This should work without problems in most systems, but if you experience problems, you might have to do it manually. Also, if you want to use a different GRASS installation, you can change that setting and point to the folder where that it is installed. GRASS 6.4 is needed for algorithms to work correctly.

If you are running Linux, you just have to make sure that GRASS is correctly installed, and that it can be run without problem from a console.

GRASS algorithms use a region for calculations. This region can be defined manually using values similar to the ones found in the SAGA configuration, or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time. If this is the behaviour you prefer, just check the *Use min covering region* option in the GRASS configuration parameters.

The last parameter that has to be configured is related to the mapset. A mapset is needed to run GRASS, and the processing frmaework creates a temporary one for each execution. You have to specify if the data you are working with uses geographical (lat/lon) coordinates or projected ones.

GDAL

No additional configuration is needed to run GDAL algorithms, since it is already incorporated to QGIS and algorithms can infere its configuration from it.

Orfeo ToolBox

Orfeo ToolBox (OTB) algorithms can be run from QGIS if you have OTB installed in your system and you have configured QGIS properly, so it can find all necessary files (command-line tools and libraries).

As in the case of SAGA OTB binaries are included in the standalone installer for Windows, but are not included if you are runing Linux, so you have to download and install the software yourself. Please check the OTB website for more information.

Once OTB is installed, start QGIS, open the processing configuration dialog and configure the OTB algorithm provider. In the *Orfeo Toolbox (image analysis)* block you will find all settings related to OTB. First ensure that algorithms are enabled.

Then configure the path to the folder where OTB command–line tools and libraries are installed:

- usually *OTB applications folder* point to `/usr/lib/otb/applications` and *OTB command line tools folder* is `/usr/bin`
- if you use OSGeo4W installer, than install `otb-bin` package and enter `C:\OSGeo4W\apps\orfeotoolbox\applications` as *OTB applications folder* and `C:\OSGeo4W\bin` as *OTB command line tools folder*. This values should be configured by default, but if you have a different OTB installation, configure them to the correspondig values in your system.

TauDEM

To use this provider you need to install TauDEM command line tools.

17.7.8 Windows

Please visit TauDEM homepage for installation instructions and precompiled binaries for 32bit and 64bit systems. **IMPORTANT**: you need TauDEM 5.0.6 executables, version 5.2 is currently not supported.

17.7.9 Linux

There are no packages for most Linux distribution, so you should compile TauDEM by yourself. As TauDEM uses MPICH2, first install it using your favorite package manager. Also TauDEM works fine with OpenMPI, so you can use it instead of MPICH2.

Download TauDEM 5.0.6 source code and extract files in some folder.

Open `linearpart.h` file and add after line

```
#include "mpi.h"
```

add new line with

```
#include <stdint.h>
```

so you'll get

```
#include "mpi.h"
#include <stdlib.h>
```

Save changes and close file. Now open `tiffIO.h`, find line `#include "stdint.h"` and replace quotes (`""`) with <>, so you'll get

```
#include <stdint.h>
```

Save changes and close file. Create build directory and cd into it

```
mkdir build
cd build
```

Configure your build with command

```
CXX=mpicxx cmake -DCMAKE_INSTALL_PREFIX=/usr/local ..
```

and then compile

```
make
```

Finaly, to install TauDEM into `/usr/local/bin`, run

```
sudo make install
```

17.8 The SEXTANTE Commander

SEXTANTE includes a practical tool that allows you to run algorithms without having to use the toolbox, but just typing the name of the algorithm you want to run.

This tools is known as the *SEXTANTE 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, pressing `Shift + Ctrl + M` (you can change that default keyboard shortcut in the QGIS configuration if you prefer a different one). Apart from executing SEXTANTE 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 reducing the usage of buttons and menus.

Moreover, the commander is configurable and you can add your custom commands and have them just a few keystrokes away, making it a powerful tool to become more productive in your daily work with QGIS

17.8.1 Available commands

The commands available in the commander fall in the following categories:

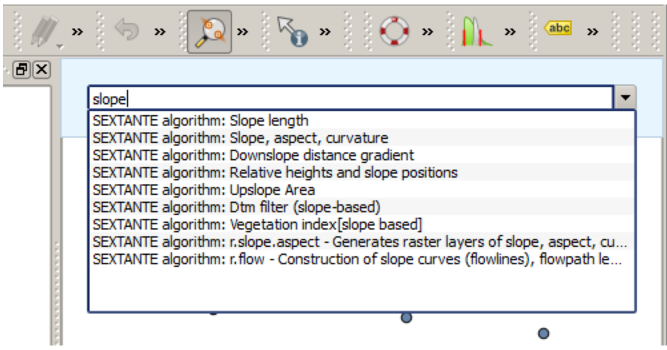


Figure 17.31: The SEXTANTE Commander

- SEXTANTE algorithms. They are shown as `SEXTANTE algorithm: <name of the algorithm>`.
- Menu items. They are shown as `Menu item: <menu entry text>`. All menus items available from the QGIS interface are available, even if they are included in a submenu.
- Python functions. You can create short Python functions that will be then included in the list of available commands. They are shown as `Function: <function name>`

To run any of the above, just start typing and then select the corresponding element from the list of available ones that appears after filtering the whole list of commands with the text you have entered.

In the case of calling a Python function, you can select the entry in the list, which is prefixed by `Function:` (for instance, `Function: removeall), or just directly type the function name (``removeall` in the previous example). There is no need to add brackets after the function name.

17.8.2 Creating custom functions

Custom functions are added by entering the corresponding Python code in the `commands.py` file that is found in the `.qgis/sextante/commander directory` in your user folder. It is just a simple Python file where you can add the functions that you need.

The file is created the first time you open the commander, with a few example functions. If you haven't launched the commander yet, you can create the file yourself. To edit the commands file, use your favorite text editor. You can also use a built-in editor by calling the `edit` command from the commander. It will open the editor with the commands file, and you can edit it directly and then save your changes.

For instance, you can add the following function, which removes all layers:

```
from qgis.gui import *

def removeall():
    mapreg = QgsMapLayerRegistry.instance()
    mapreg.removeAllMapLayers()
```

Once you have added the function, it will be available in the commander, and you can invoke it by typing `removeall`. There is no need to do anything apart from writing the function itself.

Functions can receive parameters. Add `*args` to your function definition, to receive argument. When calling the function from the commander, parameters have to be passed separated by spaces.

Here is an example of a function that loads a layer and takes a parameter with the filename of the layer to load.

```
import sextante

def load(*args):
  sextante.load(args[0])
```

If you want to load a layer in `/home/myuser/points.shp`, type `load /home/myuser/points.shp` in the commander text box.

Print Composer

The print composer provides growing layout and printing capabilities. It allows you to add elements such as the QGIS map canvas, text labels, images, legends, scalebars, basic shapes, arrows, attribute tables and HTML frames. You can size, group, align and position each element and adjust the properties to create your layout. The layout can be printed or exported to image formats, Postscript, PDF or to SVG (export to SVG is not working properly with some recent Qt4 versions, you should try and check individual on your system). You can save the layout as template and load it again in another session. Finally, generating several maps based on a template can be done throught the Atlas generator See a list of tools in table_composer_1:

Icon	Purpose	Icon	Purpose
	Save Project		New Composer
	Duplicate Composer		Composer Manager
	Load from template		Save as template
	Export to an image format		Export as PDF
	Export print composition to SVG		Print or export as Postscript
	Zoom to full extent		Zoom in
	Zoom out		Refresh view
	Revert last change		Restore last change
	Add new map from QGIS map canvas		Add image to print composition
	Add label to print composition		Add new legend to print composition
	Add new scalebar to print composition		Add basic shape to print composition
	Add arrow to print composition		Add attribute table to print composition
	Add a HTML Frame		
	Select/Move item in print composition		Move content within an item
	Group items of print composition		Ungroup items of print composition
	Raise selected items		Lower selected items
	Move selected items to top		Move selected items to bottom
	Align selected items left		Align selected items right
	Align selected items center		Align selected items center vertical
	Align selected items top		Align selected items bottom

Table Composer 1: Print Composer Tools

All Print Composer tools are available in menus and as icons in a toolbar. The toolbar can be switched off and on using the right mouse button over the toolbar.

18.1 First steps

18.1.1 Open a new Print Composer Template

Before you start to work with the print composer, you need to load some raster and vector layers in the QGIS map canvas and adapt their properties to suit your own convenience. After everything is rendered and symbolized to your liking, click the New Print Composer icon in the toolbar or choose *File* → *New Print Composer*. You will be prompt to choose a title for the new composer.

18.1.2 Using Print Composer

Opening the print composer provides you with a blank canvas to which you can add the current QGIS map canvas, text labels, images, legends, scalebars, basic shapes, arrows, attribute tables and HTML frames. Figure_composer_1 shows the initial view of the print composer with an activated *Snap to grid* mode but before any elements are added.

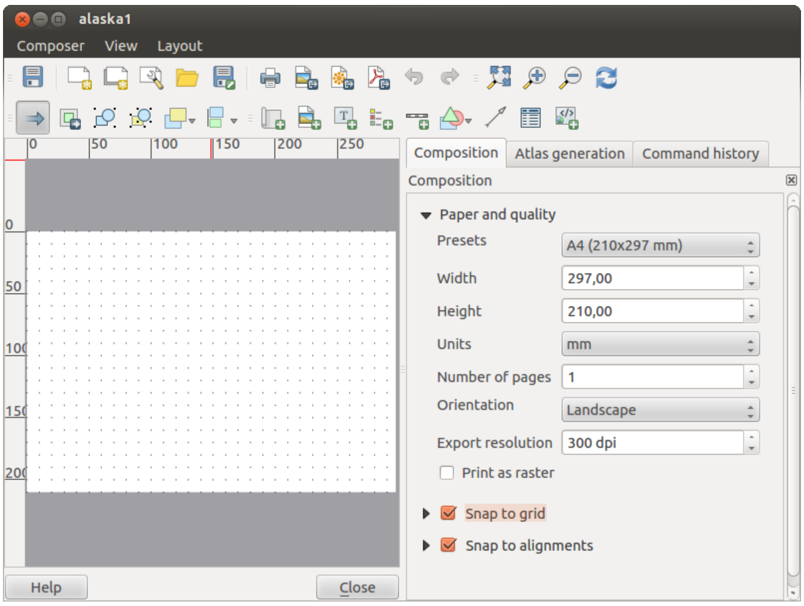


Figure 18.1: Print Composer

The print composer provides four tabs:

- The *Composition* tab allows you to set paper size, orientation, the print quality for the output file in dpi and to activate snapping to a grid of a defined resolution. You can also choose the *Number of pages* your composition will have. Please note, the *Snap to grid* feature only works, if you define a grid resolution > 0. Furthermore you can also activate the *Print as raster* checkbox. This means all elements will be rastered before printing or saving as Postscript or PDF.
- The *Item Properties* tab displays the properties for the selected item element. Click the Select/Move item icon to select an element (e.g. legend, scalebar or label) on the canvas. Then click the *Item Properties* tab and customize the settings for the selected element.
- The *Command history* tab (hidden by default) displays a history of all changes applied to the print composer layout. With a mouse click it is possible to undo and redo layout steps back and forth to a certain status.

- The *Atlas generation* tab allows to enable the generation of an atlas for the current composer and gives access to its parameters.

You can add multiple elements to the composer. It is also possible to have more than one map view or legend or scalebar in the print composer canvas, on one or several pages. Each element has its own properties and in the case of the map, its own extent. If you want to remove any elements from the composer canvas you can do that with the `Delete` or the `Backspace` key.

Navigation tools

To navigate in the canvas layout, the print composer provides 4 general tools:

- Zoom in
- Zoom out
- Zoom to full extend
- Refresh the view (if you find the view in an inconsistent state)

18.1.3 Composition tab — General composition setup

In the *Composition* tab, you can define the global settings of your composition.

- You can choose one of the *Presets* for your papersheet, or enter your custom *width* and *height*.
- Composition can now be parted on several pages. For instance, a first page can show a map canvas and a second page will show the attribute table associated to a layer while a third ons shows a HTML frame linking to your organization website. Set the *Number of pages* to the desired value.
- Choose the page *Orientation* and its *Exported resolution*
- When checked, the *print as raster* means all elements will be rasterized before printing or saving as Postscript or PDF.
- *Snap to grid* and *Snap to alignements* tools make accomplishing some tasks much easier. There's three types of grid: **Dots**, **Solid** lines and **Crosses**. You can adjust *spacings*, *offsets* and *color* to your need.
- *Selection tolerance* defines the maximum distance below which an item is snapped to the grid.
- *Snap to alignements* shows helping lines when the borders or axis of two items are aligned.

18.1.4 Composer items general options

Composer items have a set of common properties you will find on the bottom of the *Item Properties* tab: Position and size, Frame, Background, Item ID and Rendering (See figure_composer_2)

- The *Position and size* dialog lets you define size and position of the frame that contains the item. You can also choose which *Reference point* will be set at the **X** and **Y** coordinates previously defined.
- The *Frame* shows or hides the frame around the label. Click on the **[Color]** and **[Thickness]** buttons to adjust those properties.
- the *Background* enables or disables a background color. Click on the **[Color...]** button to display a dialog where you pick a color ou choose frome a custom setting. Transparency can also be adjusted throught the **alpha** field.
- Use the *Item ID* to create a relationship to other print composer items.
- *Rendering* mode can be selected in the option field. See Rendering_Mode .

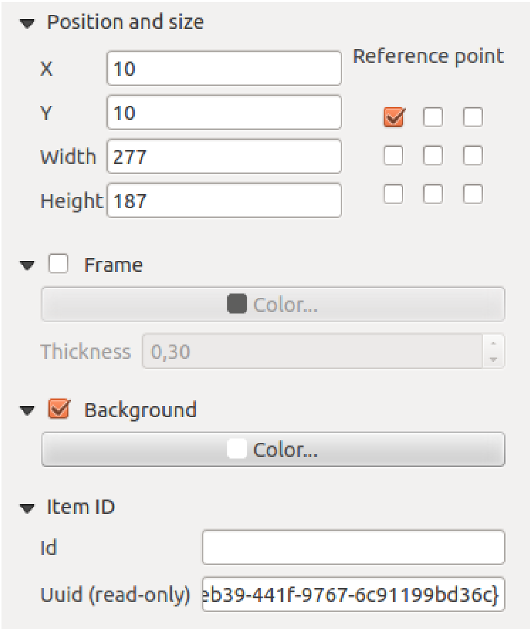


Figure 18.2: Common Item properties Dialogs

18.2 Rendering mode

QGIS now allows advanced rendering for composer items just like vector and raster layers.

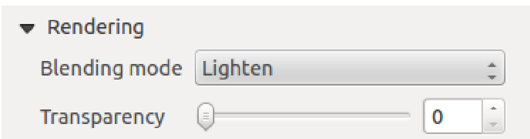


Figure 18.3: Rendering mode

- *Transparency* : you can make the underlying item in the composer visible with this tool. Use the slider to adapt the visibility of your item to your needs. You can also make a precise definition of the percentage of visibility in the the menu beside the slider.
- *Blending mode*: you can achieve special rendering effects with these tools that you previously only know from graphics programs. The pixels of your overlaying and underlaying items are mixed through the settings described below.
  - Normal: This is the standard blend mode which uses the alpha channel of the top pixel to blend with the Pixel beneath it; the colors aren't mixed
  - Lighten: It selects the maximum of each component from the foreground and background pixels. Be aware that the results tend to be jagged and harsh.
  - Screen: Light pixels from the source are painted over the destination, while dark pixels are not. This mode is most useful for mixing the texture of one layer with another layer. E.g. you can use a hillshade to texture another layer
  - Dodge: Dodge will brighten and saturate underlying pixels based on the lightness of the top pixel. So brighter top pixels cause the saturation and brightness of the underlying pixels to increase. This works best if the top pixels aren't too bright, otherwise the effect is too extreme.
  - Addition: This blend mode simply adds pixel values of one layer with the other. In case of values above 1 (in the case of RGB), white is displayed. This mode is suitable for highlighting features.

- Darken: Creates a resultant pixel that retains the smallest components of the foreground and background pixels. Like lighten, the results tend to be jagged and harsh
- Multiply: It multiplies the numbers for each pixel of the top layer with the corresponding pixel for the bottom layer. The results are darker pictures.
- Burn: Darker colors in the top layer causes the underlying layers to darken. Can be used to tweak and colorise underlying layers.
- Overlay: Combines multiply and screen blending modes. In the resulting picture light parts of the picture become lighter and dark parts become darker.
- Soft light: Very similar to overlay, but instead of using multiply/screen it uses color burn/dodge. This one is supposed to emulate shining a soft light onto an image.
- Hard light: Hard light is very similar to the overlay mode. It's supposed to emulate projecting a very intense light onto an image.
- Difference: Difference subtracts the top pixel from the bottom pixel or the other way round, to always get a positive value. Blending with black produces no change, as values for all colors are 0.
- Subtract: This blend mode simply subtracts pixel values of one layer with the other. In case of negative values, black is displayed.

18.3 Composer Items

18.3.1 Adding a current QGIS map canvas to the Print Composer

Click on the Add new map toolbar button in the print composer toolbar to add the QGIS map canvas. Now drag a rectangle on the composer canvas with the left mouse button to add the map. To display the current map, you can choose between three different modes in the map *Item Properties* tab:

- **Rectangle** is the default setting. It only displays an empty box with a message 'Map will be printed here'.
- **Cache** renders the map in the current screen resolution. If case you zoom in or out the composer window, the map is not rendered again but the image will be scaled.
- **Render** means, that if you zoom in or out the composer window, the map will be rendered again, but for space reasons, only up to a maximum resolution.

Cache is default preview mode for newly added print composer maps.

You can resize the map element by clicking on the Select/Move item button, selecting the element, and dragging one of the blue handles in the corner of the map. With the map selected, you can now adapt more properties in the map *Item Properties* tab.

To move layers within the map element select the map element, click the Move item content icon and move the layers within the map element frame with the left mouse button. After you found the right place for an element, you can lock the element position within the print composer canvas. Select the map element and click on the right mouse button to Lock the element position and again to unlock the element. You can lock the map element also activating the *Lock layers for map item* checkbox in the *Map* dialog of the *Item Properties* tab.

Main properties

The *Main properties* dialog of the map *Item Properies* tab provides following functionalities (see figure_composer_4):

- The **Preview** area allows to define the preview modes 'Rectangle', 'Cache' and 'Render', as described above. If you change the view on the QGIS map canvas by zooming or panning or changing vector or raster

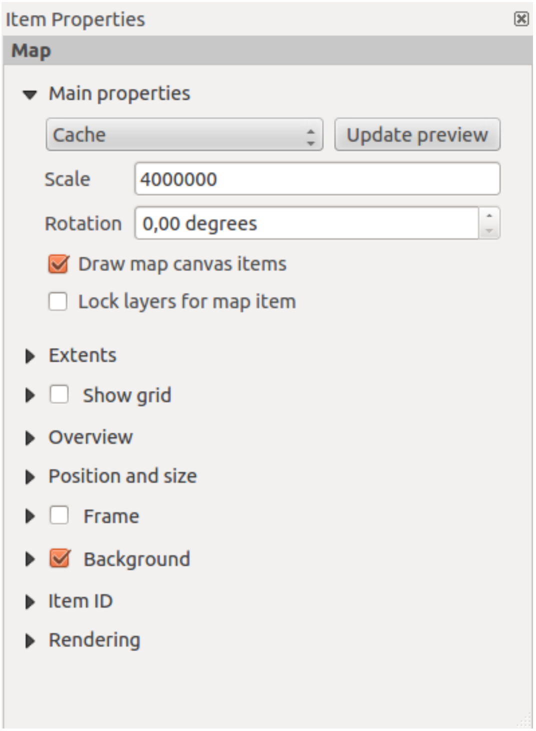


Figure 18.4: Map Item properties Tab

properties, you can update the print composer view selecting the map element in the print composer and clicking the **[Update preview]** button.

- The field *Scale* 1,00 sets a manual scale.
- The field *Rotation* 1,00 allows to rotate the map element content clockwise in degrees. Note, a coordinate frame can only be added with the default value 0.
- The *Draw map canvas items* lets you show annotations that may be placed on the map canvas in the main QGIS window.
- You can choose to lock the layers shown on a map item. Check the *Lock layers for map item*. Any layer that would be displayed or hidden in the main QGIS window after checked on won't appear or be hidden in the map item of the composer. But style and labels of a locked layer is still refreshed accordingly to the main QGIS interface.

Extents

The *Extents* dialog of the map item tab provides following functionalities (see Figure figure_composer_5):

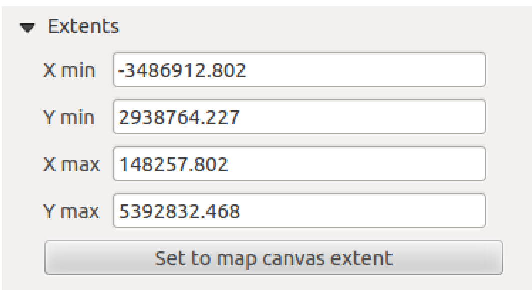


Figure 18.5: Map Extents Dialog

- The **Map extent** area allow to specify the map extent using Y and X min/max values or clicking the **[Set to map canvas extent]** button.

If you change the view on the QGIS map canvas by zooming or panning or changing vector or raster properties, you can update the print composer view selecting the map element in the print composer and clicking the **[Update preview]** button in the map *Item Properties* tab (see Figure figure_composer_2).

Grid

The *Grid* dialog of the map *Item Properties* tab provides following functionalities (see Figure_composer_6):

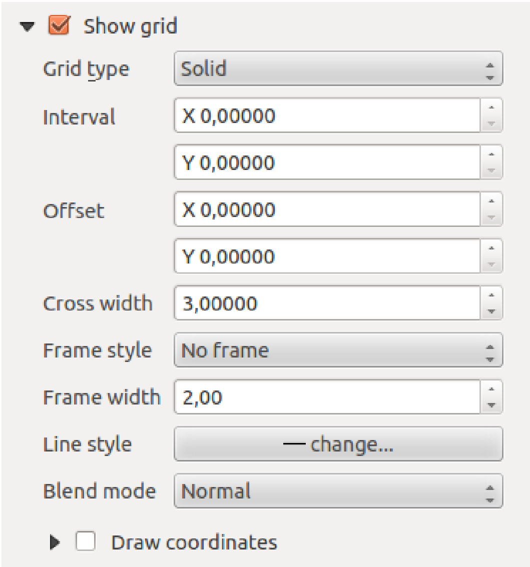


Figure 18.6: Map Grid Dialog

- The *Show grid* checkbox allows to overlay a grid to the map element. As grid type you can specify to use solid line or cross. Symbology of the grid can be chosen. See Section Rendering_Mode. Furthermore you can define an interval in X and Y direction, an X and Y offset, and the width used for cross or line grid type.
- You can choose to paint the frame with a Zebra style. If not selected, general frame option is used (See Section Frame_dialog) Advanced rendering mode is also available for grids. See Section Rendering_mode)
- The *Draw coordinates* checkbox allows to add coordinates to the map frame. The annotation can be drawn inside or outside the map frame. The annotation direction can be defined as horizontal, vertical, horizontal and vertical, or boundary direction, for each border individually. Units can be in meters or in degrees. Finally you can define the grid color, the annotation font, the annotation distance from the map frame and the precision of the drawn coordinates.

Overview

The *Overview* dialog of the map *Item Properties* tab provides following functionalities (see Figure_composer_7):

If the composer has more than one map, you can choose to use a map to show the extents of a second map. The *Overview* dialog of the map *Item Properties* tab allows to customize the appearance of that feature.

- The *Overview frame* combolist references the map item whose extents will be drawn on the present map item.
- The *Overview Style* allows to change the frame color. See Section vector_style_manager .

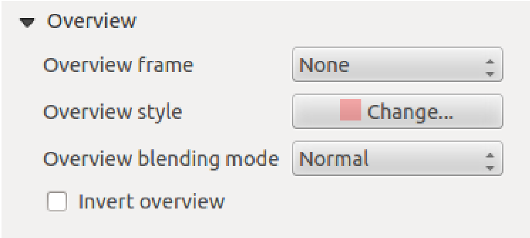


Figure 18.7: Map Overview Dialog

- The *Overview Blend mode* allows different transparency blend modes, to enhance visibility of the frame. See Rendering_Mode
- If checked, the *Invert overview* creates a mask around the extents : the referenced map extents are shown clearly whereas everything else is blended with the frame color.

18.3.2 Adding a Label item to the Print Composer

To add a label, click the Add label icon, place the element with the left mouse button on the print composer canvas and position and customize its appearance in the label *Item Properties* tab.

The *Item Properties* tab of a Label item provides following functionalities:

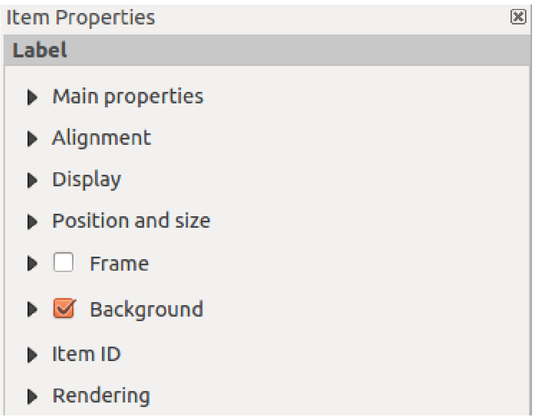


Figure 18.8: Label Item properties Tab

Main properties

The *Main properties* dialog of the Label *Item Properties* tab provides following functionalities (see Figure_composer_9):

- The Main properties dialog is where is inserted the text (html or not) or the expression needed to fill the label added to the composer canvas.
- Labels can be interpreted as html code: check the *Render as HTML*. You can now insert a url, an clickable image that link to a web page or something more complex.
- You can also insert an expression. Click on the **[Insert an expression]** to open a new dialog. Build an expression by clicking the functions available in the left side of the panel. On the right side of the *Insert an expression dialog* is displayed the help file associated with the function selected. Two special categories can be useful, particularly associted with the Atlas functionnality : geometry functions and records functions. On the bottom side, a preview of the expression is shown.

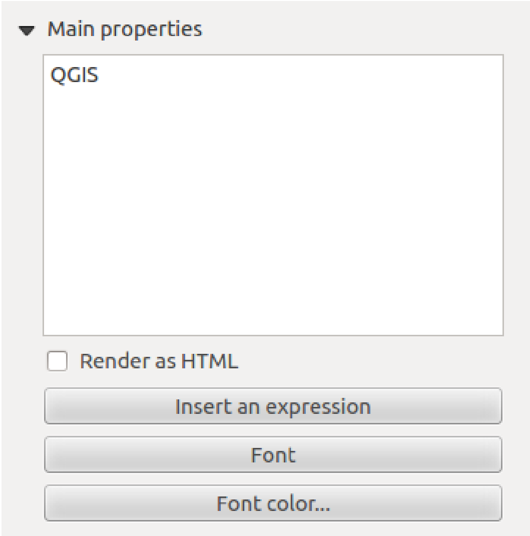


Figure 18.9: Label Main properties Dialog

- Define font and font color by clicking on the **[Font]** and **[Font color...]** buttons

Alignment and Display

The *Alignment* and *Display* dialogs of the Label *Item Properties* tab provide following functionalities (see Figure_composer_10):

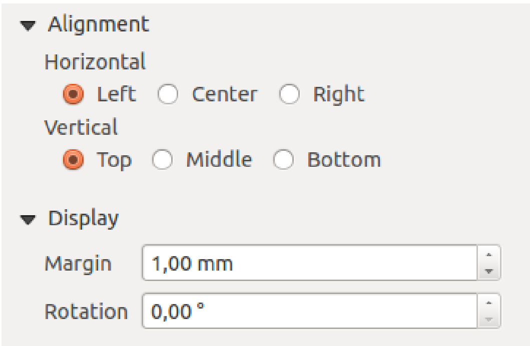


Figure 18.10: Label Alignment and Display Dialogs

- You can define the horizontal and vertical alignment in the *Alignment* zone
- In the **Display** tag, you can define a margin in mm and/or a rotation angle in degrees for the text.

18.3.3 Adding an Image item to the Print Composer

To add an image, click the Add image icon, place the element with the left mouse button on the print composer canvas and position and customize its appearance in the image *Item Properties* tab.

The image *Item Properties* tab provides following functionalities (see figure_composer_11):

Main properties, Search directories and Rotation

The *Main properties* and *Search directories* dialogs of the Image *Item Properties* tab provide following functionalities (see Figure_composer_12):

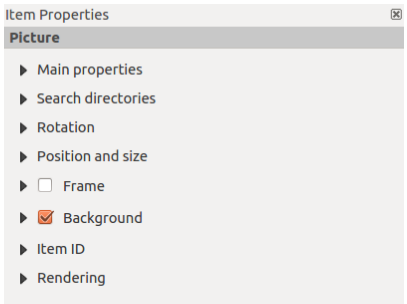


Figure 18.11: Image Item properties Tab

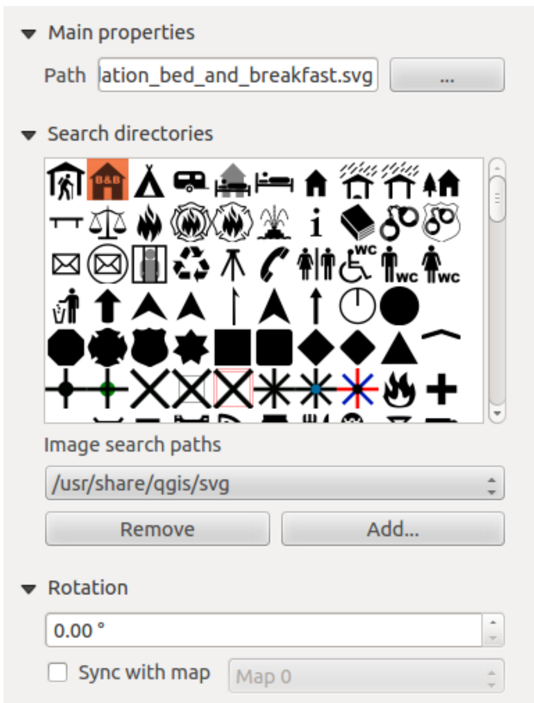


Figure 18.12: Image Main properties, Search directories and Rotation Dialogs

- The **Main properties** dialog shows the current image that is displayed in the image item. Click on the **[...]** button to select a file on your computer.
- This dialog shows all pictures stored in the selected directories.
- The **Search directories** area allows to add and remove directories with images in SVG format to the picture database.
- Image can be rotate, with the *Rotation* 1,00 field.
- Activating the *Sync with map* checkbox synchronizes the rotation of a picture in the QGIS map canvas (i.e. a rotated north arrow) with the appropriate print composer image.

18.3.4 Adding a Legend item to the Print Composer

To add a map legend, click the Add new legend icon, place the element with the left mouse button on the print composer canvas and position and customize their appearance in the legend *Item Properties* tab.

The *Item properties* of a legend item tab provides following functionalities (see figure_composer_14):

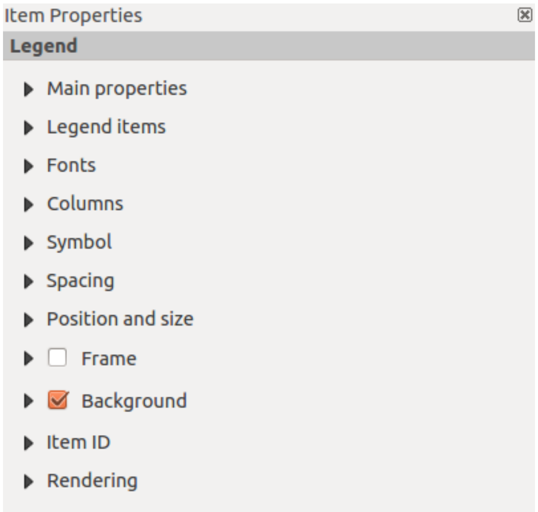


Figure 18.13: Legend Item properties Tab

Main properties

The *Main properties* dialog of the legend *Item Properties* tab provides following functionalities (see figure_composer_14):

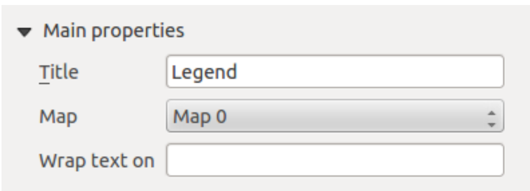


Figure 18.14: Legend Main properties Dialog

- Here you can adapt the legend title.
- Choose which *Map* item the current legend will refer to in the select list.
- Since QGIS 1.8, you can wrap the text of the legend title to a given character.

Legend items

The *Legend items* dialog of the legend *Item Properties* tab provides following functionalities (see figure_composer_15):

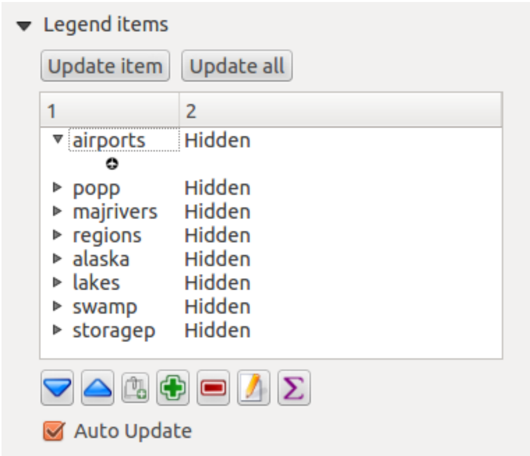


Figure 18.15: Legend Legend Items Dialog

- The legend items window lists all legend items and allows to change item order, group layers, remove and restore items of the list, edit layer names. After changing the symbology in the QGIS main window you can click on **[Update]** to adapt the changes in the legend element of the print composer. The item order can be changed using the **[Up]** and **[Down]** buttons or with 'drag and drop' functionality.
- The feature count for each vector layer can be shown by enable the **[Sigma]** button.
- Legend can be updated automatically, *Auto-update* is checked.

Fonts, Columns, Symbol and Spacing

The *Fonts*, *Columns*, *Symbol* and *Spacing* dialogs of the legend *Item Properties* tab provide following functionalities (see figure_composer_16):

- You can change the font of the legend title, group, subgroup and item (layer) in the legend item. Click on a category button to open a **Select font** dialog.
- All those items will get the same **Color**
- Legend items can be arranged in several columns. Select the correct value in the *Count* 1,00 field.
- The *Equal columns widths* sets how legend columns should be adjusted.
- The *Split layers* option allows a categorized or a graduated layer legend to be divided upon columns.
- You can change width and height of the legend symbol in this dialog.
- Spacing aroung title, group, subgroup, symbol, icon label, box space or column space can be customized throught that dialog.

18.3.5 Adding a Scalebar item to the Print Composer

To add a scalebar, click the Add new scalebar icon, place the element with the left mouse button on the print composer canvas and position and customize their appearance in the scalebar *Item Properties* tab.

The *Item properties* of a scalebar item tab provides following functionalities (see figure_composer_17):

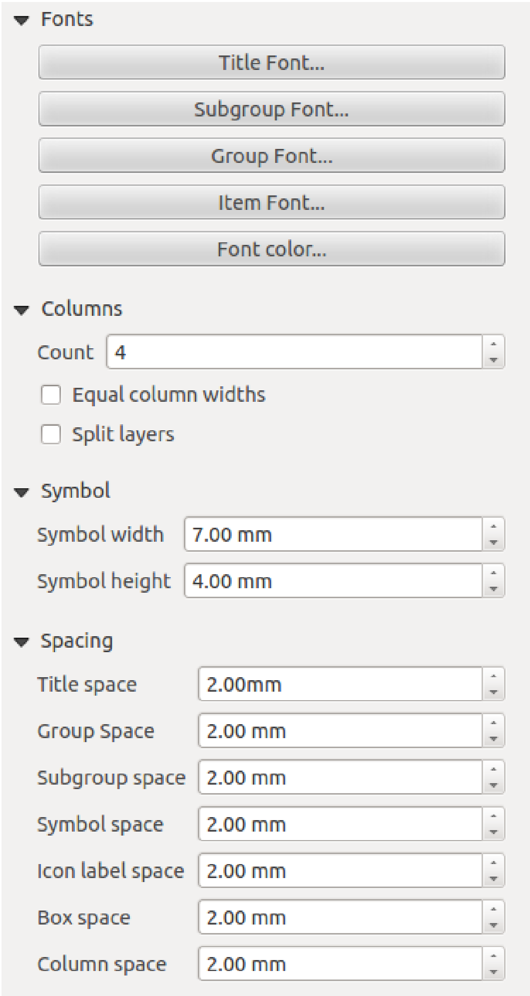


Figure 18.16: Legend Fonts, Columns, Symbol and Spacing Dialogs

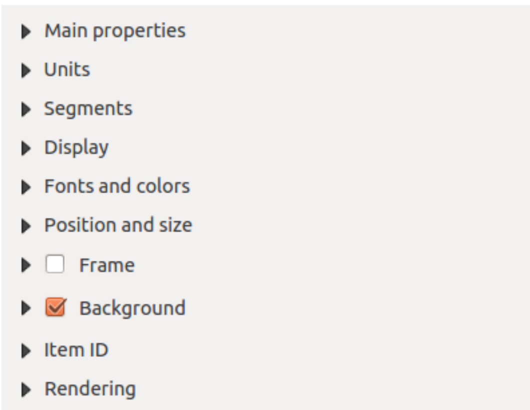


Figure 18.17: Scalebar Item properties Tab

Main properties

The *Main properties* dialog of the scalebar *Item Properties* tab provides following functionalities (see figure_composer_18):

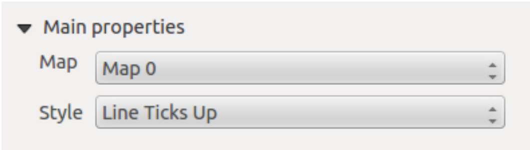


Figure 18.18: Scalebar Main properties Dialog

- First choose the map the scalebar will be attached to.
- then choose the style of your scalebar. Six styles are available :
- **Single box** and **Double box** styles which contain one or two lines of boxes alternating colors,
- **Middle**, **Up** or **Down** line ticks,
- **Numeric** : the scale ratio is printed, i.e. 1:50000.

Units and Segments

The *Units* and *Segments* dialogs of the scalebar *Item Properties* tab provide following functionalities (see figure_composer_19):

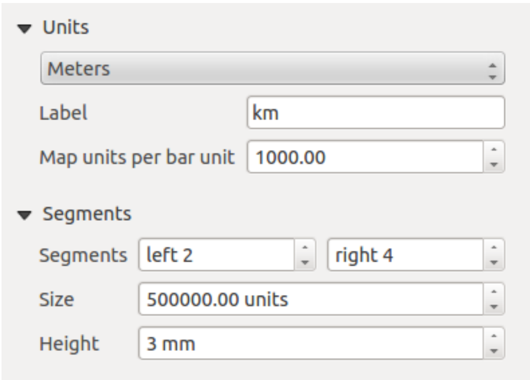


Figure 18.19: Scalebar Units and Segments Dialogs

In those two dialogs, you can set how the scalebar will be represented.

- Select the map units used. There's three possible choices : **Map Units** is the automated unit selection, **Meters** or **Feet** force unit conversions.
- The *Label* field defines the text used to describe the unit of the scalebar.
- The *Map units per bar unit* allows to fix the ratio between a map unit and its representation in the scalebar.
- You can define how many *Segments* will be drawn on the left and on the right side of the scalebar, and how long will be each segment (*Size* field). *Height* can also be defined.

Display, Fonts and colors

The *Display* and *Fonts and colors* dialogs of the scalebar *Item Properties* tab provide following functionalities (see figure_composer_20):

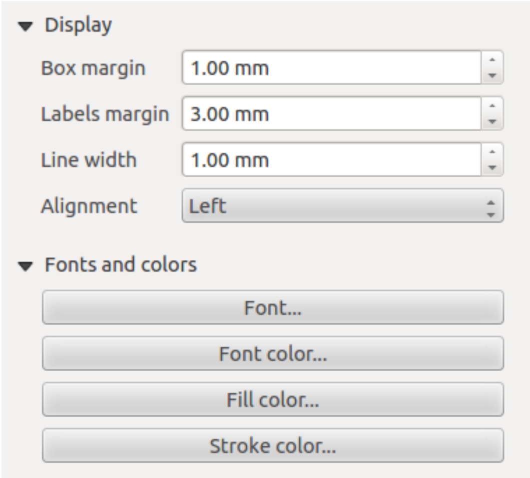


Figure 18.20: Scalebar Display, Fonts and colors Dialogs

- You can define how the scalebar will be displayed in its frame. Adjust the *Box margin* between text and frame borders, *Labels margin* between text and scalebar drawing and the *Line width* of the scalebar drawing.
- The *Alignment* in the *Display* dialog only applies to *Numeric* styled scalebars and puts text on the left, middle or right side of the frame.

18.3.6 Adding a Basic shape or Arrow item to the Print Composer

It is possible to add basic shapes (Ellipse, Rectangle, Triangle) and arrows to the print composer canvas : click the Add basic shape icon or the Add Arrow icon, place the element with the left mouse button on the print composer canvas and position and customize their appearance in the *Item Properties* tab.

The *Shape* Item properties tab allows to draw an ellipse, rectangle, or triangle in the print composer canvas. You can define its outline and fill color, the outline width and a clockwise rotation.

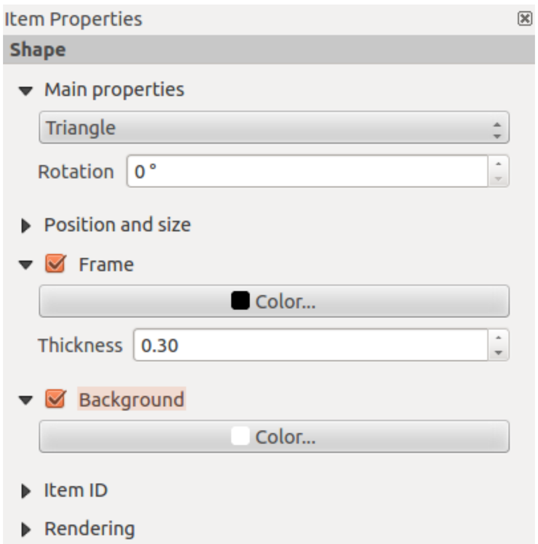


Figure 18.21: Shape Item properties Tab

The *Arrow* Item properties tab allows to draw an arrow in the print composer canvas. You can define color, outline and arrow width and it is possible to use a default marker and no marker and a SVG marker. For the SVG marker you can additionally add a SVG start and end marker from a directory on your computer.

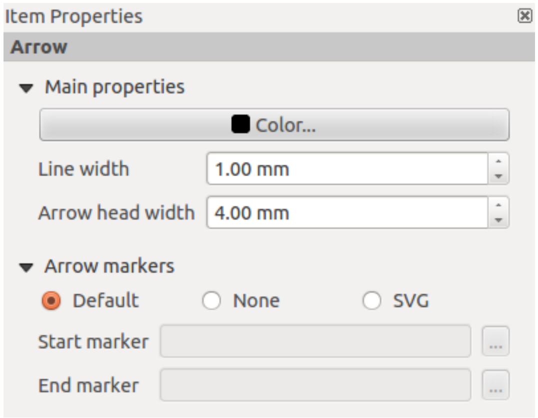


Figure 18.22: Arrow Item properties Tab

Main properties

- For Basic shapes, this dialog allows you to choose a **Ellipse**, **Rectangle** or **Triangle** shape and its rotation.
- Unlike the other items, line style, line color and background color of a basic shape are adjusted with the Frame and Background dialog. No frame is drawn.
- For arrows, you can define here the line style : *Color*, *Line width* and *Arrow head width*.
- *Arrows markers* can be adjusted. If you want to set a SVG *Start marker* and/or *End marker*, browse to your SVG file by clicking on the **[...]** button after selecting *SVG* radio button.

Note: Unlike other items, background color for a basic shape is the shape background and not the frame one.

18.3.7 Add attribute table values to the Print Composer

It is possible to add parts of a vector attribute table to the print composer canvas : click the Add attribute table icon, place the element with the left mouse button on the print composer canvas and position and customize their appearance in the *Item Properties* tab.

The *Item properties* of a attribute table item tab provides following functionalities (see figure_composer_23):

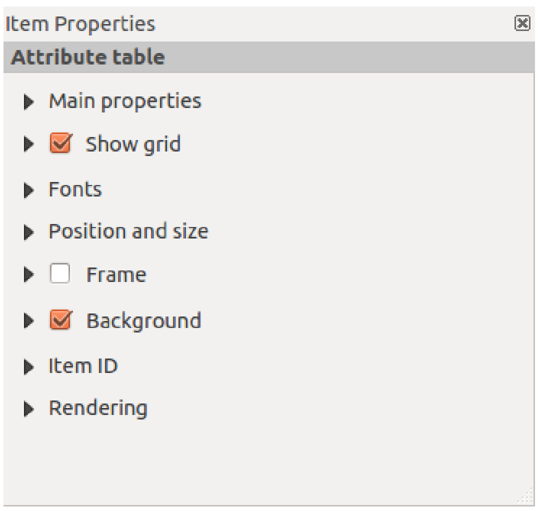


Figure 18.23: Scalebar Item properties Tab

Main properties, Show grid and Fonts

The *Main properties*, *Show grid* and *Fonts* dialogs of the attribute table *Item Properties* tab provide following functionalities (see figure_composer_24):

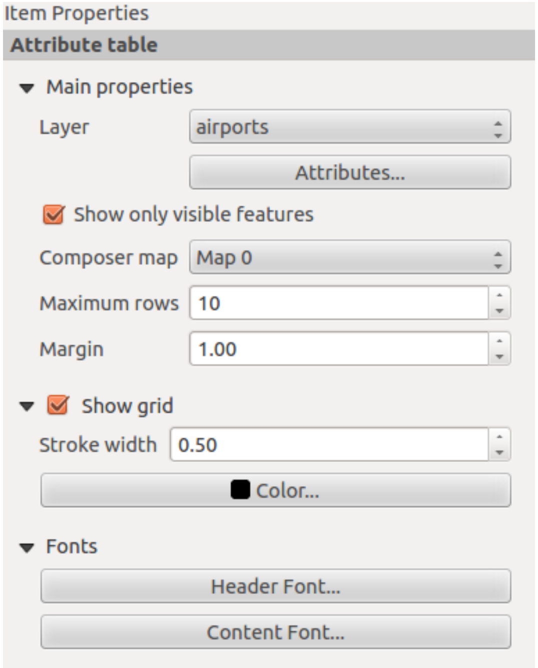


Figure 18.24: Attribute table Main properties, Show grid and Fonts Dialog

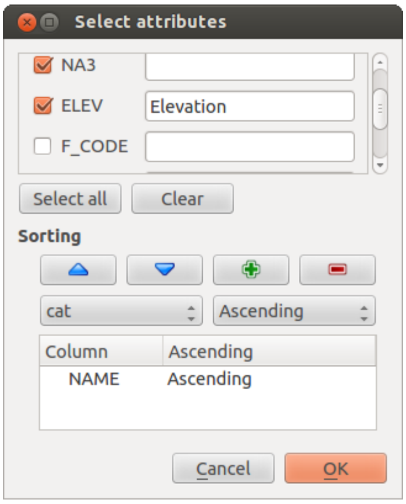


Figure 18.25: Attribute table Select attributes Dialog

- The *Table* dialog allows to select the vector layer and columns of the attribute table. Attribute columns can be sorted and you can define to show its values ascending or descending (see figure_composer_25).
- You can choose to display only the attribute of features visibled on a map. Check *Show only visible features* and select the corresponding *Composer map* to filter.
- You can define the *Maximum number of rows* to be displayed and *margin* around text.
- Additionally you can define the grid characteristics of the table (*Stroke width* and *Color* of the grid) and the header and content font.

18.3.8 Add a HTML frame to the Print Composer

It is possible to add a clickable frame, linked to an URL : click the Add html frame icon, place the element with the left mouse button on the print composer canvas and position and customize their appearance in the *Item Properties* tab.

Main properties

The *Main properties* dialog of the HTML frame *Item Properties* tab provides following functionalities (see figure_composer_26):

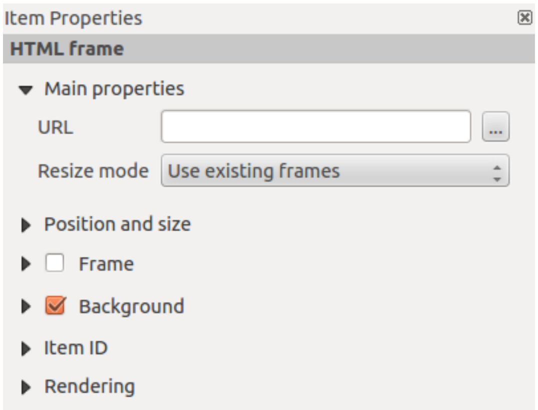


Figure 18.26: HTML frame Item properties Tab

- Point the *URL* field to the URL or the HTML file you want to insert in the composer.
- You can adjust the rendering of that page with the *Resize mode*.
- **Use existing frames** constraints the page inside its first frame or in the frame created with the next settings.
- **Extent to next page** will create as many frames (and their pages) as necessary to render the height of the webpage. Each frame can be moved around on the layout. If you resize a frame, the webpage will be divided up upon the other frames. The last frame will be trimmed to fit the webpage.
- **Repeat on every page** will first repeat the upper left of the webpage on every page, in same sized frames.
- **Repeat until finished** will also create as many frames as the **Extend to next page** option, except All frames will have the same size.

18.4 Item alignment

Raise or lower functionalities for elements are inside the Raise selected items pulldown menu. Choose an element on the print composer canvas and select the matching functionality to raise or lower the selected element compared to the other elements (see table_composer_1).

There are several alignment functionalities available within the Align selected items pulldown menu (see table_composer_1). To use an alignment functionality , you first select some elements and then click on the matching alignment icon. All selected will then be aligned within to their common bounding box. When moving items on the composer canvas, alignment helper lines appear when borders, centers or corners are aligned.

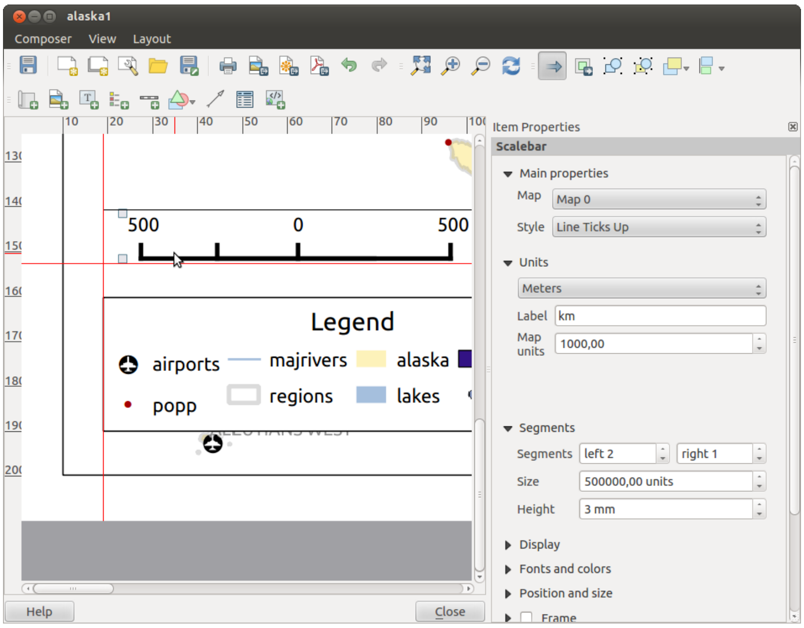


Figure 18.27: Alignment helper lines in the Print Composer

18.4.1 Revert and Restore tools

During the layout process it is possible to revert and restore changes. This can be done with the revert and restore tools:

- Revert last changes
- Restore last changes

or by mouse click within the *Command history* tab (see figure_composer_28).

Figure 18.28: Command history in the Print Composer

18.5 Atlas generation

The print composer includes generation functions that allow to create map books in an automated way. The concept is to use a coverage layer, which contains geometries and fields. For each geometry in the coverage layer, a new output will be generated where the content of some canvas maps will be moved to highlight the current geometry. Fields associated to this geometry can be used within text labels.

There can only be one atlas map by print composer but this one can contain multiple pages. Every pages will be generated with each feature. To enable the generation of an atlas and access generation parameters, refer to the *Atlas generation* tab. This tab contains the following widgets (see Figure_composer_29):

- A *Generate an atlas* enables or disables the atlas generation.

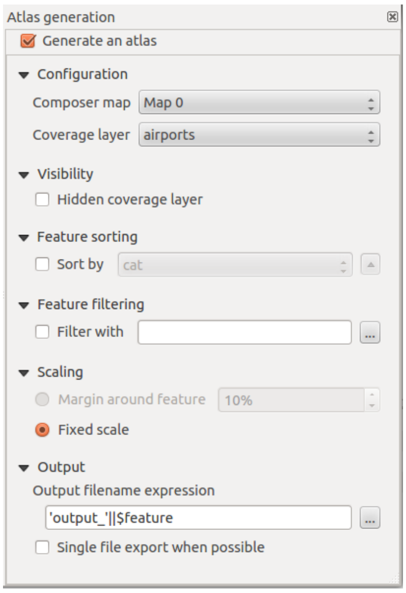


Figure 18.29: Atlas generation tab

- A combobox *Composer map* that allows to choose which map item will be used as the atlas map, i.e. on which map geometries from the coverage layer will be iterated over and displayed.
- A combobox *Coverage layer* that allows to choose the (vector) layer containing the geometries on which to iterate over.
- An optional *Hidden coverage layer*, that if checked, will hide the coverage layer (but not the other ones) during the generation.
- An optional *Features sorting* that, if checked, allows to sort features of the coverage layer. The associated combobox allows to choose which column will be used as the sorting key. Sort order (either ascending or descending) is set by a two-state button that displays an up or a down arrow.
- An optional *Feature filtering* text area that allows to specify an expression for filtering features from the coverage layer. If the expression is not empty, only features that evaluate to `True` will be selected. The button on the right allows to display the expression builder.
- An input box *Scaling* that allows to select the amount of space added around each geometry within the allocated map. Its value is meaningful only when using the autoscaling mode.
- A *Fixed scale* that allows to toggle between auto-scale and fixed-scale mode. In fixed scale mode, the map will only be translated for each geometry to be centered. In auto-scale mode, the map's extents are computed in such a way that each geometry will appear in its whole.
- An *Output filename expression* textbox that is used to generate a filename for each geometry if needed. It is based on expressions. This field is meaningful only for rendering to multiple files.
- A *Single file export when possible* that allows to force the generation of a single file if this is possible by the chosen output format (PDF for instance). If this field is checked, the value of the *Output filename expression* field is meaningless.

In order to adapt labels to the feature the atlas plugin iterates over, use a label with this special notation *[%expression using field_name%]*. For example, with a city layer with fields CITY_NAME and ZIPCODE, you could insert this :

"*[% 'The area of ' || upper(CITY_NAME) || ',' || ZIPCODE || ' is ' format_number($area/1000000,2) || ' km2' %]*"

And that would result in the generated atlas as

"*The area of PARIS,75001 is 1.94 km2*".

18.5.1 Generation

The atlas generation is done when the user asks for a print or an export. The behaviour of these functions will be slightly changed if an atlas map has been selected. For instance, when the user asks for an export to PDF, if an atlas map is defined, the user will be asked for a directory where to save all the generated PDF files (except if the *Single file export when possible* has been selected).

18.6 Creating Output

Figure_composer_30 shows the print composer with an example print layout including each type of map element described in the sections above.

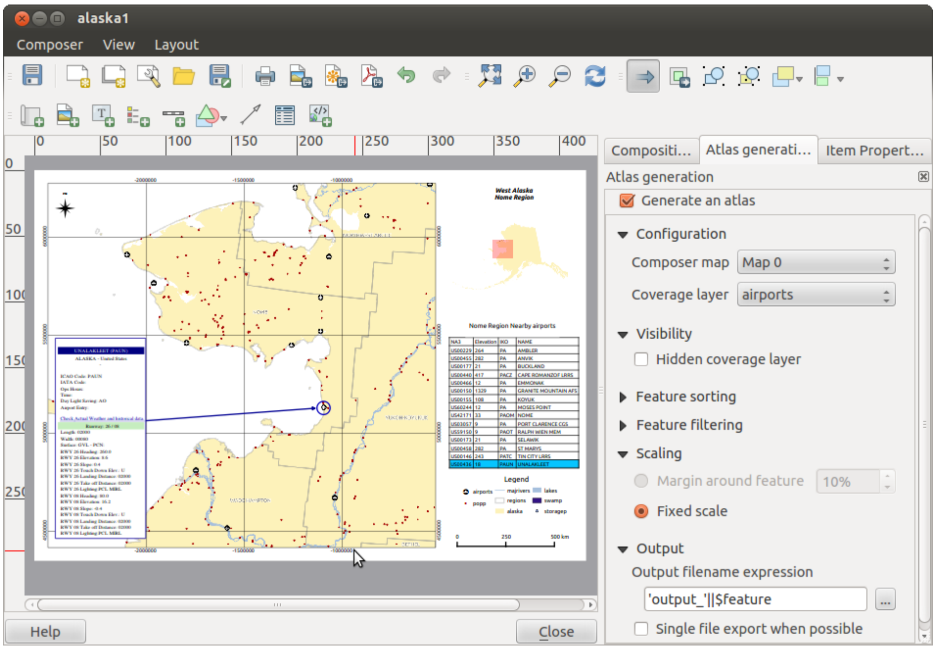


Figure 18.30: Print Composer with map view, legend, image, scalebar, coordinates , text and HTML frame added

The print composer allows you to create several output formats and it is possible to define the resolution (print quality) and paper size:

- The Print icon allows to print the layout to a connected printer or a Postscript file depending on installed printer drivers.
- The Export as image icon exports the composer canvas in several image formats such as PNG, BPM, TIF, JPG,...
- The Export as PDF saves the defined print composer canvas directly as a PDF.
- The Export as SVG icon saves the print composer canvas as a SVG (Scalable Vector Graphic).

Note: Currently the SVG output is very basic. This is not a QGIS problem, but a problem of the underlaying Qt library. This will hopefully be sorted out in future versions. Export big raster can sometimes fail, even if there seems to be enough memory. This is also a problem of the underlaying Qt management of raster.

18.7 Manage the Composer

With the Save as template and Load from template icons you can save the current state of a print composer session as a .qpt template and load the template again in another session.

The Composer Manager button in the QGIS toolbar and in *Composer* → *Composer Manager* allows to add a new composer template, create a new composition based on a previously saved template or to manage already existing templates.

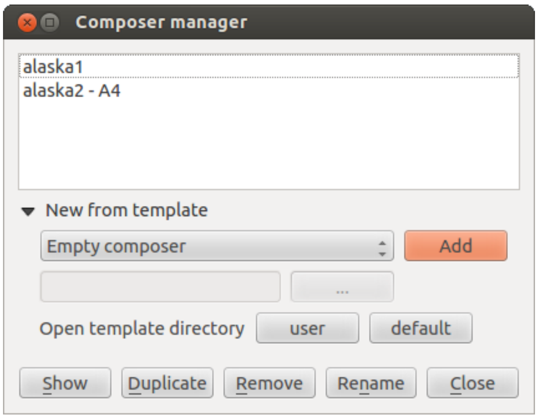


Figure 18.31: The Print Composer Manager

By default, the composer manager searches for user templates in ~/.qgis2/composer_template.

The New Composer and Duplicate Composer buttons in the QGIS toolbar and in *Composer* → *New Composer* and *Composer* → *Duplicate Composer* allow to open a new composer dialog, or to duplicate an existing composition from a previously created one.

Finally you can save your print composition with the Save Project button. This is the same feature as in the QGIS main window. All changes will be saved in a QGIS project file.

CHAPTER 19

Plugins

19.1 QGIS Plugins

QGIS has been designed with a plugin architecture. This allows many new features/functions to be easily added to the application. Many of the features in QGIS are actually implemented as either **core** or **external plugins**.

- **Core Plugins** are maintained by the QGIS Development Team and are automatically part of every QGIS distribution. They are written in one of two languages: C++ or Python. More information about core plugins are provided in Section *Using QGIS Core Plugins*
- **External Plugins** are currently all written in Python. They are stored in external repositories and maintained by the individual authors. They can be added to QGIS in the *Get more* section the of *Plugin Manager*. More information about external plugins is provided in Section *Loading an external QGIS Plugin*.

19.1.1 Managing Plugins

Managing plugins in general means loading or unloading them using the *Plugin Manager*. To deactivate and reactivate external plugins, the *Plugin Manager* is used again.

Loading a QGIS Core Plugin

Loading a QGIS Core Plugin is done from the main menu *Plugins* → *Manage and Install Plugins*.

The *Installed* menu of the *Plugin Manager* lists all the available plugins and their status (loaded or unloaded), including all core plugins and all external plugins that have been installed and automatically activated using the *Get more* menu (see Section *Loading an external QGIS Plugin*). Those plugins that are already loaded have a check mark to the left of their name. Figure_plugins_1 shows the *Installed* dialog.

To enable a particular core plugin, click on the checkbox to the left of the plugin name, and click **[OK]**. When you exit the application, a list of loaded plugins is retained, and the plugins are automatically loaded.

Loading an external QGIS Plugin

External QGIS plugins are written in Python. They are by default stored in either the 'Official' QGIS Repository, or in various other external repositories maintained by individual authors. You can find the external plugins in the *Get more* menu.

In the *Installed* menu you can see the path if it is an external plugin. External plugins are only installed in your home directory while core plugins are stored in `/usr` .

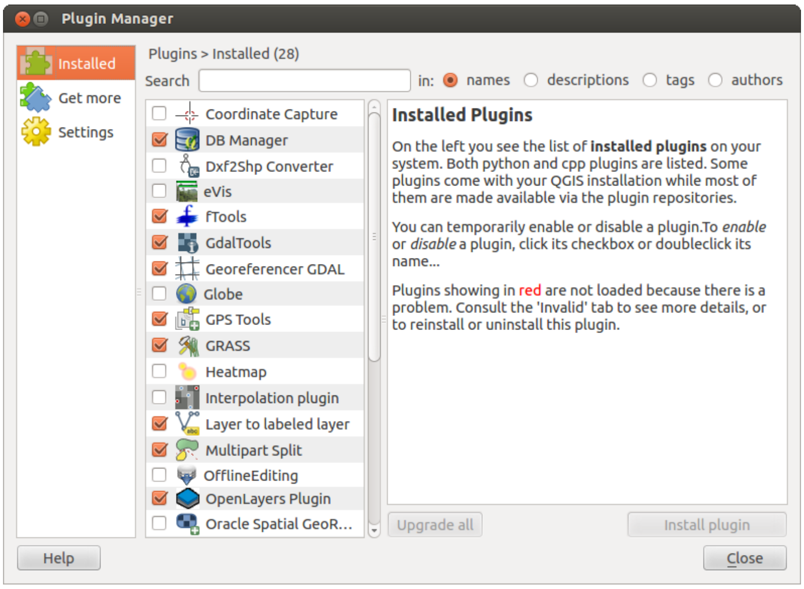


Figure 19.1: Plugin Manager

Detailed documentation about the usage, minimum QGIS version, homepage, authors, and other important information are provided for the 'Official' QGIS Repository at http://plugins.qgis.org/plugins/. For other external repositories, they might be available with the external plugins themselves. In general it is not included in this manual.

Currently there are over 150 external plugins available from the 'Official' QGIS Repository. Some of these plugins offer functionality that will be required by many users (for example: providing the user with the ability to view and edit OpenStreetMap data, or to add GoogleMap layers) while others offer very specialized functions (for example: Calculate economic pipe diameters for water supply networks).

It is, however, quite straightforward to search through all the available external plugins by providing keywords, choosing a named repository and/or filtering on the status of plugins (currently installed or uninstalled in your system). Searching and filtering is done from the QGIS Python Plugin Installer

Tip: Add more repositories

To add external author repositories, open the Plugin Installer *Manage and Install Plugins*, go to the menu *Settings*, and click **[Add]** (see figure_plugins_2). If you do not want one or more of the added repositories, they can be disabled via the **[Edit...]** button, or completely removed with the **[Delete]** button.

As such, we cannot take any responsibility for them. You can also manage the repository list manually, that is add, remove, and edit the entries. Temporarily disabling a particular repository is possible by clicking the **[Edit ...]** button.

Tip: Check for updates

You can stay informed about plugins. Activate the checkbox *Check for updates on startup* from the *Settings* menu. QGIS will inform you whenever a plugin or plugin update is available.

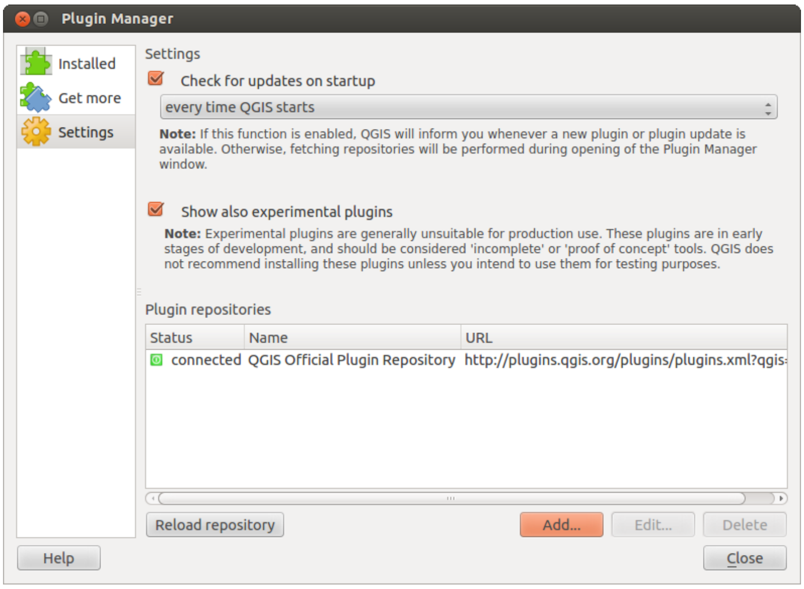


Figure 19.2: Plugin Settings

19.2 Using QGIS Core Plugins

Icon	Plugin	Description	Manual Reference
	Coordinate Capture	Capture mouse coordinate in different CRS	*Coordinate Capture Plugin*
	DB Manager	Manage your databases within QGIS	*DB Manager Plugin*
	DXF2Shape Converter	Converts from DXF to SHP file format	*Dxf2Shp Converter Plugin*
	eVis	Event Visualization Tool	*eVis Plugin*
	fTools	A suite of vector tools	*fTools Plugin*
	GPS Tools	Tools for loading and importing GPS data	*GPS Plugin*
	GRASS	GRASS functionality	*GRASS GIS Integration*
	GDAL Tools	GDAL raster functionality	*GDAL Tools Plugin*
	Georeferencer GDAL	Georeference Raster with GDAL	*Georeferencer Plugin*
	Heatmap	Create heatmap raster from input vector points	*Heatmap Plugin*
	Interpolation plugin	Interpolation on base of vertices of a vector layer	*Interpolation Plugin*
	Offline Editing	Offline editing and synchronizing with database	*Offline Editing Plugin*
	Oracle Spatial Georaster	Access Oracle Spatial GeoRasters	*Oracle GeoRaster Plugin*
	Plugin Manager	Manage core and external plugins	*Managing Plugins*
	Raster Terrain Analysis	Compute geomorphological features from DEMs	*Raster Terrain Analysis Plugin*
	Road Graph Plugin	Shortest path analysis	*Road Graph Plugin*
	SQL Anywhere plugin	Access SQL anywhere DB	*SQL Anywhere Plugin*
	Spatial Query	Spatial queries on vectors	*Spatial Query Plugin*
	Zonal Statistics	Calculate raster statistics for vector polygons	*Zonal Statistics Plugin*

19.3 Coordinate Capture Plugin

The coordinate capture plugin is easy to use and provides the ability to display coordinates on the map canvas for two selected Coordinate Reference Systems (CRS).

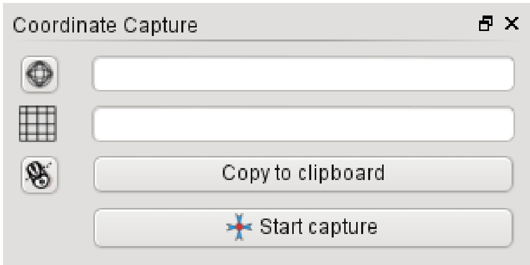


Figure 19.3: Coordinate Capture Plugin

1. Start QGIS, select *Project Properties* from the *Settings* (KDE, Windows) or *File* (Gnome, OSX) menu and click on the *Projection* tab. As an alternative you can also click on the CRS status icon in the lower right-hand corner of the statusbar.
2. Click on the *Enable on the fly projection* checkbox and select a projected coordinate system of your choice (see also *Working with Projections*).
3. Load the coordinate capture plugin in the Plugin Manager (see *Loading a QGIS Core Plugin*) and ensure that the dialog is visible by going to *View* → *Panels* and ensuring that *Coordinate Capture* is enabled. The coordinate capture dialog appears as shown in Figure figure_coordinate_capture_1. Alternatively, you can also go to *Vector* → *Coordinate Capture* and see if *Coordinate Capture* is enabled.
4. Click on the Click to the select the CRS to use for coordinate display icon and select a different CRS from the one you selected above.
5. To start capturing coordinates, click on **[Start capture]**. You can now click anywhere on the map canvas and the plugin will show the coordinates for both of your selected CRS.
6. To enable mouse coordinate tracking click the mouse tracking icon.
7. You can also copy selected coordinates to the clipboard.

19.4 DB Manager Plugin

The DB Manager Plugin is officially part of QGIS core and intends to replace the SPIT Plugin and additionally to integrate all other database formats supported by QGIS in one user interface. The DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported. You can also use the DB Manager to execute SQL queries against your spatial database and then view the spatial output for queries by adding the results to QGIS as a query layer.

The *Database* menu allows to connect to an existing database, to start the SQL-window and to exit the DB Manager Plugin. One you are connected to an existing database, the menus *Schema* and *Table* additionally appear.

The *Schema* menu includes tools to create and delete (empty) schemas and, if topology is available (e.g. PostGIS 2) to start a *TopoViewer*.

The menu *Table* allows to create and edit tables and to delete tables and views. It is also possible to empty tables and to move tables from one to another schema. As further functionality you can perform a VACUUM and then an ANALYZE for each selected table. Plain VACUUM simply reclaims space and makes it available for re-use

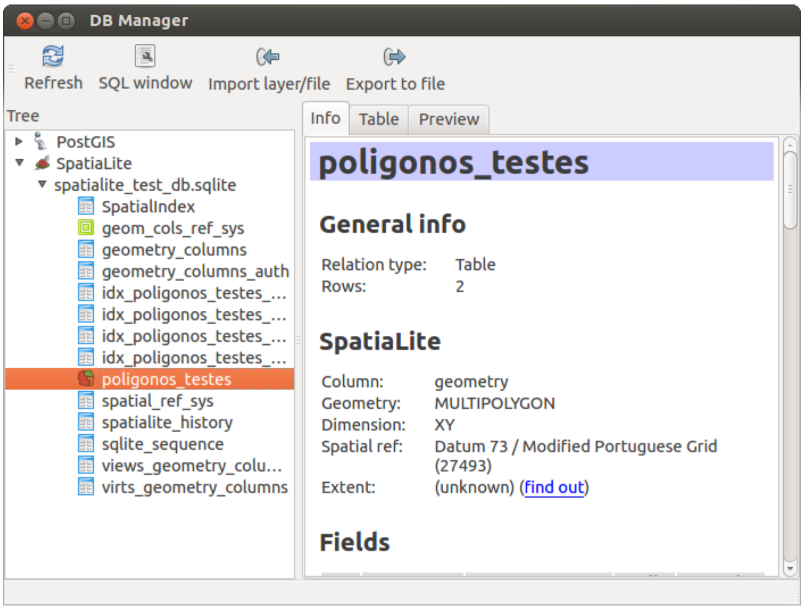


Figure 19.4: DB Manager dialog

and ANALYZE updates statistics to determine the most efficient way to execute a query. Finally you can import layers/files, if they are loaded in QGIS or exist in the file system. And you can export database tables to Shape with the Export file feature.

The *Tree* window lists all existing databases supported by QGIS. With a double-click you can connect to the database. With the right-mouse button you can rename and delete existing schemas and tables. Tables can also be added to the QGIS canvas with the context menu.

If connected to a database, the **main** window of the DB Manager offers three tabs. The *Info* tab provides information about the table and its geometry as well as about existing Fields, Constraints and Indexes. It also allows to run Vacuum Analyze and to create a spatial index on a selected table, if not already done. The *Table* tab shows all attributes and the *Preview* tab renders the geometries as preview.

19.5 Dxf2Shp Converter Plugin

The dxf2shape converter plugin can be used to convert vector data from DXF to Shapefile format. It requires the following parameters to be specified before running:

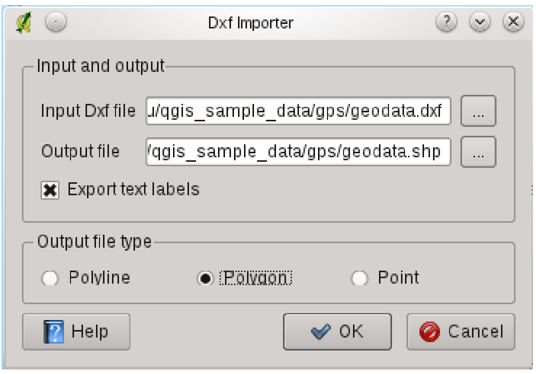


Figure 19.5: Dxf2Shape Converter Plugin

- **Input DXF file**: Enter path to the DXF file to be converted
- **Output Shp file**: Enter desired name of the Shapefile to be created

- **Output file type**: Specify the geometry type of the output Shapefile. Currently supported types are polyline, polygon, and point.
- **Export text labels**: When this checkbox is enabled, an additional Shapefile point layer will be created, and the associated dbf table will contain information about the "TEXT" fields found in the dxf file, and the text strings themselves.

19.5.1 Using the Plugin

1. Start QGIS, load the Dxf2Shape plugin in the Plugin Manager (see Section *Loading a QGIS Core Plugin*) and click on the Dxf2Shape Converter icon which appears in the QGIS toolbar menu. The Dxf2Shape plugin dialog appears as shown in Figure_dxf2shape_1.
2. Enter input DXF file, a name for the output Shapefile and the Shapefile type.
3. Enable the *Export text labels* checkbox if you want to create an extra point layer with labels.
4. Click **[OK]**.

19.6 eVis Plugin

The Biodiversity Informatics Facility at the American Museum of Natural History's (AMNH) Center for Biodiversity and Conservation (CBC) (this section is derived from Horning, N., K. Koy, P. Ersts. 2009. eVis (v1.1.0) User's Guide. American Museum of Natural History, Center for Biodiversity and Conservation. Available from http://biodiversityinformatics.amnh.org/, and released under the GNU FDL.) has developed the Event Visualization Tool (eVis), another software tool to add to the suite of conservation monitoring and decision support tools for guiding protected area and landscape planning. This plugin enables users to easily link geocoded (i.e., referenced with latitude and longitude or X and Y coordinates) photographs, and other supporting documents, to vector data in QGIS.

eVis is now automatically installed and enabled in new versions of QGIS, and as with all plugins, it can be disabled and enabled using the Plugin Manager (See *Managing Plugins*).

The eVis plugin is made up of three modules: the 'Database Connection tool', 'Event ID tool', and the 'Event Browser'. These work together to allow viewing of geocoded photographs and other documents that are linked to features stored in vector files, databases, or spreadsheets.

19.6.1 Event Browser

The Event Browser module provides the functionality to display geocoded photographs that are linked to vector features displayed in the QGIS map window. Point data, for example, can be from a vector file that can be input using QGIS or it can be from the result of a database query. The vector feature must have attribute information associated with it to describe the location and name of the file containing the photograph and, optionally, the compass direction the camera was pointed when the image was acquired. Your vector layer must be loaded into QGIS before running the Event Browser.

Launch the Event Browser module

To launch the Event browser module either click on the eVis Event Browser icon or click on *Database* → *eVis* → *eVis Event Browser*. This will open the *Generic Event Browser* window.

The *Event Browser* window has three tabs displayed at the top of the window. The *Display* tab is used to view the photograph and its associated attribute data. The *Options* tab provides a number of settings that can be adjusted to control the behavior of the eVis plugin. Lastly, the *Configure External Applications* tab is used to maintain a table of file extensions and their associated application to allow eVis to display documents other than images.

Understanding the Display window

To see the *Display* window click on the *Display* tab in the *Event Browser* window. The *Display* window is used to view geocoded photographs and their associated attribute data.

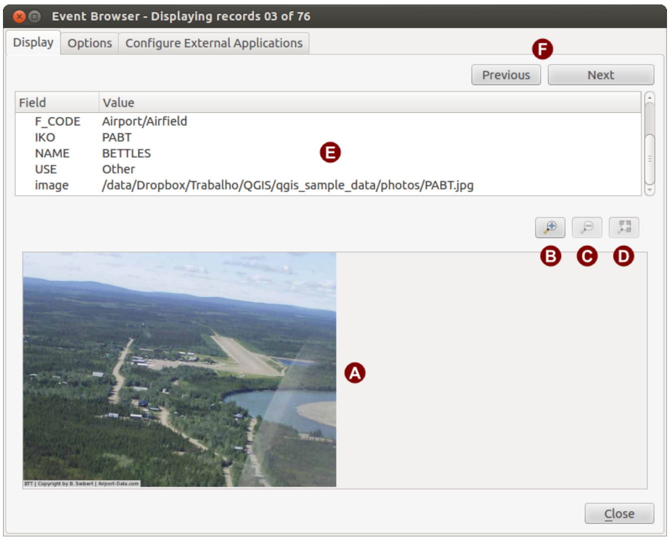


Figure 19.6: The *eVis* display window

1. **Display window**: A window where the photograph will appear.
2. **Zoom in button**: Zoom in to see more detail. If the entire image cannot be displayed in the display window, scroll bars will appear on the left and bottom sides of the window to allow you to pan around the image.
3. **Zoom out button**: Zoom out to see more area.
4. **Zoom to full extent** button: Displays the full extent of the photograph.
5. **Attribute information window**: All of the attribute information for the point associated with the photograph being viewed is displayed here. If the file type being referenced in the displayed record is not an image but is of a file type defined in the *Configure External Applications* tab then when you double-click on the value of the field containing the path to the file the application to open the file will be launched to view or hear the contents of the file. If the file extension is recognized the attribute data will be displayed in green.
6. **Navigation buttons**: Use the Previous and Next buttons to load the previous or next feature when more than one feature is selected.

Understanding the Options window

1. **File path**: A dropdown list to specify the attribute field that contains the directory path or URL for the photographs or other documents being displayed. If the location is a relative path then the checkbox must be clicked. The base path for a relative path can be entered in the *Base Path* text box below. Information about the different options for specifying the file location are noted in the section *Specifying the location and name of a photograph* below.
2. **Compass bearing**: A dropdown list to specify the attribute field that contains the compass bearing associated with the photograph being displayed. If compass bearing information is available it is necessary to click the checkbox below the dropdown menu title.
3. **Compass offset**: Compass offsets can be used to compensate for declination (adjust bearings collected using magnetic bearings to true north bearings). Click the *Manual* radiobutton to enter the offset in the text box or click the *From Attribute* radiobutton to select the attribute field containing the offsets. For both of these options east declinations should be entered using positive values and west declinations should use negative values.

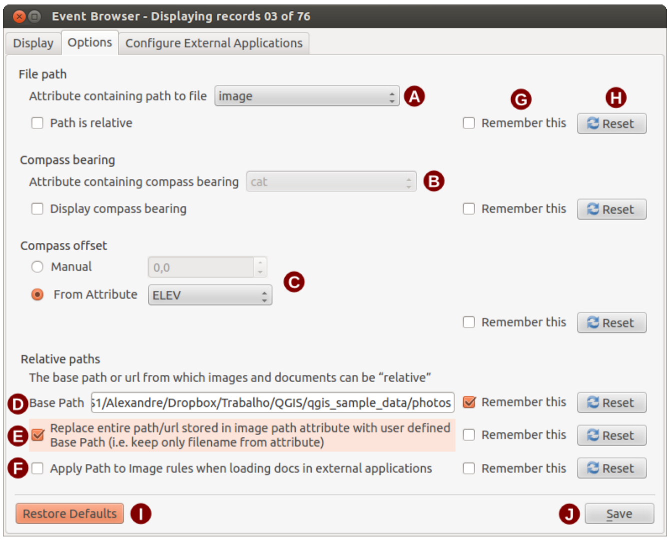


Figure 19.7: The *eVis* Options window

4. **Directory base path**: The base path onto which the relative path defined in Figure_eVis_2 (A) will be appended.
5. **Replace path**: If this checkbox is checked, only the file name from the A will be appended to the Base Path.
6. **Apply rule to all documents**: If checked, the same path rules that are defined for photographs will be used for non-image documents such as movies, text documents, and sound files. If not checked the path rules will only apply to photographs and other documents will ignore the Base Path parameter.
7. **Remember settings**: If the checkbox is checked the values for the associated parameters will be saved for the next session when the window is closed or when the **[Save]** button below is pressed.
8. **Reset values**: Resets the values on this line to the default setting.
9. **Restore defaults**: This will reset all of the fields to their default settings. It has the same effect as clicking all of the **[Reset]** buttons.
10. **Save**: This will save the settings without closing the *Options* pane.

Understanding the Configure External Applications window

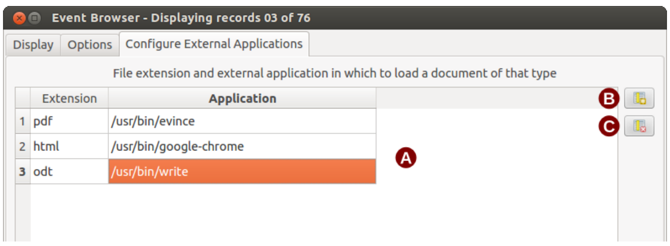


Figure 19.8: The *eVis* External Applications window

1. **File reference table**: A table containing file types that can be opened using eVis. Each file type needs a file extension and the path to an application that can open that type of file. This provides the capability of opening a broad range of files such as movies, sound recordings, and text documents instead of only images.
2. **Add new file type**: Add a new file type with a unique extension and the path for the application that can open the file.
3. **Delete current row**: Delete the file type highlighted in the table and defined by a file extension and a path to an associated application.

19.6.2 Specifying the location and name of a photograph

The location and name of the photograph can be stored using an absolute or relative path or a URL if the photograph is available on a web server. Examples of the different approaches are listed in Table 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
```

19.6.3 Specifying the location and name of other supporting documents

Supporting documents such as text documents, videos, and sound clips can also be displayed or played by eVis. To do this it is necessary to add an entry in the file reference table that can be accessed from the *Configure External Applications* window in the *Generic Event Browser* that matches the file extension to an application that can be used to open the file. It is also necessary to have the path or URL to the file in the attribute table for the vector layer. One additional rule that can be used for URLs that don't contain a file extension for the document you want to open is to specify the file extension before the URL. The format is — `file extension:URL`. The URL is preceded by the file extension and a colon, and is particularly useful for accessing documents from Wikis and other web sites that use a database to manage the web pages (see Table evis_examples).

19.6.4 Using the Event Browser

When the *Event Browser* window opens a photograph will appear in the display window if the document referenced in the vector file attribute table is an image and if the file location information in the *Options* window is properly set. If a photograph is expected and it does not appear it will be necessary to adjust the parameters in the *Options* window.

If a supporting document (or an image that does not have a file extension recognized by eVis) is referenced in the attribute table the field containing the file path will be highlighted in green in the attribute information window if that file extension is defined in the file reference table located in the *Configure External Applications* window. To open the document double-click on the green-highlighted line in the attribute information window. If a supporting document is referenced in the attribute information window and the file path is not highlighted in green then it will be necessary to add an entry for the file's filename extension in the *Configure External Applications* window. If the file path is highlighted in green but does not open when double-clicked it will be necessary to adjust the parameters in the *Options* window so the file can be located by eVis.

If no compass bearing is provided in the *Options* window a red asterisk will be displayed on top of the vector feature that is associated with the photograph being displayed. If a compass bearing is provided then an arrow will appear pointing in the direction indicated by the value in the compass bearing display field in the *Event Browser* window. The arrow will be centered over the point that is associated with the photograph or other document.

To close the *Event Browser* window click on the **[Close]** button from the *Display* window.

19.6.5 Event ID Tool

The 'Event ID' module allows you to display a photograph by clicking on a feature displayed in the QGIS map window. The vector feature must have attribute information associated with it to describe the location and name of the file containing the photograph and optionally the compass direction the camera was pointed when the image was acquired. This layer must be loaded into QGIS before running the 'Event ID' tool.

Launch the Event ID module

To launch the 'Event ID' module either click on the Event ID icon or click on *Database* → *eVis* → *Event ID Tool*. This will cause the cursor to change to an arrow with an 'i' on top of it signifying that the ID tool is active.

To view the photographs linked to vector features in the active vector layer displayed in the QGIS map window, move the Event ID cursor over the feature and then click the mouse. After clicking on the feature, the *Event Browser* window is opened and the photographs on or near the clicked locality are available for display in the browser. If more than one photograph is available, you can cycle through the different features using the **[Previous]** and **[Next]** buttons. The other controls are described in the ref:*evis_browser* section of this guide.

19.6.6 Database connection

The 'Database Connection' module provides tools to connect to and query a database or other ODBC resource, such as a spreadsheet.

eVis can directly connect to four types of databases:PostgreSQL, MySQL, SQLite, and can also read from ODBC connections (e.g.MS Access). When reading from an ODBC database (such as an Excel spreadsheet) it is necessary to configure your ODBC driver for the operating system you are using.

Launch the Database Connection module

To launch the 'Database Connection' module either click on the appropriate icon eVis Database Connection or click on *Database* → *eVis* → *Database Connection*. This will launch the *Database Connection* window. The window has three tabs: *Predefined Queries*, *Database Connection*, and *SQL Query*. The *Output Console* window at the bottom of the window displays the status of actions initiated by the different sections of this module.

Connect to a database

Click on the *Database Connection* tab to open the database connection interface. Next, use the *Database Type* combobox to select the type of database that you want to connect to. If a password or username is required, that information can be entered in the *Username* and *Password* textboxes.

Enter the database host in the *Database Host* textbox. This option is not available if you selected 'MS Access' as the database type. If the database resides on your desktop you should enter "localhost".

Enter the name of the database in the *Database Name* textbox. If you selected 'ODBC' as the database type, you need to enter the data source name.

When all of the parameters are filled in, click on the **[Connect]** button. If the connection is successful, a message will be written in the *Output Console* window stating that the connection was established. If a connection was not established you will need to check that the correct parameters were entered above.

1. **Database Type**: A dropdown list to specify the type of database that will be used.
2. **Database Host**: The name of the database host.
3. **Port** The port number if a MySQL or PostgreSQL database type is selected.
4. **Database Name** The name of the database.
5. **Connect** A button to connect to the database using the parameters defined above.
6. **Output Console** The console window where messages related to processing are displayed.
7. **Username**: Username for use when a database is password protected.
8. **Password**: Password for use when a database is password protected.
9. **Predefined Queries**: Tab to open the "Predefined Queries" window.
10. **Database Connection**: Tab to open the "Database Connection" window.

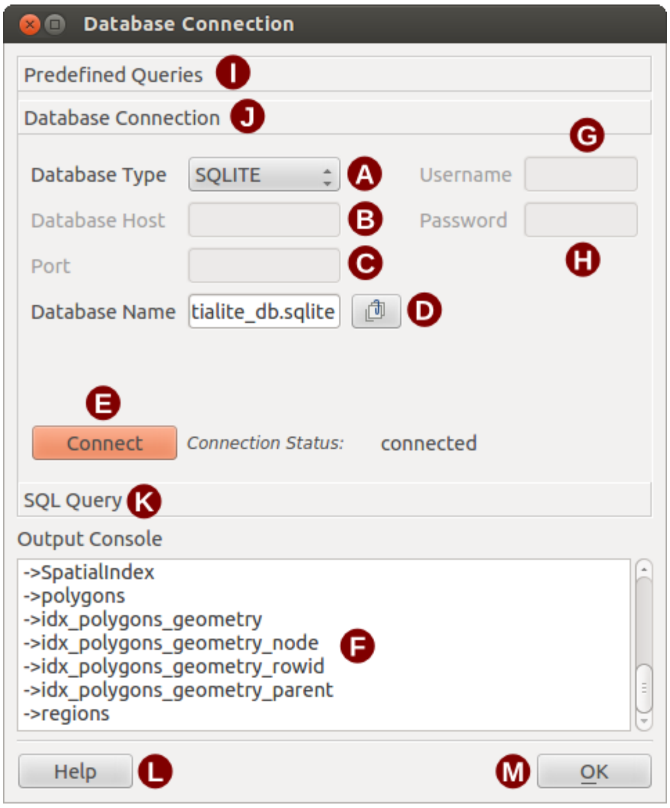


Figure 19.9: The *eVis* Database connection window

11. **SQL Query**: Tab to open the "SQL Query" window.
12. **Help**: Displays the on line help.
13. **OK**: Close the main "Database Connection" window.

Running SQL queries

SQL queries are used to extract information from a database or ODBC resource. In eVis the output from these queries is a vector layer added to the QGIS map window. Click on the *SQL Query* tab to display the SQL query interface. SQL commands can be entered in this text window. A helpful tutorial on SQL commands is available at http://www.w3schools.com/sql. For example, to extract all of the data from a worksheet in an Excel file, `select * from [sheet1$]` where `sheet1` is the name of the worksheet.

Click on the **[Run Query]** button to execute the command. If the query is successful a *Database File Selection* window will be displayed. If the query is not successful an error message will appear in the *Output Console* window.

In the *Database File Selection* window, enter the name of the layer that will be created from the results of the query in the *Name of New Layer* textbox.

1. **SQL Query Text Window**: A screen to type SQL queries.
2. **Run Query**: Button to execute the query entered in the *SQL Query Window*.
3. **Console Window**: The console window where messages related to processing are displayed.
4. **Help**: Displays the on line help.
5. **OK**: Closes the main *Database Connection* window.

Use the *X Coordinate* and *Y Coordinate* comboboxes to select the field from the database that store the `X` (or longitude) and `Y` (or latitude) coordinates. Clicking on the **[OK]** button causes the vector layer created from the SQL query to be displayed in the QGIS map window.

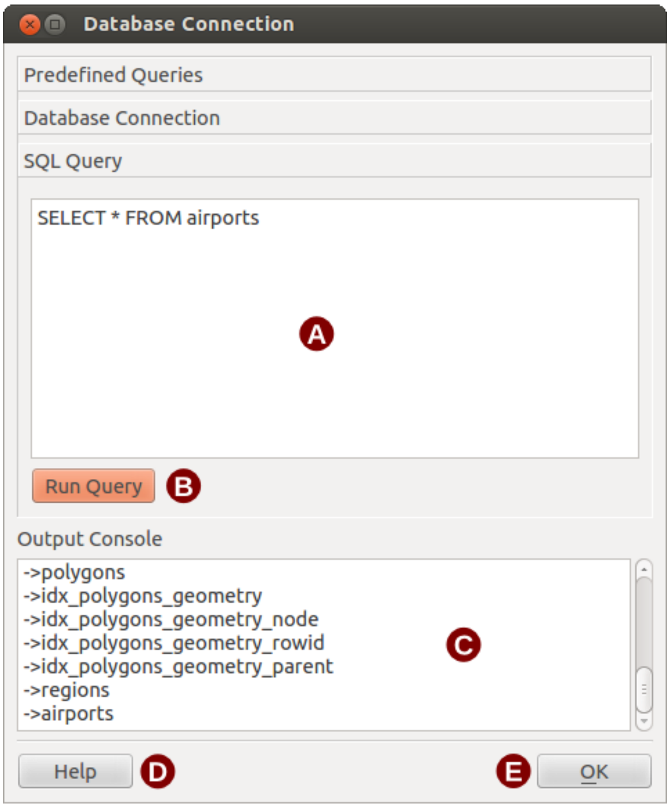


Figure 19.10: The eVis SQL query tab

To save this vector file for future use, you can use the QGIS 'Save as...' command that is accessed by right clicking on the layer name in the QGIS map legend and then selecting 'Save as...'

Tip: Creating a vector layer from a Microsoft Excel Worksheet

When creating a vector layer from a Microsoft Excel Worksheet you might see that unwanted zeros ("0") have been inserted in the attribute table rows beneath valid data.This can be caused by deleting the values for these cells in Excel using the `Backspace` key. To correct this problem you need to open the Excel file (you'll need to close QGIS if there if you are connected to the file to allow you to edit the file) and then use *Edit* → *Delete* to remove the blank rows from the file. To avoid this problem you can simply delete several rows in the Excel Worksheet using *Edit* → *Delete* before saving the file.

Running predefined queries

With predefined queries you can select previously written queries stored in XML format in a file. This is particularly helpful if you are not familiar with SQL commands. Click on the *Predefined Queries* tab to display the predefined query interface.

To load a set of predefined queries click on the Open File icon. This opens the *Open File* window which is used to locate the file containing the SQL queries. When the queries are loaded their titles, as defined in the XML file, will appear in the dropdown menu located just below the Open File icon, the full description of the query is displayed in the text window under the dropdown menu.

Select the query you want to run from the dropdown menu and then click on the *SQL Query* tab to see that the query has been loaded into the query window. If it is the first time you are running a predefined query or are switching databases, you need to be sure to connect to the database.

Click on the **[Run Query]** button in the *SQL Query* tab to execute the command. If the query is successful a *Database File Selection* window will be displayed. If the query is not successful an error message will appear in the *Output Console* window.

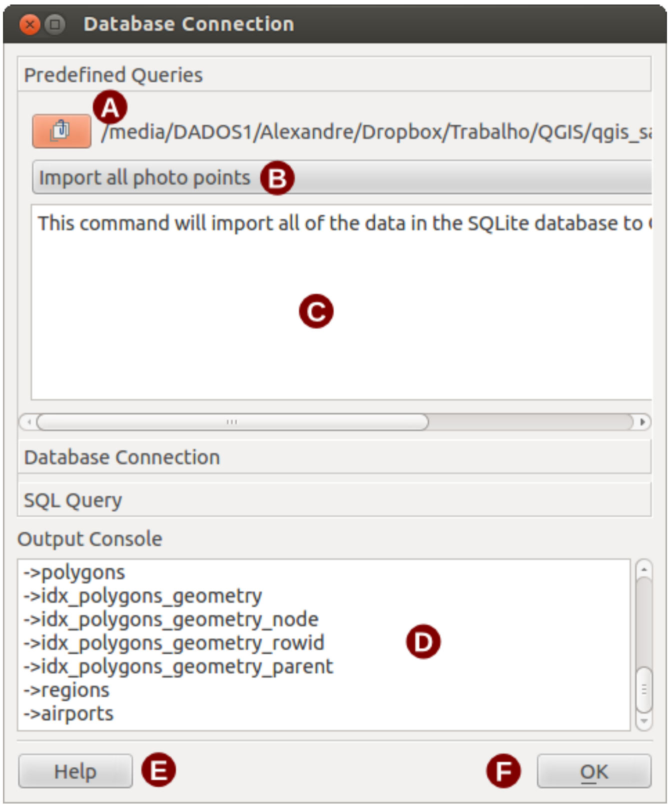


Figure 19.11: The *eVis* Predefined queries tab

1. **Open File**: Launches the "Open File" file browser to search for the XML file holding the predefined queries.
2. **Predefined Queries**: A dropdown list with all of the queries defined by the predefined queries XML file.
3. **Query description**: A short description of the query. This description is from the predefined queries XML file.
4. **Console Window**: The console window where messages related to processing are displayed.
5. **Help**: Displays the on line help.
6. **OK**: Closes the main "Database Connection" window.

XML format for eVis predefined queries

The XML tags read by eVis

Tag	Description
query	Defines the beginning and end of a query statement.
shortdescription	A short description of the query that appears in the eVis dropdown menu.
description	A more detailed description of the query displayed in the Predefined Query text window.
databasetype	The database type as defined in the Database Type dropdown menu in the Database Connection tab.
databaseport	The port as defined in the Port textbox in the Database Connection tab.
databasename	The database name as defined in the Database Name textbox in the Database Connection tab.
databaseusername	The database username as defined in the Username textbox in the Database Connection tab.
databasepassword	The database password as defined in the Password textbox in the Database Connection tab.
sqlstatement	The SQL command.
autoconnect	A flag (“true”” or “false”) to specify if the above tags should be used to automatically connect to database without running the database connection routine in the Database Connection tab.

A complete sample XML file with three queries is displayed below:

```
<?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 />
   <databasename>C:\textbackslash Workshop/textbackslash
eVis\_Data\textbackslash PhotoPoints.db</databasename>
   <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 />
   <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='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>
```

19.7 fTools Plugin

The goal of the fTools python plugin is to provide a one-stop resource for many common vector-based GIS tasks, without the need for additional software, libraries, or complex workarounds. It provides a growing suite of spatial data management and analysis functions that are both fast and functional.

fTools is now automatically installed and enabled in new versions of QGIS, and as with all plugins, it can be disabled and enabled using the Plugin Manager (See Section *Loading a QGIS Core Plugin*). When enabled, the fTools plugin adds a *Vector* menu to QGIS, providing functions ranging from Analysis and Research Tools to Geometry and Geoprocessing Tools, as well as several useful Data Management Tools.

19.7.1 Analysis tools

Icon	Tool	Purpose
	Distance Matrix	Measure distances between two point layers, and output results as a) Square distance matrix, b) Linear distance matrix, or c) Summary of distances. Can limit distances to the k nearest features.
	Sum line length	Calculate the total sum of line lengths for each polygon of a polygon vector layer.
	Points in polygon	Count the number of points that occur in each polygon of an input polygon vector layer.
	List unique values	List all unique values in an input vector layer field
	Basic statistic	Compute basic statistics (mean, std dev, N, sum, CV) on an input field.
	Nearest Neighbor analysis	Compute nearest neighbor statistics assess the level of clustering in a point vector layer
	Mean coordinate(s)	Compute either the normal or weighted mean center of an entire vector layer, or multiple features based on a unique ID field.
	Line intersections	Locate intersections between lines, and output results as a point shapefile. Useful for locating road or stream intersections, ignores line intersections with length > 0.

Table Ftools 1: fTools Analysis tools

19.7.2 Research tools

Icon	Tool	Purpose
	Random selection	Randomly select n number of features, or n percentage of features
	Random selection within subsets	Randomly select features within subsets based on a unique ID field.
	Random points	Generate pseudo-random points over a given input layer.
	Regular points	Generate a regular grid of points over a specified region and export them as a point shapefile.
	Vector grid	Generate a line or polygon grid based on user specified grid spacing.
	Select by location	Select features based on their location relative to another layer to form a new selection, or add or subtract from the current selection.
	Polygon from layer extent	Create a single rectangular polygon layer from the extent of an input raster or vector layer.

Table Ftools 2: fTools Research tools

19.7.3 Geoprocessing tools

Icon	Tool	Purpose
	Convex hull(s)	Create minimum convex hull(s) for an input layer, or based on an ID field.
	Buffer(s)	Create buffer(s) around features based on distance, or distance field.
	Intersect	Overlay layers such that output contains areas where both layers intersect.
	Union	Overlay layers such that output contains intersecting and non-intersecting areas.
	Symmetrical difference	Overlay layers such that output contains those areas of the input and difference layers that do not intersect.
	Clip	Overlay layers such that output contains areas that intersect the clip layer.
	Difference	Overlay layers such that output contains areas not intersecting the clip layer.
	Dissolve	Merge features based on input field. All features with identical input values are combined to form one single feature.
	Eliminate sliver polygons	Merges selected features with the neighbouring polygon with the largest area or largest common boundary.

Table Ftools 3: fTools Geoprocessing tools

19.7.4 Geometry tools

Icon	Tool	Purpose
	Check geometry validity	Check polygons for intersections, closed-holes, and fix node ordering.
	Export/Add geometry columns	Add vector layer geometry info to point (XCOORD, YCOORD), line (LENGTH), or polygon (AREA, PERIMETER) layer.
	Polygon centroids	Calculate the true centroids for each polygon in an input polygon layer.
	Delaunay triangulation	Calculate and output (as polygons) the delaunay triangulation of an input point vector layer.
	Voronoi Polygons	Calculate voronoi polygons of an input point vector layer.
	Simplify geometry	Generalize lines or polygons with a modified Douglas-Peucker algorithm.
	Densify geometry	Densify lines or polygons by adding vertices
	Multipart to singleparts	Convert multipart features to multiple singlepart features. Creates simple polygons and lines.
	Singleparts to multipart	Merge multiple features to a single multipart feature based on a unique ID field.
	Polygons to lines	Convert polygons to lines, multipart polygons to multiple singlepart lines.
	Lines to polygons	Convert lines to polygons, multipart lines to multiple singlepart polygons.
	Extract nodes	Extract nodes from line and polygon layers and output them as points.

Table Ftools 4: fTools Geometry tools

Note: The *Simplify geometry* tool can be used to remove duplicate nodes in line and polygon geometries, just set the *Simplify tolerance* parameter to 0 and this will do the trick.

19.7.5 Data management tools

Icon	Tool	Purpose
	Define current projection	Specify the CRS for shapefiles whose CRS has not been defined.
	Join attributes by location	Join additional attributes to vector layer based on spatial relationship. Attributes from one vector layer are appended to the attribute table of another layer and exported as a shapefile.
	Split vector layer	Split input layer into multiple separate layers based on input field.
	Merge shapefiles to one	Merge several shapefiles within a folder into a new shapefile based on the layer type (point, line, area).
	Create spatial index	Create a spatial index for OGR supported formats.

Table Ftools 5: fTools Data management tools

19.8 GDAL Tools Plugin

19.8.1 What is GDALTools?

The GDAL Tools plugin offers a GUI to the collection of tools in the Geospatial Data Abstraction Library, http://gdal.osgeo.org . These are raster management tools to query, re-project, warp and merge a wide variety of raster formats. Also included are tools to create a contour (vector) layer, or a shaded relief from a raster DEM, and to make a vrt (Virtual Raster Tile in XML format) from a collection of one or more raster files. These tools are available when the plugin is installed and activated.

The GDAL Library

The GDAL library consists of a set of command line programs, each with a large list of options. Users comfortable with running commands from a terminal may prefer the command line, with access to the full set of options. The GDALTools plugin offers an easy interface to the tools, exposing only the most popular options.

19.8.2 List of GDAL tools

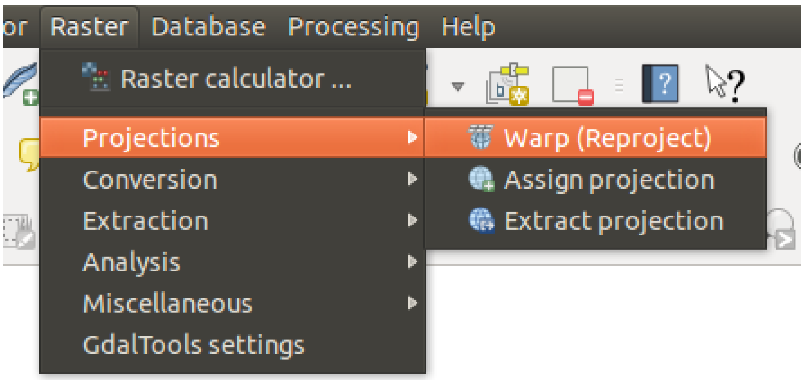


Figure 19.12: The *GDALTools* menu list

Projections

Warp (Reproject)	This utility is an image mosaicing, reprojection and warping utility. The program can reproject to any supported projection, and can also apply GCPs stored with the image if the image is "raw" with control information. For more information you can read on the GDAL website http://www.gdal.org/gdalwarp.html
Assign projection	This tool allows to assign projection to rasters that already georeferenced but miss projection information. Also with it help it is possible to alter existing projection definition. Both single file and batch mode are supported. For more information please visit utility page at GDAL site http://www.gdal.org/gdalwarp.html
Extract projection	This utility helps you to extract projection information from an input file. If you want to extract projection from a whole directory you can use the Batch mode. It creates both `.prj` and `.wld` files.

Conversion

Rasterize	This program burns vector geometries (points, lines and polygons) into the raster band(s) of a raster image. Vectors are read from OGR supported vector formats. Note that the vector data must in the same coordinate system as the raster data; on the fly reprojection is not provided. For more information see http://www.gdal.org/gdal_rasterize.html
Polygonize	This utility creates vector polygons for all connected regions of pixels in the raster sharing a common pixel value. Each polygon is created with an attribute indicating the pixel value of that polygon. The utility will create the output vector datasource if it does not already exist, defaulting to ESRI shapefile format. See also http://www.gdal.org/gdal_polygonize.html
Translate	This utility can be used to convert raster data between different formats, potentially performing some operations like subsettings, resampling, and rescaling pixels in the process. For more information you can read on http://www.gdal.org/gdal_translate.html
RGB to PCT	This utility will compute an optimal pseudo-color table for a given RGB image using a median cut algorithm on a downsampled RGB histogram. Then it converts the image into a pseudo-colored image using the color table. This conversion utilizes Floyd-Steinberg dithering (error diffusion) to maximize output image visual quality. The utility is also desribed at http://www.gdal.org/rgb2pct.html
PCT to RGB	This utility will convert a pseudocolor band on the input file into an output RGB file of the desired format. For more information see http://www.gdal.org/pct2rgb.html

Extraction

Contour	This program generates a vector contour file from the input raster elevation model (DEM). On http://www.gdal.org/gdal_contour.html you can find more information.
Clipper	This utility allows to clip (extract subset) raster using selected extent or based on mask layer bounds. More information can be found at http://www.gdal.org/gdal_translate.html.

Analysis

Sieve	This utility removes raster polygons smaller than a provided threshold size (in pixels) and replaces them with the pixel value of the largest neighbor polygon. The result can be written back to the existing raster band, or copied into a new file. For more information see http://www.gdal.org/gdal_sieve.html .
Near Black	This utility will scan an image and try to set all pixels that are nearly black (or nearly white) around the edge to exactly black (or white). This is often used to "fix up" lossy compressed aerial photos so that color pixels can be treated as transparent when mosaicing. See also http://www.gdal.org/nearblack.html .
Fill nodata	This utility fills selection raster regions (usually nodata areas) by interpolation from valid pixels around the edges of the area. On http://www.gdal.org/gdal_fillnodata.html you can find more information.
Proximity	This utility generates a raster proximity map indicating the distance from the center of each pixel to the center of the nearest pixel identified as a target pixel. Target pixels are those in the source raster for which the raster pixel value is in the set of target pixel values. For more information see http://www.gdal.org/gdal_proximity.html .
Grid (Interpolation)	This utility creates regular grid (raster) from the scattered data read from the OGR datasource. Input data will be interpolated to fill grid nodes with values, you can choose from various interpolation methods. The utility is also described on the GDAL website http://www.gdal.org/gdal_grid.html .
DEM (Terrain models)	Tools to analyze and visualize DEMs. It can create a shaded relief, a slope, an aspect, a color relief, a Terrain Ruggedness Index, a Topographic Position Index and a roughness map from any GDAL-supported elevation raster. For more information you can read on http://www.gdal.org/gdaldem.html

Miscellaneous

Build Virtual Raster (Catalog)	This program builds a VRT (Virtual Dataset) that is a mosaic of the list of input gdal datasets. See also http://www.gdal.org/gdalbuildvrt.html .
Merge	This utility will automatically mosaic a set of images. All the images must be in the same coordinate system and have a matching number of bands, but they may be overlapping, and at different resolutions. In areas of overlap, the last image will be copied over earlier ones. The utility is also described on http://www.gdal.org/gdal_merge.html .
Information	This utility lists various information about a GDAL supported raster dataset. On http://www.gdal.org/gdalinfo.html you can find more information.
Build Overviews	The gdaladdo utility can be used to build or rebuild overview images for most supported file formats with one of several downsampling algorithms. For more information see http://www.gdal.org/gdaladdo.html .
Tile Index	This utility builds a shapefile with a record for each input raster file, an attribute containing the filename, and a polygon geometry outlining the raster. See also http://www.gdal.org/gdaltindex.html .

19.9 Georeferencer Plugin

The Georeferencer Plugin is a tool for generating world files for rasters. It allows you to reference rasters to geographic or projected coordinate systems by creating a new GeoTiff or by adding a world file to the existing image. The basic approach to georeferencing a raster is to locate points on the raster for which you can accurately determine their coordinates.

Features

Icon	Purpose	Icon	Purpose
	Open raster		Start georeferencing
	Generate GDAL Script		Load GCP Points
	Save GCP Points As		Transformation settings
	Add Point		Delete Point
	Move GCP Point		Pan
	Zoom In		Zoom Out
	Zoom To Layer		Zoom Last
	Zoom Next		Link Georeferencer to QGIS
	Link QGIS to Georeferencer		

Table Georeferencer 1: Georeferencer Tools

19.9.1 Usual procedure

As X and Y coordinates (DMS (dd mm ss.ss), DD (dd.dd) or projected coordinates (mmmm.mm) which correspond with the selected point on the image, two alternative procedures can be used:

- The raster itself sometimes provides crosses with coordinates "written" on the image. In this case you can enter the coordinates manually.
- Using already georeferenced layers, this can be either vector or raster data that contain the same objects/features that you have on the image that you want to georeference and the projection you want to have your image. In this case you can enter the coordinates by clicking on the reference dataset loaded in QGIS map canvas.

The usual procedure for georeferencing an image involves selecting multiple points on the raster, specifying their coordinates, and choosing a relevant transformation type. Based on the input parameters and data, the plugin will compute the world file parameters. The more coordinates you provide, the better the result will be.

The first step is to start QGIS, load the Georeferencer Plugin (see Section *Loading a QGIS Core Plugin*) and click on the Georeferencer icon which appears in the QGIS toolbar menu. The Georeferencer Plugin dialog appears as shown in figure_georeferencer_1.

For this example, we are using a topo sheet of South Dakota from SDGS. It can later be visualized together with the data from the GRASS `spearfish60` location. You can download the topo sheet here: http://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz.

Entering ground control points (GCPs)

1. To start georeferencing an unreferenced raster, we must load it using the button. The raster will show up in the main working area of the dialog. Once the raster is loaded, we can start to enter reference points.

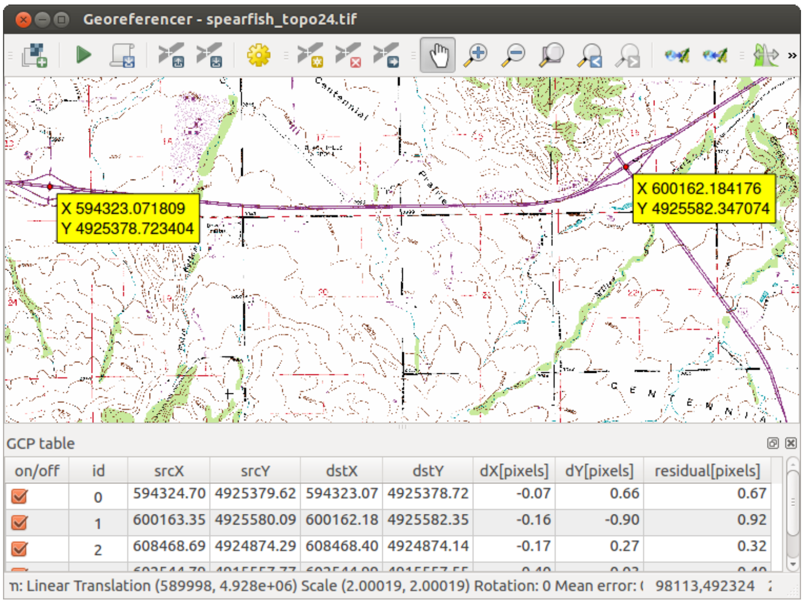


Figure 19.13: Georeferencer Plugin Dialog

2. Using the Add Point button, add points to the main working area and enter their coordinates (see Figure figure_georeferencer_2). For this procedure you have three options:

   - Click on a point in the raster image and enter the X and Y coordinates manually.
   - Click on a point in the raster image and choose the button from map canvas to add the X and Y coordinates with the help of a georeferenced map already loaded in the QGIS map canvas.
   - With the button, you can move the GCPs in both windows, if they are at the wrong place.

3. Continue entering points. You should have at least 4 points, and the more coordinates you can provide, the better the result will be. There are additional tools on the plugin dialog to zoom and pan the working area in order to locate a relevant set of GCP points.

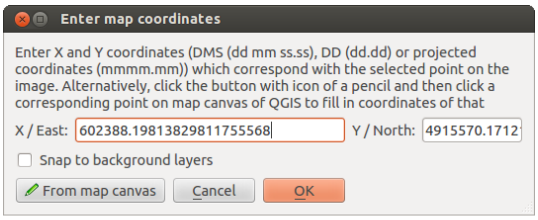


Figure 19.14: Add points to the raster image

The points that are added to the map will be stored in a separate text file (`[filename].points`) usually together with the raster image. This allows us to reopen the Georeferencer plugin at a later date and add new points or delete existing ones to optimize the result. The points file contains values of the form: `mapX, mapY, pixelX, pixelY`. You can use the Load GCP points and Save GCP points as buttons to manage the files.

Defining the transformation settings

After you have added your GCPs to the raster image, you need to define the transformation settings for the georeferencing process.

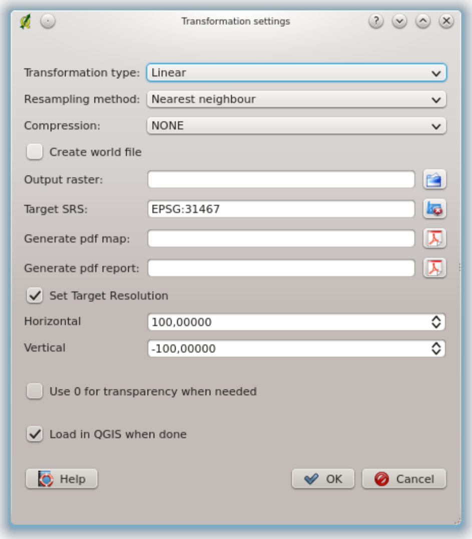


Figure 19.15: Defining the georeferencer transformation settings

Available Transformation algorithms

Depending on how many ground control point you have captured, you may want to use different transformation algorithms. Choice of transformation algorithm is also dependent on the type and quality of input data and the amount of geometric distortion that you are willing to introduce to final result.

Currently, following *Transformation types* are available:

- The **Linear** algorithm is used to create a world-file, and is different from the other algorithms, as it does not actually transform the raster. This algorithm likely won't be sufficient if you are dealing with scanned material.
- The **Helmert** transformation performs simple scaling and rotation transformations.
- The **Polynomial** algorithms 1-3 are among the most widely used algorithms introduced to match source and destination ground control points. The most widely used polynomial algorithm is the second order polynomial transformation, which allows some curvature. First order polynomial transformation (affine) preserves colliniarity and allows scaling, translation and rotation only.
- The **Thin Plate Spline** (TPS) algorithm is a more modern georeferencing method, which is able to introduce local deformations in the data. This algorithm is useful when very low quality originals are being georeferenced.
- The **Projective** transformation is a linear rotation and translation of coordinates.

Define the Resampling method

The type of resampling you choose will likely depending on your input data and the ultimate objective of the exercise. If you don't want to change statistics of the image, you might want to choose 'Nearest neighbour', whereas a 'Cubic resampling' will likely provide a more smoothed result.

It is prossible to choose between five different resampling methods.

1. Nearest neighbour
2. Linear
3. Cubic
4. Cubic Spline

5. Lanczos

Define the transformation settings

There are several options that need to be defined for the georeferenced output raster.

- The checkbox *Create world file* is only available, if you decide to use the linear transformation type, because this means that the raster image actually won't be transformed. In this case, the field *Output raster* is not activated, because only a new world-file will be created.
- For all other transformation type you have to define an *Output raster*. As default a new file ([filename]_modified) will be created in the same folder together with the original raster image.
- As a next step you have to define the *Target SRS* (Spatial Reference System) for the georeferenced raster (see section *Working with Projections*).
- If you like, you can **generate a pdf map** and also **a pdf report**. The report includes information about the used transformation parameters. An image of the residuals and a list with all GCPs and their RMS errors.
- Furthermore you can activate the *Set Target Resolution* checkbox and define pixel resolution of the output raster. Default horizontal and vertical resolution is 1,
- The *Use 0 for transparency when needed* can be activated, if pixels with the value 0 shall be visualized transparent. In our example toposheet all white areas would be transparent.
- Finally *Load in |qg| when done* loads the output raster automatically into the QGIS map canvas when the transformation is done.

Show and adapt raster properties

Clicking on the *Raster properties* dialog in the *Settings* menu opens the raster properties of the layer that you want to georeference.

Configure the georeferencer

- You can define if you want to show GCP coordiniates and/or IDs.
- As residual units pixels and map units can be chosen.
- For the PDF report a left and right margin can be defined and you can also set the paper size for the PDF map.
- Finally you can activate to *show georeferencer window docked*.

Running the transformation

After all GCPs have been collected and all transformation settings are defined, just press the button Start georeferencing to create the new georeferenced raster.

19.10 Interpolation Plugin

The Interplation plugin can be used to generate a TIN or IDW interpolation of a point vector layer. It is very simple to handle and provides an intuitive graphical user interface for creating interpolated raster layers (See Figure_interpolation_1). The plugin requires the following parameters to be specified before running:

- Input **Vector layers**: Specify the input point vector layer(s) from a list of loaded point layers. If several layers are specified, then data from all layers is used for interpolation. Note: It is possible to insert lines or polygons as constraints for the triangulation, by specifying either "points", "structure lines" or "break lines" in the *Type* combobox.
- **Interpolation attribute**: Select attribute column to be used for interpolation or enable the *Use Z-Coordinate* checkbox to use the layers stored Z values.
- **Interpolation Method**: Select interpolation method. This can be either 'Triangulated Irregular Network (TIN)' or 'Inverse Distance Weighted (IDW)'.
- **Number of columns/rows**: Specify the number row and colums for the output raster file.
- **Output file**: Specify a name for the output raster file.
- *Add result to project* to load the result into the map canvas.

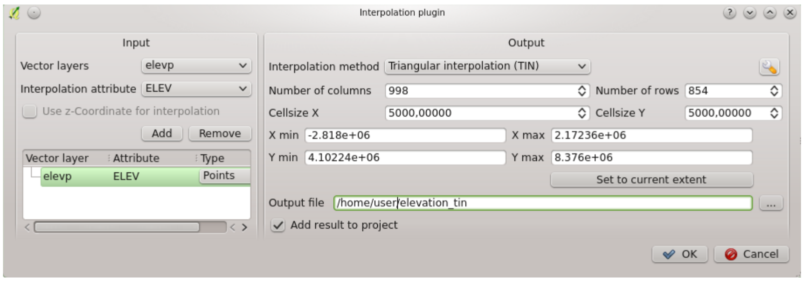


Figure 19.16: Interpolation Plugin

19.10.1 Using the plugin

1. Start QGIS and load a point vector layer (e.g., `elevp.csv`).
2. Load the Interpolation plugin in the Plugin Manager (see Section *Loading a QGIS Core Plugin*) and click on the Interpolation icon which appears in the QGIS toolbar menu. The Interpolation plugin dialog appears as shown in Figure_interpolation_1.
3. Select an input layer (e.g., *elevp*) and column (e.g., `ELEV`) for interpolation.
4. Select an interpolation method (e.g. 'Triangulated Irregular Network (TIN)'), and specify a cellsize of 5000 as well as the raster output filename (e.g., `elevation_tin`).
5. Click **[OK]**.

19.11 Offline Editing Plugin

For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master data source, e.g. a PostGIS database. If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

The Offline Editing Plugin automates the synchronisation by copying the content of a datasource (usually PostGIS or WFS-T) to a SpatiaLite database and storing the offline edits to dedicated tables. After being connected to the network again, it is possible to apply the offline edits to the master dataset.

19.11.1 Using the plugin

- Open some vector layers, e.g. from a PostGIS or WFS-T datasource

- Save it as a project
- Press the Convert to offline project icon and select the layers to save. The content of the layers is saved to SpatiaLite tables.
- Edit the layers offline.
- After being connected again, upload the changes with the Synchronize button.

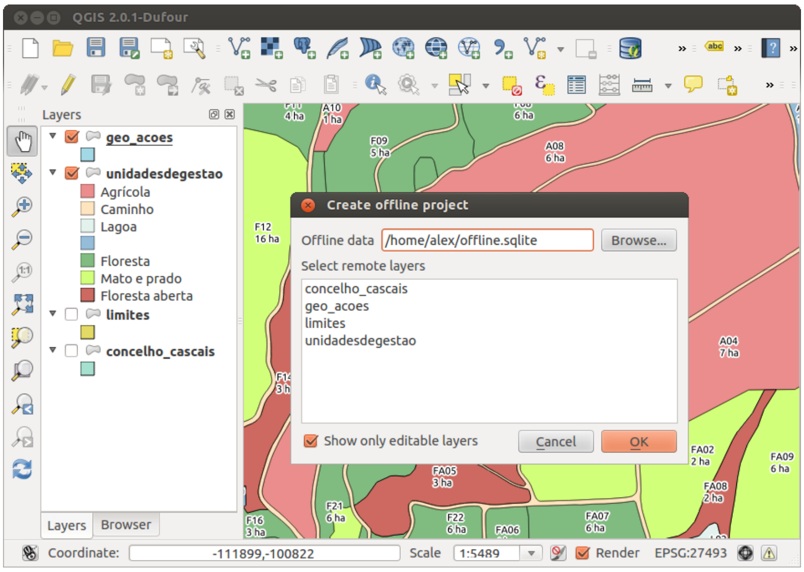


Figure 19.17: Create an offline project from PostGIS or WFS layers

19.12 Oracle GeoRaster Plugin

In Oracle databases, raster data can be stored in SDO_GEORASTER objects available with the Oracle Spatial extension. In QGIS, the OracleGeoRasterPlugin is supported by GDAL, and depends on Oracle's database product being installed and working on your machine. While Oracle is proprietary software, they provide their software free for development and testing purposes. Here is one simple example of how to load raster images to GeoRaster:

```
$ gdal_translate -of georaster input_file.tif geor:scott/tiger@orcl
```

This will load the raster into the default GDAL_IMPORT table, as a column named `RASTER`.

19.12.1 Managing connections

Firstly, the Oracle GeoRaster Plugin must be enabled using the Plugin Manager (see Section *Loading a QGIS Core Plugin*). The first time you load a GeoRaster in QGIS, you must create a connection to the Oracle database that contains the data. To do this, begin by clicking on the Select GeoRaster toolbar button, it will open the *Select Oracle Spatial GeoRaster* dialog window. Click on **[New]** to open the dialog window, and specify the connection parameters (See Figure_oracle_raster_1):

- **Name**: Enter a name for the database connection
- **Database instance**: Enter the name of the database that you will connect to
- **Username**: Specify your own username that you will use to access the database
- **Password**: The password associated with your username that is required to access the database

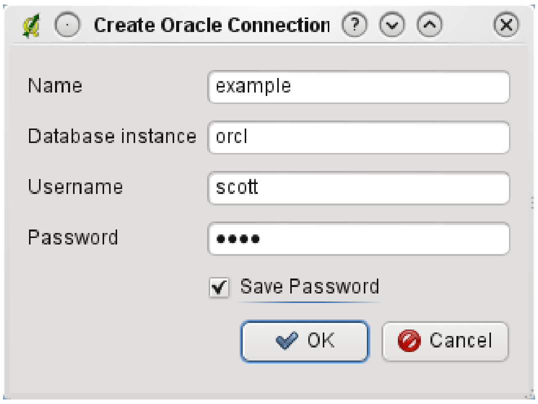


Figure 19.18: Create Oracle connection dialog

Now, back on the main *Oracle Spatial GeoRaster* dialog window (see Figure_oracle_raster_2), use the drop-down list to choose one connection, and use the **[Connect]** button to establish a connection. You may also **[Edit]** the connection by opening the previous dialog and making changes to the connection information, or use the **[Delete]** button to remove the connection from the drop-down list.

19.12.2 Selecting a GeoRaster

Once a connection has been established, the sub-datasets window will show the names of all the tables that contains GeoRaster columns in that database in the format of a GDAL subdataset name.

Click on one of the listed subdatasets and then click on **[Select]** to choose the table name. Now another list of subdatasets will show with the names of GeoRaster columns on that table. This is usually a short list, since most users will not have more than one or two GeoRaster columns on the same table.

Click on one of the listed subdatasets and then click on **[Select]** to choose one of the the table/column combination. The dialog will now show all the rows that contains GeoRaster objects. Note that the subdataset list will now show the Raster Data Table and Raster Id's pairs.

At anytime the Selection entry can be edited in order to go directly to a known GeoRaster or to go back to the beginning and select another table name.

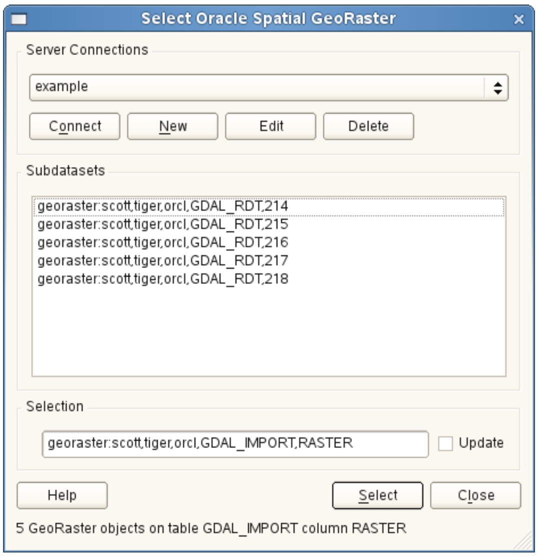


Figure 19.19: Select Oracle GeoRaster dialog

The Selection data entry can also be used to enter a `WHERE` clause at the end of the identification string, e.g. `geor:scott/tiger@orcl,gdal_import,raster,geoid=`. See http://www.gdal.org/frmt_georaster.html for more information.

19.12.3 Displaying GeoRaster

Finally, by selecting a GeoRaster from the list of Raster Data Table and Raster Id's, the raster image will be loaded into QGIS.

The *Select Oracle Spatial GeoRaster* dialog can be closed now and next time it opens it will keep the same connection, and will show the same previous list of subdataset making it very easy to open up another image from the same context.

Note: GeoRasters that contains pyramids will display much faster but the pyramids need to be generated outside of QGIS using Oracle PL/SQL or gdaladdo.

The following is example using `gdaladdo`:

```
gdaladdo georaster:scott/tiger@orcl,georaster\_table,georaster,georid=6 -r nearest 2 4 6 8 16 32
```

This is an example using 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;
```

19.13 Raster Terrain Analysis Plugin

The Raster Terrain Analysis Plugin can be used to calculate the slope, aspect, hillshade, ruggedness index and relief for digital elevation models (DEM). It is very simple to handle and provides an intuitive graphical user interface for creating new raster layers (See Figure_raster_terrain_1).

Description of the analysis:

- **Slope**: Calculates slope angle for each cell in degrees (based on first order derivative estimation).
- **Aspect**: Exposition (starting with 0 for north direction, in degrees counterclockwise).
- **Hillshade**: Create shaded map using light and shadow to provide a more three-dimensional appearance for a shaded relief map.
- **Ruggedness Index**: A quantitative measurement of terrain heterogeneity as described by Riley et al. (1999). It is calculated for every location, by summarizing the change in elevation within the 3x3 pixel grid.
- **Relief**: Creating a shaded relief map from digital elevation data. Implemented is a method to choose the elevation colors analysing the frequency distribution.

19.13.1 Using the plugin

1. Start QGIS and load the `gtopo30` raster layer from the GRASS sample location.
2. Load the Raster Terrain Analysis plugin in the Plugin Manager (see Section *Loading a QGIS Core Plugin*)
3. Select an analysis method from menu (e.g. *Raster* → *Terrain Analysis* → *Slope*). The *Slope* dialog appears as shown in Figure_raster_terrain_1.

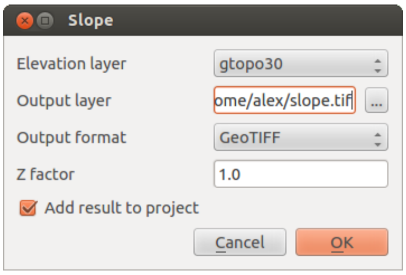


Figure 19.20: Raster Terrain Modelling Plugin (slope calculation)

4. Specify an output file path, and an output file type.
5. Click **[OK]**.

19.14 Heatmap Plugin

The *Heatmap* plugin uses Kernel Density Estimation to create a density (heatmap) raster of an input point vector layer. The density is calculated based on the number of points in a location, with larger numbers of clustered points resulting in larger values. Heatmaps allow easy identification of "hotspots" and clustering of points.

19.14.1 Activate the Heatmap plugin

First this core plugin needs to be activated using the Plugin Manager (see Section *Loading a QGIS Core Plugin*).

After activation the heatmap icon can be found in the Raster Toolbar, and under the *Raster* → *Heatmap* menu.

Select the menu *View* → *Toolbars* → *Raster* to show the Raster Toolbar if it is not visible.

19.14.2 Using the Heatmap plugin

Clicking the *Heatmap* toolbutton opens the Heatmap plugin dialog (see figure_heatmap_2).

The dialog has the following options:

- **Input point layer**: lists all the vector point layers in the current project and is used to select the layer to be analysed.
- **Output raster**: use the ... button to select the folder and file name for the output raster the Heatmap plugin generates. A file extension is not required.
- **Output format**: selects the output format. Although all formats supported by GDAL can be choosen, in most cases GeoTIFF is the best format to choose.
- **Radius**: used to specify the heatmap search radius (or kernel bandwidth) in meters or map units. The radius specifies the distance around a point at which the influence of the point will be felt. Larger values result in greater smoothing, but smaller values may show finer details and variation in point density.

When the *Advanced* checkbox is checked additional options will be available:

- **Rows** and **Columns**: used to change the dimensions of the output raster. These values are also linked to the **Cell size X** and **Cell size Y** values. Increasing the number of rows or colums will decrease the cell size and increase the file size of the output file. The values in Rows and Columns are also linked, so doubling the number of rows will automatically double the number of columns and the cell sizes will also be halved. The geographical area of the output raster will remain the same!
- **Cell size X** and **Cell size Y**: control the geographic size of each pixel in the output raster. Changing these values will also change the number of Rows and Columns in the output raster.

- **Kernel shape**: The kernel shape controls the rate at which the influence of a point decreases as the distance from the point increases. Different kernels decay at different rates, so a triweight kernel gives features greater weight for distances closer to the point then the Epanechnikov kernel does. Consequently, triweight results in "sharper" hotspots, and Epanechnikov results in "smoother" hotspots. A number of standard kernel functions are available in QGIS, which are described and illustrated on Wikipedia.
- **Decay ratio**: can be used with Triangular kernels to futher control how heat from a feature decreases with distance from the feature.
  - A value of 0 (=minimum) indicates that the heat will be concentrated in the centre of the given radius and be completely extinguished at the edge.
  - A value of 0.5 indicates that pixels at the edge of the radius will be given half the heat as pixels at the centre of the search radius.
  - A value of 1 means the heat is spread evenly over the whole search radius circle. (This is equivalent to the 'Uniform' kernel.)
  - A value greater than 1 indicates that the heat is higher towards the edge of the search radius then at the centre.

The input point layer may also have attribute fields which can affect how they influence the heatmap:

- **Use radius from field**: sets the search radius for each feature from an attribute field in the input layer.
- **Use weight from field**: allows input features to be weighted by an attribute field. This can be used to increase the influence certain features have on the resultant heatmap.

When an output raster file name is specified, the **[OK]** button can be used to create the heatmap.

19.14.3 Tutorial: Creating a Heatmap

For the following example we will use the `airports` vector point layer from the QGIS sample dataset (see *Sample Data*). Another exellent QGIS tutorial on making heatmaps can be found at http://qgis.spatialthoughts.com.

In Figure_Heatmap_1 the airports of Alaska are shown.

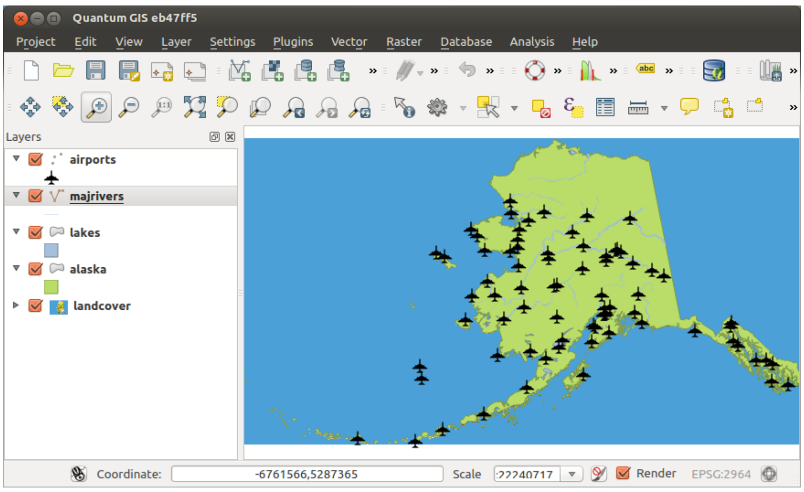


Figure 19.21: Airports of Alaska

1. Select the *Heatmap* toolbutton to open the Heatmap dialog (see Figure_Heatmap_2).
2. In the *Input point layer* field select `airports` from the list of point layers loaded in current project.
3. Specify an output file name by clicking the ... button next to the *Output raster* field. Enter the file name `heatmap_airports` (no file extension is necessary).

4. Leave the *Output format* as the default format, `GeoTIFF`.
5. Change the *Radius* to `1000000` meters.
6. Click on **[OK]** to create and load the airports heatmap (see Figure_Heatmap_3).

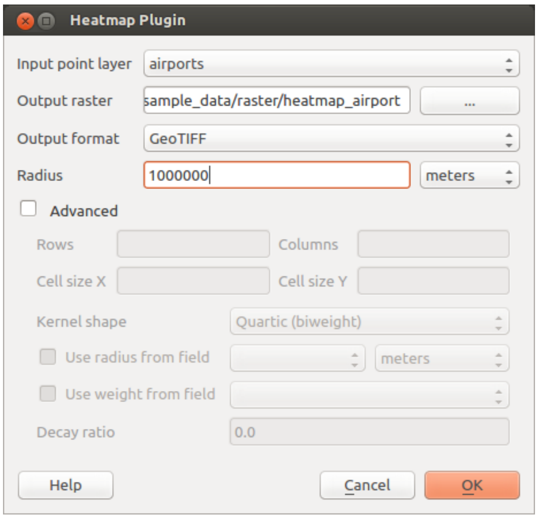


Figure 19.22: The Heatmap Dialog

QGIS will generate the heatmap and add the results to your map window. By default, the heatmap is shaded in greyscale, with lighter areas showing higher concentrations of airports. The heatmap can now be styled in QGIS to improve its appearance.

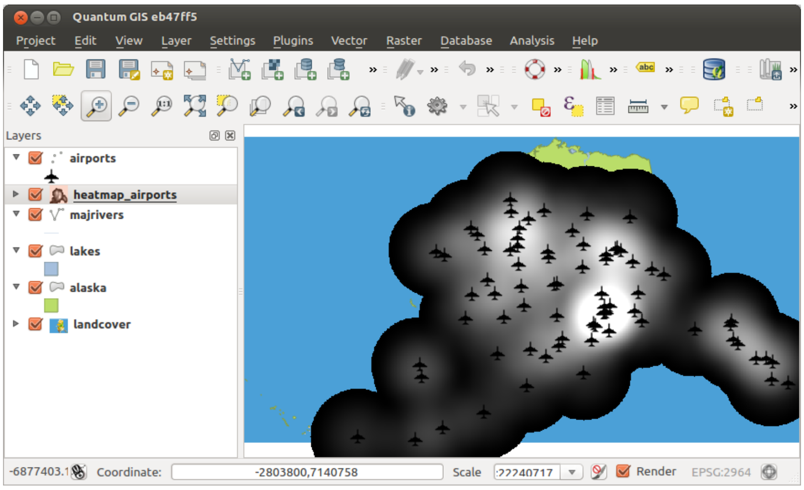


Figure 19.23: The heatmap after loading looks like a grey surface

1. Open the properties dialog of the `heatmap_airports` layer (select the layer `heatmap_airports`, open the context menu with the right mouse button and select *Properties*).
2. Select the *Style* tab.
3. Change the *Render type* to 'Singleband pseudocolor'.
4. Select a suitable *Color map*, for instance `YlOrRed`.
5. Click the **[Load]** button to fetch the minimum and maximum values from the raster, then click the **[Classify]** button.
6. Press **[OK]** to update the layer.

The final result is shown in Figure_Heatmap_4.

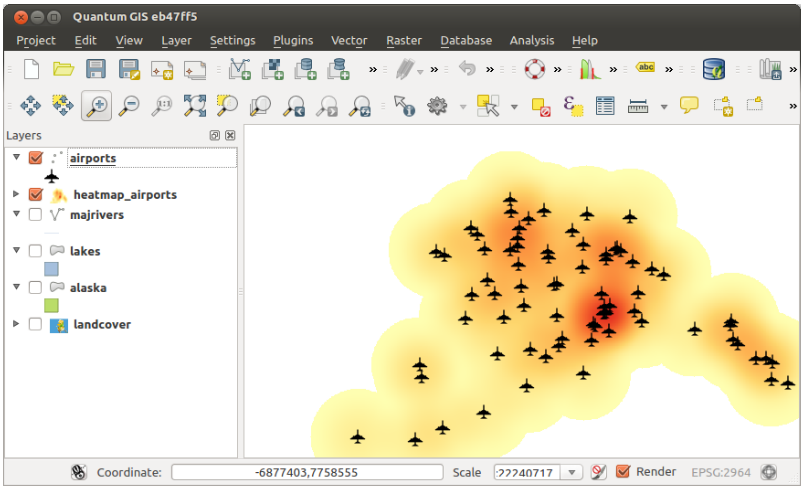


Figure 19.24: Styled heatmap of airports of Alaska

19.15 Road Graph Plugin

The Road Graph Plugin is a C++ plugin for QGIS, that calculates the shortest path between two points on any polyline layer and plots this path over the road network.

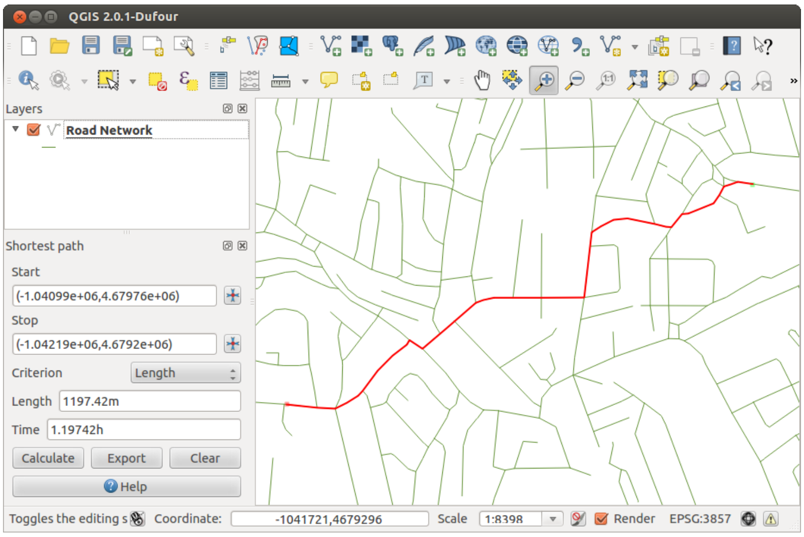


Figure 19.25: Road Graph Plugin

Main features:

- calculate path, it's length and travel time
- optimize by length or by travel time
- export path to a vector layer
- highlight roads directions (this is slow and used mainly for debug purposes and for the settings testing)

As a roads layer you can use any polyline vector layer in any QGIS supported format. Two lines with a common point are considered connected. Please note, it is required to use layer CRS as project CRS while editing roads layer. This is due to the fact that recalculation of the coordinates between different CRS introduce some errors that can result in discontinuities, even when 'snapping' is used.

In the layer attribute table the following fields can be used:

- speed on road section — numeric field;
- direction — any type, that can be casted to string. Forward and reverse directions are correspond to the one-way road, both directions — two-way road

If some fields don't have any value or do not exist — default values are used. You can change defaults and some plugin settings in plugin settings dialog.

19.15.1 Using the plugin

After plugin activation you will see an additional panel on the left side of the main QGIS window. Now make some definitions to the *Road graph plugin settings* dialog in the menu *Vector* → *Road Graph* (see figure_road_graph_2).

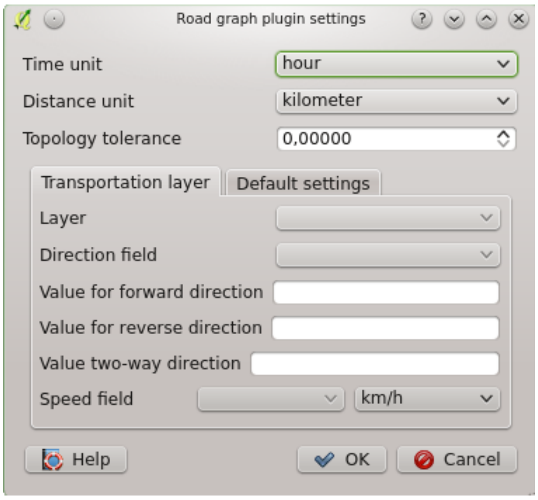


Figure 19.26: Road graph plugin settings

After setting the *Time unit*, *Distance unit* and *Topology tolerance* you can choose the vector layer in the *Transportation layer* tab. Here you can also choose the *Direction field* and *Speed field*. In the *Default settings* tab you can set the *Direction* for the calculation.

Finally in the *Shortest Path* panel select a Start and a Stop point in the road network layer and click on **[Calculate]**.

19.16 Spatial Query Plugin

The Spatial Query plugin allows to make a spatial query (select features) in a target layer with reference to another layer. The functionality is based on the GEOS library and depends on the selected source feature layer.

Possible operators are:

- Contains
- Equals
- Overlap
- Crosses
- Intersects
- Is disjoint
- Touches
- Within

19.16.1 Using the plugin

As an example we want to find regions in the Alaska dataset that contain airports. Following steps are necessary:

1. Start QGIS and load the vector layers `regions.shp` and `airports.shp`.
2. Load the Spatial Query plugin in the Plugin Manager (see Section *Loading a QGIS Core Plugin*) and click on the Spatial Query icon which appears in the QGIS toolbar menu. The plugin dialog appears.
3. Select layer `regions` as source layer and `airports` as reference feature layer.
4. Select 'Contains' as operator and click **[Apply]**.

Now you get a list of feature IDs from the query and you have several options as shown in figure_spatial_query_1.

- Click on the Create layer with list of items
- Select an ID from the list and click on Create layer with selected
- Select the 'Remove from current selection' in the field *And use the result to* .
- Additionally you can *Zoom to item* or display *Log messages*.

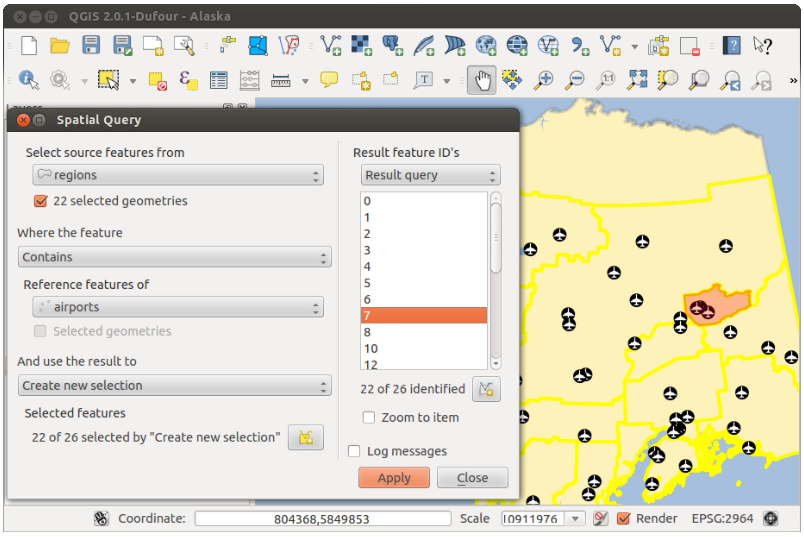


Figure 19.27: Spatial Query analysis - regions contain airports

19.17 SQL Anywhere Plugin

SQL Anywhere is a proprietary relational database management system (RDBMS) from Sybase. SQL Anywhere includes spatial support including OGC, shape files etc. and built in functions to export to KML, GML and SVG formats.

The SQL Anywhere allows to connect to spatially enabled SQL Anywhere databases. The *Add SQL Anywhere layer* dialog is similar in functionality to the dialogs for PostGIS and SpatiaLite.

19.18 Topology Checker Plugin

Topology describes the relationships between points, lines and polygons that represent the features of a geographic region. With the Topology Checker plugin you can look over your vector files and check the topology with several topology rules. These rules check with spatial relations whether your features 'Equal', 'Contain', 'Cover', are 'CoveredBy', 'Cross', 'Disjoint', 'Intersect', 'Overlap', 'Touches' and are 'Within' each other. It depends on

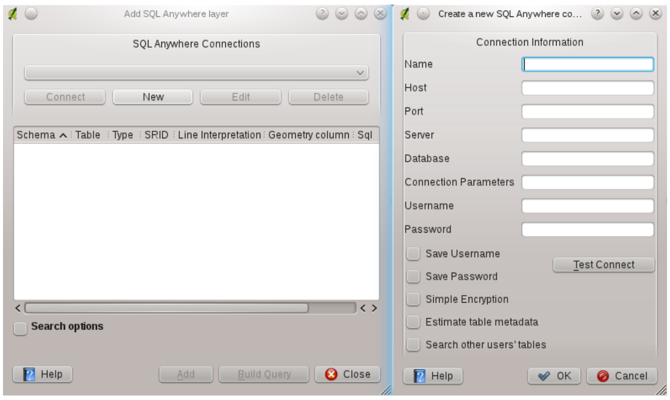


Figure 19.28: SQL Anywhere dialog (KDE)

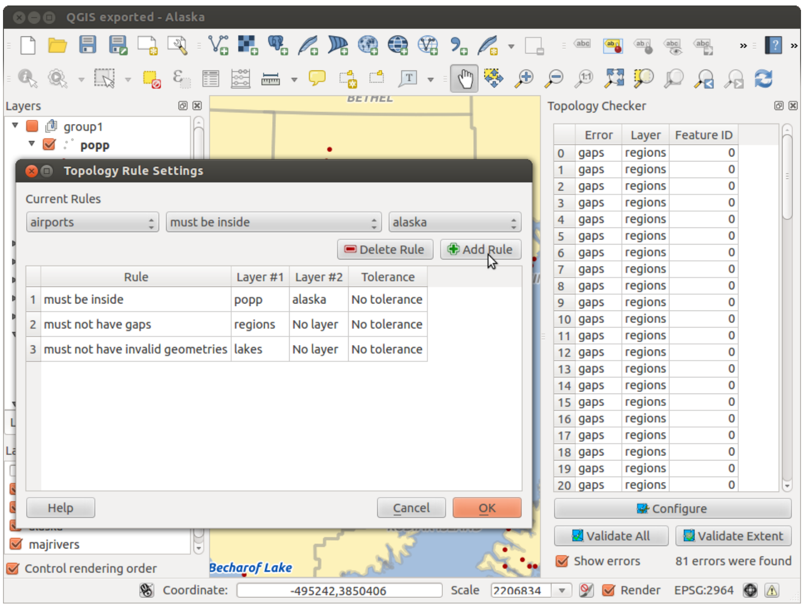


Figure 19.29: The Topology Checker Plugin

your individual questions which topology rules you apply to your vector data. E.g. normally you won't accept overshoots in line layers but if they depict dead-end streets you won't remove them from your vector layer.

QGIS has the built-in topological editing feature which is great for creating new features without errors. But existing data errors and user induced errors are hard to find out. This plugin helps you find out such errors through a list of rules.

It is very simple to create topology rules with the Topology Checker plugin.

On **point layers** the following rules are available:

- **must be covered by**: Here you can choose a vector layer from your project. Points that aren't covered by the given vector layer occur in the 'Error' field
- **must be covered by endpoints of**: Here you can choose a line layer from your project.
- **must be inside**: Here you can choose a polygon layer from your project. The points must be inside a polygon. Otherwise QGIS writes an 'Error' for the point.
- **must not have duplicates**: Whenever a point is represented twice or more it will occur in the 'Error' field.
- **must not have invalid geometries**: Checks whether the geometries are valid.
- **must not have multi-part-geometries**: All multi-part points are written into the 'Error' field.

On **line layers** the following rules are available:

- **end points must be covered by**: Here you can select a point layer from your project.
- **must not have dangles**: This will show the overshoots in the line layer.
- **must not have duplicates**: Whenever a line feature is represented twice or more it will occur in the 'Error' field.
- **must not have invalid geometries**: Checks whether the geometries are valid.
- **must not have multi-part geometries**: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. All multi-part lines are written into the 'Error' field.
- **must not have pseudos**: A line geometry's endpoint should be connected to the endpoints of two other geometries. If the endpoint is connected to only one other geometry's endpoint, the endpoint is called a psuedo node.

On **polygon layers** the following rules are available:

- **must contain**: Polygon layer must contain at least one point geometry from the second layer.
- **must not have duplicates**: Polygons from the same layer must not have identical geometries. Whenever a polygon feature is represented twice or more it will occur in the 'Error' field.
- **must not have gaps**: Adjacent polygons should not form gaps between them. Administrative boundaries could be mentioned as an example (US state polygons do not have any gaps between them...).
- **must not have invalid geometries**: Checks whether the geometries are valid. Some of the rules that define a valid geometry are,
  - Polygon rings must close.
  - Rings that define holes should be inside rings that define exterior boundaries.
  - Rings may not self-intersect (they may neither touch nor cross one another).
  - Rings may not touch other rings, except at a point.
- **must not have multi-part geometries**: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. For example, a country consisting of multiple islands can be represented as a multi-polygon.
- **must not overlap**: Adjacent polygons should not share common area.

- **must not overlap with**: Adjacent polygons from one layer should not share common area with polygons from another layer.

19.19 Zonal Statistics Plugin

With the Zonal Statistics Plugin you can analyze the results of a thematic classification. It allows to calculate several values of the pixels of a raster layer with the help of a polygonal vector layer (see figure_zonal_statistics). You can calculate the sum, the mean value and the total count of the pixels that are within a polygon. The plugin generates output columns in the vector layer with a user-defined prefix.

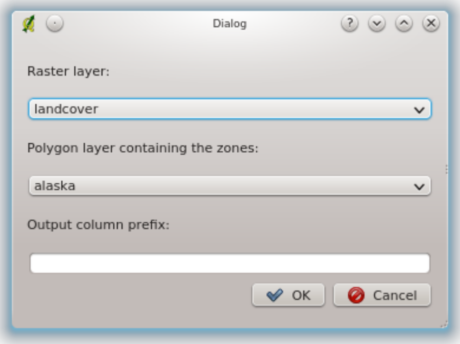


Figure 19.30: Zonal statistics dialog (KDE)

CHAPTER 20

Help and Support

20.1 Mailing lists

QGIS is under active development and as such it won't always work like you expect it to. The preferred way to get help is by joining the qgis-users mailing list. Your questions will reach a broader audience and answers will benefit others.

20.1.1 qgis-users

This mailing list is used for discussion of QGIS in general, as well as specific questions regarding its installation and use. You can subscribe to the qgis-users mailing list by visiting the following URL: http://lists.osgeo.org/mailman/listinfo/qgis-user

20.1.2 fossgis-talk-liste

For the german speaking audience the german FOSSGIS e.V. provides the fossgis-talk-liste mailing list. This mailing list is used for discussion of open source GIS in general including QGIS. You can subscribe to the fossgis-talk-liste mailing list by visiting the following URL: https://lists.fossgis.de/mailman/listinfo/fossgis-talk-liste

20.1.3 qgis-developer

If you are a developer facing problems of a more technical nature, you may want to join the qgis-developer mailing list here: http://lists.osgeo.org/mailman/listinfo/qgis-developer

20.1.4 qgis-commit

Each time a commit is made to the QGIS code repository an email is posted to this list. If you want to be up to date with every change to the current code base, you can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-commit

20.1.5 qgis-trac

This list provides email notification related to project management, including bug reports, tasks, and feature requests. You can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-trac

20.1.6 qgis-community-team

This list deals with topics like documentation, context help, user-guide, online experience including web sites, blog, mailing lists, forums, and translation efforts. If you like to work on the user-guide as well, this list is a good starting point to ask your questions. You can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-community-team

20.1.7 qgis-release-team

This list deals with topics like the release process, packaging binaries for various OS and announcing new releases to the world at large. You can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-release-team

20.1.8 qgis-tr

This list deals with the translation efforts. If you like to work on the translation of the manuals or the graphical user interface (GUI), this list is a good starting point to ask your questions. You can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-tr

20.1.9 qgis-edu

This list deals with QGIS education efforts. If you like to work on QGIS education materials, this list is a good starting point to ask your questions. You can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-edu

20.1.10 qgis-psc

This list is used to discuss Steering Committee issues related to overall management and direction of QGIS. You can subscribe to this list at: http://lists.osgeo.org/mailman/listinfo/qgis-psc

You are welcome to subscribe to any of the lists. Please remember to contribute to the list by answering questions and sharing your experiences. Note that the qgis-commit and qgis-trac are designed for notification only and not meant for user postings.

20.2 IRC

We also maintain a presence on IRC - visit us by joining the #qgis channel on irc.freenode.net. Please wait around for a response to your question as many folks on the channel are doing other things and it may take a while for them to notice your question. Commercial support for QGIS is also available. Check the website http://qgis.org/en/commercial-support.html for more information.

If you missed a discussion on IRC, not a problem! We log all discussion so you can easily catch up. Just go to http://qgis.org/irclogs and read the IRC-logs.

20.3 BugTracker

While the qgis-users mailing list is useful for general 'how do I do XYZ in QGIS' type questions, you may wish to notify us about bugs in QGIS. You can submit bug reports using the QGIS bug tracker at http://hub.qgis.org/projects/quantum-gis/issues. When creating a new ticket for a bug, please provide an email address where we can request additional information.

Please bear in mind that your bug may not always enjoy the priority you might hope for (depending on its severity). Some bugs may require significant developer effort to remedy and the manpower is not always available for this.

Feature requests can be submitted as well using the same ticket system as for bugs. Please make sure to select the type `Feature`.

If you have found a bug and fixed it yourself you can submit this patch also. Again, the lovely redmine ticketsystem at http://hub.qgis.org/wiki/quantum-gis/issues has this type as well. Check `Patch supplied` checkbox and attach your patch before submitting bug. Someone of the developers will review it and apply it to QGIS. Please don't be alarmed if your patch is not applied straight away — developers may be tied up with other committments.

20.4 Blog

The QGIS community also runs a weblog at http://planet.qgis.org/planet/ which has some interesting articles for users and developers as well provided by other blogs in the community. You are invited to contribute your own QGIS blog!

20.5 Plugins

The website http://plugins.qgis.org provides the official QGIS plugins web portal. Here you find a list of all stable and experimental QGIS plugins available via the 'Official QGIS Plugin Repository'.

20.6 Wiki

Lastly, we maintain a WIKI web site at http://hub.qgis.org/projects/quantum-gis/wiki where you can find a variety of useful information relating to QGIS development, release plans, links to download sites, message translation-hints and so on. Check it out, there are some goodies inside!

CHAPTER 21

Appendix

21.1 GNU General Public License

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to

any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

(c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

QGIS Qt exception for GPL

In addition, as a special exception, the QGIS Development Team gives permission to link the code of this program with the Qt library, including but not limited to the following versions (both free and commercial): Qt/Non-commerical Windows, Qt/Windows, Qt/X11, Qt/Mac, and Qt/Embedded (or with modified versions of Qt that use the same license as Qt), and distribute linked combinations including the two. You must obey the GNU General Public License in all respects for all of the code used other than Qt. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

21.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **Document**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **Opaque**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".)

To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

3. State on the Title page the name of the publisher of the Modified Version, as the publisher.

4. Preserve all the copyright notices of the Document.

5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
8. Include an unaltered copy of this License.
9. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
11. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
13. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
14. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
15. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

CHAPTER 22

Literature and Web References

GDAL-SOFTWARE-SUITE. Geospatial data abstraction library. http://www.gdal.org, 2013.

GRASS-PROJECT. Geographic ressource 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. Spatial support for postgresql. http://postgis.refractions.net/ , 2013.

Index