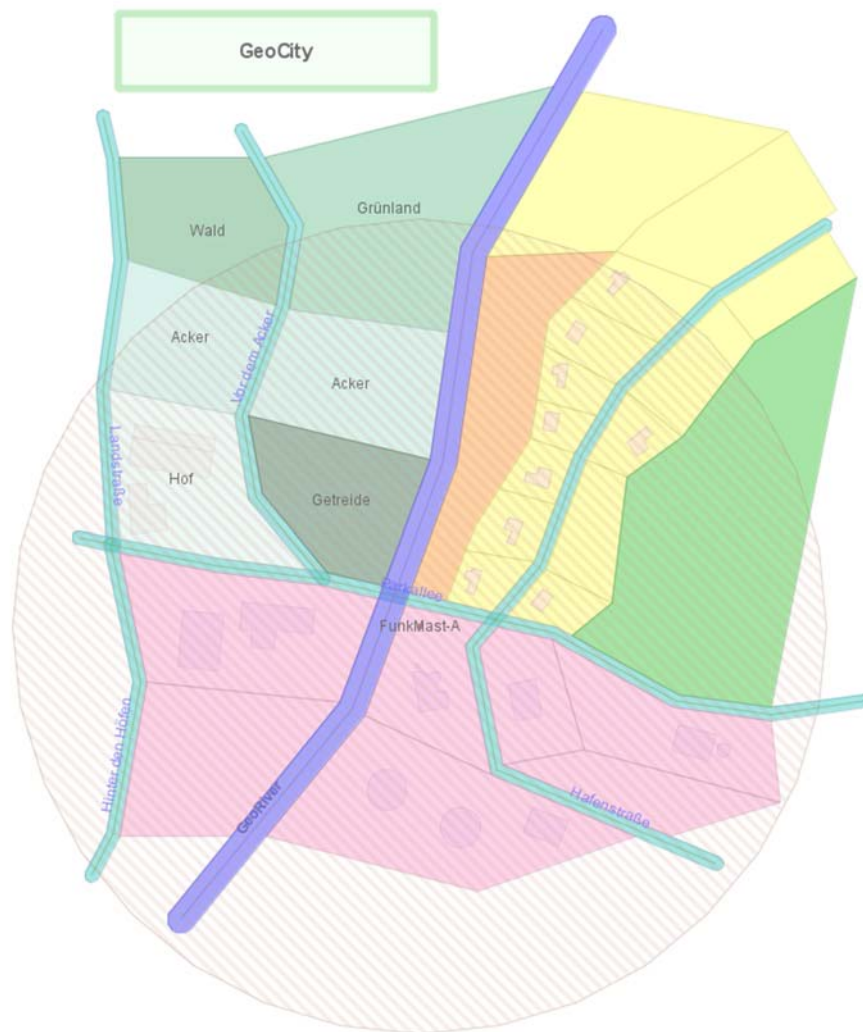


OpenJUMP Tutorial (basics)



Based on German Version 3/2006

by Uwe Dalluege

HCU Hamburg

Dept. Geomatik

author:

Uwe Dalluege
HafenCity Universität Hamburg
Department Geomatik
Hebebrandstr. 1
22297 Hamburg
E-Mail: uwe.dalluege@rzcn.haw-hamburg.de

translation:

Stefan Steiniger
Jump Pilot Project

jump-pilot-devel@lists.sourceforge.net

terms of usage:

This text is free; you can redistribute it and/or modify it under the terms of **GNU Free Documentation License** (<http://www.gnu.org/licenses/fdl.txt>).

With support of (in alphabetical order):

Dipl.-Ing. Jürgen Faber, HCU Hamburg
Dipl.-Ing. Kay Zobel, HCU Hamburg
Dipl.-Ing. Stefan Steiniger, University of Zurich

Content

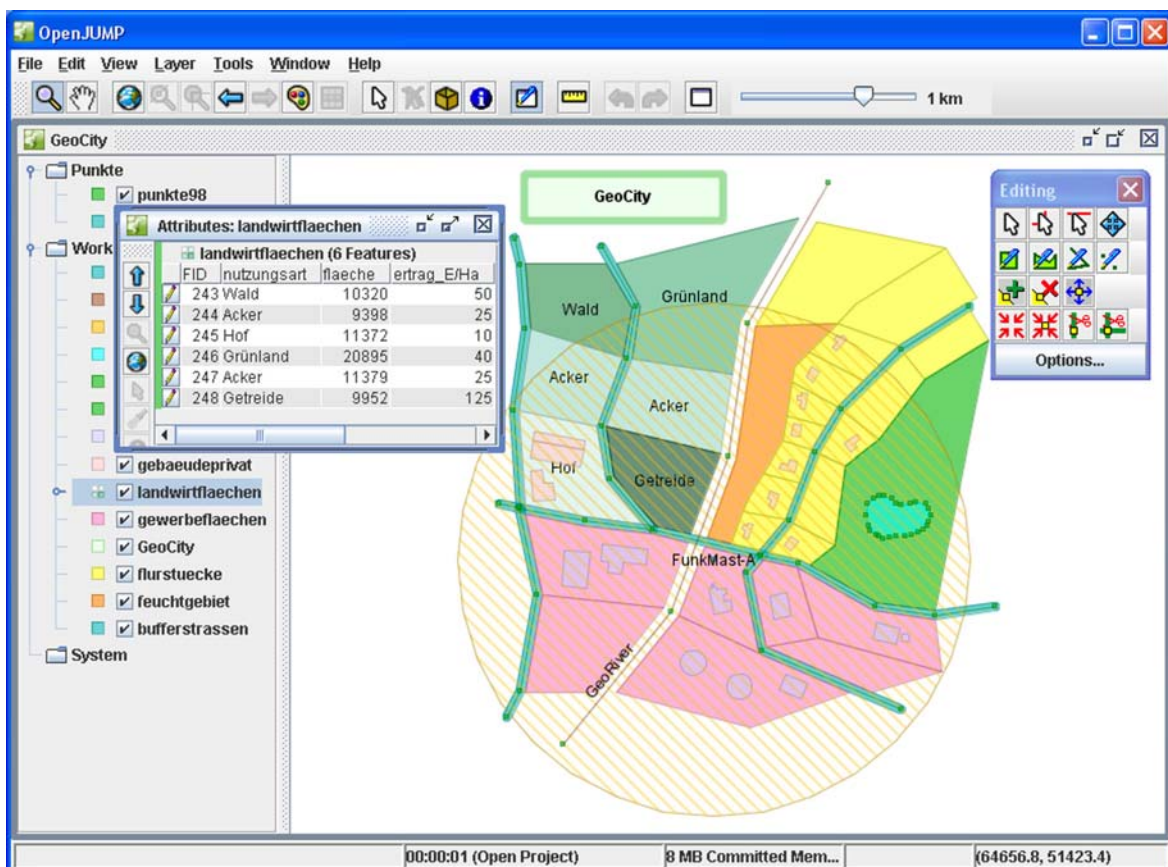
1	Before The Journey Begins	4
2	User Interface of OpenJUMP	5
2.1	The Toolbar	6
3	Data	7
3.1	Geometry data	7
3.2	Attribute values	7
4	The Editing Toolbox	8
4.1	Editing Toolbox	9
4.1.1	Options (Snap/Grid)	9
4.2	Features	10
4.2.1	Drawing Features	10
4.2.1.1	Drawing an area (Polygon):	10
4.2.1.2	Drawing a „hole“ in a polygon:	10
4.2.2	Combining/Grouping Features (Geometry Collection)	11
4.2.3	Resolving/Ungrouping Features	11
4.2.4	Selecting and Moving Features	11
4.3	Editing line points (vertex, vertices)	12
4.4	Splitting Lines (LineStrings)	12
5	Editing Layers	13
5.1	Layer	13
5.1.1	Layer Properties	13
5.1.2	Saving Layers and Projects	15
5.2	WMS-Layer	16
6	Schema editing	18
7	Editing Attributes	19
8	Layer display (Styles)	20
8.1	Normal Styles (Rendering)	20
8.2	Colour Theming	21
8.3	Labeling	23
8.4	Start- and End Point Display (Decorations)	24
9	Analysis Tools	25
9.1	Buffer	25
9.2	Overlay	26
9.3	Union	27
9.4	Geometry Functions	28
10	Connection to a PostgreSQL/PostGIS database	30
10.1	Reading Tables	30
10.2	Writing Tables	31
11	Printing	32
12	Glossary	33
13	Literature References	37
14	Link Collection	39
15	Index	40

1 Before The Journey Begins

OpenJUMP is a **Geographical Information System (GIS)**, which has been developed originally by the two Canadian companies **Vivid Solutions** und **Refractions Research** under the name **JUMP**. The name JUMP is an abbreviation for Unified Mapping Platform. The „J“ points to the used programming language „**Java**“. „Open“ is for „**Open Source**“, which means that the quell code is accessible for everybody. OpenJUMP underlies the **GNU General Public License** and is maintained and improved by programmers around the globe.

The main **features** of OpenJUMP are:

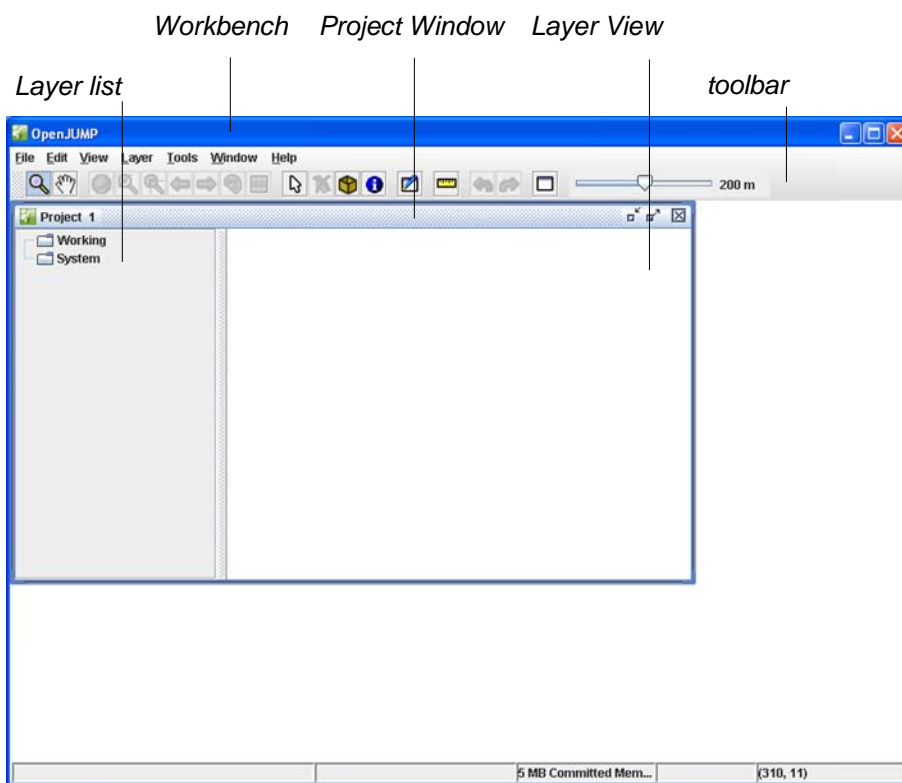
- vector-based GIS
- underlies the GNU General Public License <http://www.gnu.org/licenses/licenses.html#GPL>
- is based on Open GIS standards <http://www.opengeospatial.org/>
- written in Java and open source (accessible programming code)
- can be easily extended by PlugIns
- easy to use editing and analysis functions
- supports several languages (English, French, Portuguese, Spanish, German)



GeoCity in OpenJUMP

2 User Interface of OpenJUMP

After starting OpenJUMP the OpenJUMP Workbench is shown with an empty project window. This window contains a list of layers and a graphic display, the Layer View, in which the layers are displayed. **Layers** are used to display geographic objects (**features**). A Layer usually contains objects of specific geographic theme (e.g. water bodies, forest, buildings, soils). A **Project** contains a collection of all layers.



After starting OpenJUMP




















One can open more than one **project window** at the same time. With the menu option „window“ one can switch between the project windows. A new project will be created with **„File>NewProject...“**. An existing project can be opened using the menu command **„File>OpenProject...“**.

Note: Every layer has to be saved in a **file** or **database** first, if this has been done one can save the **project** (see „Layer“, page 13).

2.1 The Toolbar

The Toolbar enables to have fast access to functions like zooming, panning, selection or to activate the graphic editing tools.



 Zooming	 Panning
 Show all data	 Zoom to selected item
 Zoom to fence	
 Previous view	 Next view
 Change layer visualisation	 Display attributes (also for modification)
 Select items	 Clear selection
 Draw a fence	 Show feature information
 Editing toolbox	 Measure
 Undo	 Redo
 Output Window	 200 m Fast Zooming

3 Data

An important part of a GIS are the data, consisting usually of geometry data and attribute data. The question is: How can we add these data to our project.

3.1 Geometry data

There exists several methods to add geometric data to a project:

1. By making some freehand drawings with the graphic editing tools. This is a very simple method is not very common with out any original to digitize (see „**The Editing Toolbox**“, page 8)
2. By importing data, stored in a specific file format. OpenJUMP supports the following formats:
 - a) JUMP GML format (.jml), which is a simplified GML-format.
 - b) GML 2.0 format.
 - c) WKT (Well Known Text) format.
 - d) ESRI Shapefile.

Using „**File>Load Dataset(s) ...**“

3. By displaying image raster data, which can be received from a WMS-Server. Here, one can digitize the data from the images with the editing tools.
4. By loading a database table from PostGIS (only with additional external plugins). Thereby the data (geometries and attributes) are stored in the object relational database PostgreSQL with a so called spatial extension PostGIS and are transformed into the OpenJUMP format to display the data.
5. Note: There exists several external plugins to load raster image data from files (further info can be obtained from www.openjump.org)

3.2 Attribute values


There exists several methods to add attribute data to a project::

1. Through direct input in OpenJUMP. Therefore the geometry data must exist already (see also under Schema editing *p. 18*)
2. By importing the data from files in a specific format. OpenJUMP supports the following file formats, which also may contain geometry data:
 - a) JUMP GML format (.jml), which is a simplified GML-format.
 - b) GML 2.0 format.
 - c) ESRI Shapefile (dbf files).
3. By loading a database table from PostGIS (only with additional external plugins).

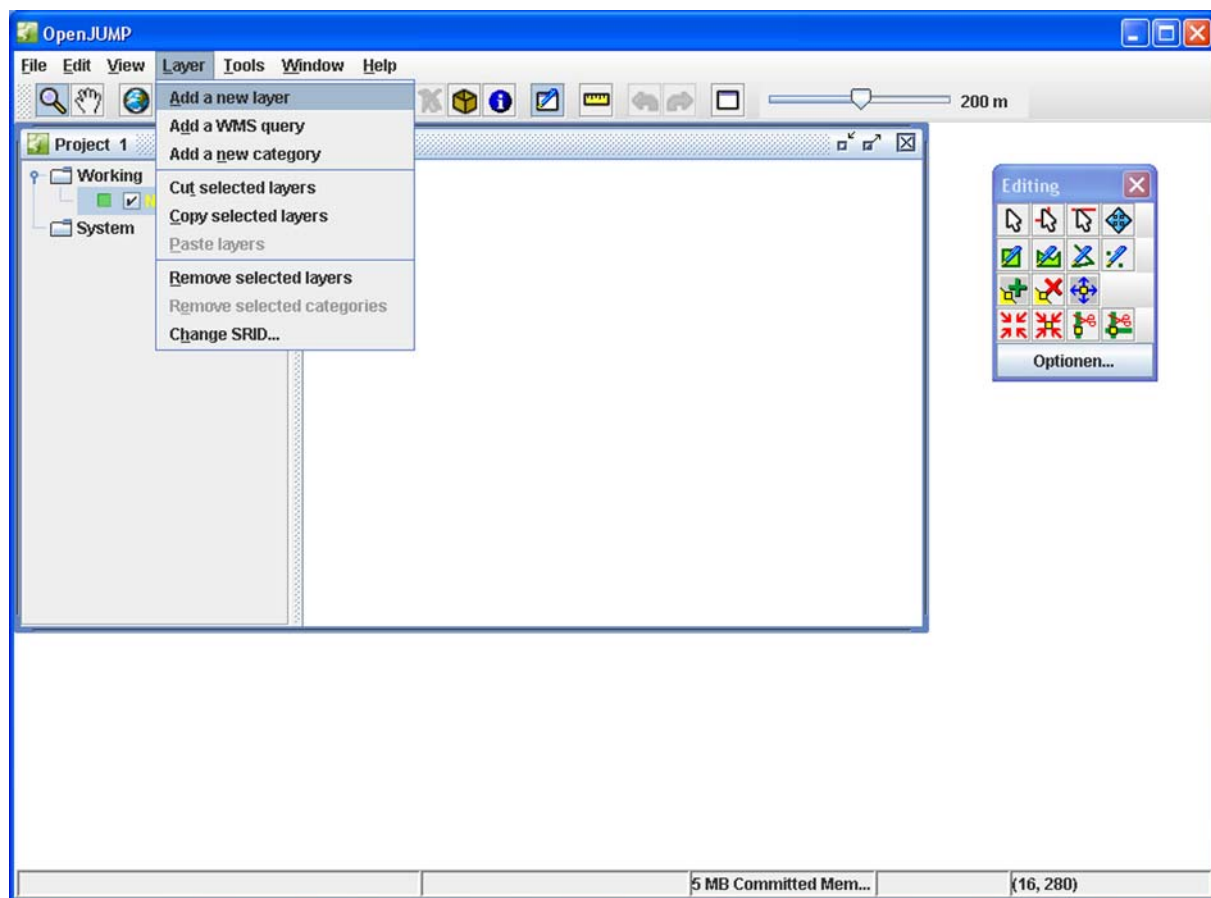
4 The Editing Toolbox

In OpenJUMP exists an editing toolbox to create new or modify geometries of geographic objects (**features**).

One can create and edit the following geometries: **points**, **lines** (Linestring) and **polygons** (Polygon).

The toolbox can be activated with the button  from the Tool Bar.

New features can be drawn in a new layer, which can be created by using „**Layer>Add a new layer**“. New features can also be drawn in existing layers, if they are made „editable“ (see layer mouse menu).



Layer>Add a new layer

A new created layer is the name „New“ assigned and is set to editable by default. Is a layer editable, then it's name is drawn in red. If one selects an editable layer, the mouse the name will be displayed in yellow and new geometries are drawn in that layer. If Geometries should be modified the layer must be editable as well. The mouse menu the option „Editable“ (Right click on layer name) sets the editing status of the selected layer. After double-left-click on the layer name, the name can be changed.

4.1 Editing Toolbox

The **Editing Toolbox** contains tools for modifying and generating new geometries. The most important ones are:



Selection and Move tools

Rectangle, Polygon, Line and Point drawing tools

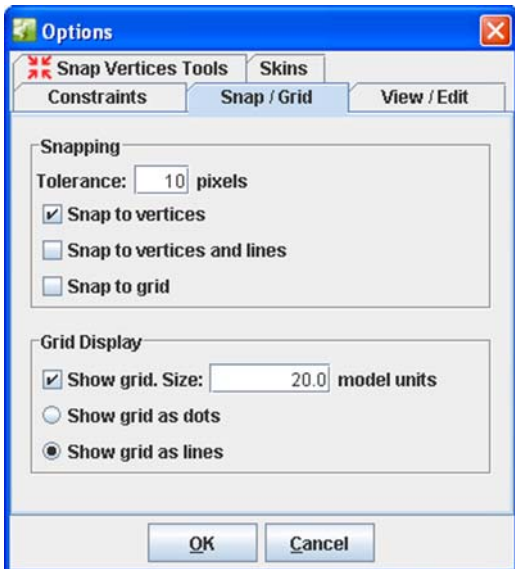
Add, delete and move a vertex (line point)

Snap vertices; split lines

Editing Toolbox

4.1.1 Options (Snap/Grid)

In the Options window two supporting tools can be set: the Snapping Mode and Grid Display. Default snapping tolerance is 10 pixels and „Snap to vertices“.



snapping and grid display options

4.2 Features

Features are abstracted/generalized objects of the real world. For instance roads are abstracted and displayed as lines, buildings as polygons and trees as points.

In OpenJUMP every feature consists of a spatial attribute (geometry) and non or several non-spatial attributes (fields, properties) e.g. road name, owner, tree height.


4.2.1 Drawing Features

Three different features / geometry types can be drawn:



Point , **Linestring**  and **Areas (Polygon)**  

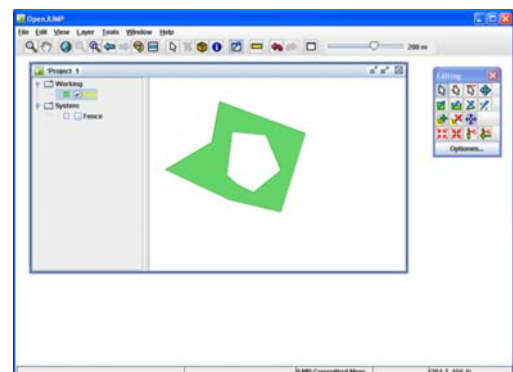
After selection of the geometry type from the editing toolbox, the geometry is drawn with left-mouse clicks. Linestrings and polygons are finished with a double click on the left mouse button.

4.2.1.1 Drawing an area (Polygon):

- Select the layer and make him editable.
- Press „Draw Polygon Tool“ 
- Mark the start point with a left-mouse click in the layer view.
- Add further vertices (line-points) with left clicks.
- Close the area with a double left click.

4.2.1.2 Drawing a „hole“ in a polygon:

- Select the layer and make him editable
- select the area with 
- Press „Draw Polygon Tool“ 
- draw the new polygon/hole on top of the other.




polygon with hole


4.2.2 Combining/Grouping Features (Geometry Collection)

Several geometries can be combined to one „logical“ entity. Therefore the features have to be all in the same layer.

Note: If features are combined, non-spatial attributes can be lost.


- Select the features, which should be combined with **Shift +** 
- Right-mouse click on the layer view
- select in the mouse menu: „**Combine selected features**“

4.2.3 Resolving/Ungrouping Features

- Select the Geometry Collection with 
- Right-mouse click on the layer view
- select in the mouse menu: „**Explode selected features**“

4.2.4 Selecting and Moving Features

OpenJUMP has three selection tools:

 select a complete feature.

 select a part of Geometry Collection

 select a **hole**

 move selected features

4.3 Editing line points (vertex, vertices)



Adding a line point (vertex) to a feature

- select the feature
- add the vertex with left-mouse click on the line segment



Delete vertex

- select feature(s)
- delete the vertex by clicking on it



Move vertex

- select feature
- move the vertex by left-clicking on it and dragging the mouse



Merge vertices

- select the first node and drag, while still pressing the button, to the second node



Merge two selected vertices

- select feature (the function works only with one feature)
- select the tool and draw a box around both vertices
- press the shift-key and click on the target vertex in the box

4.4 Splitting Lines (LineStrings)



Split line

- select feature
- left click on the split position



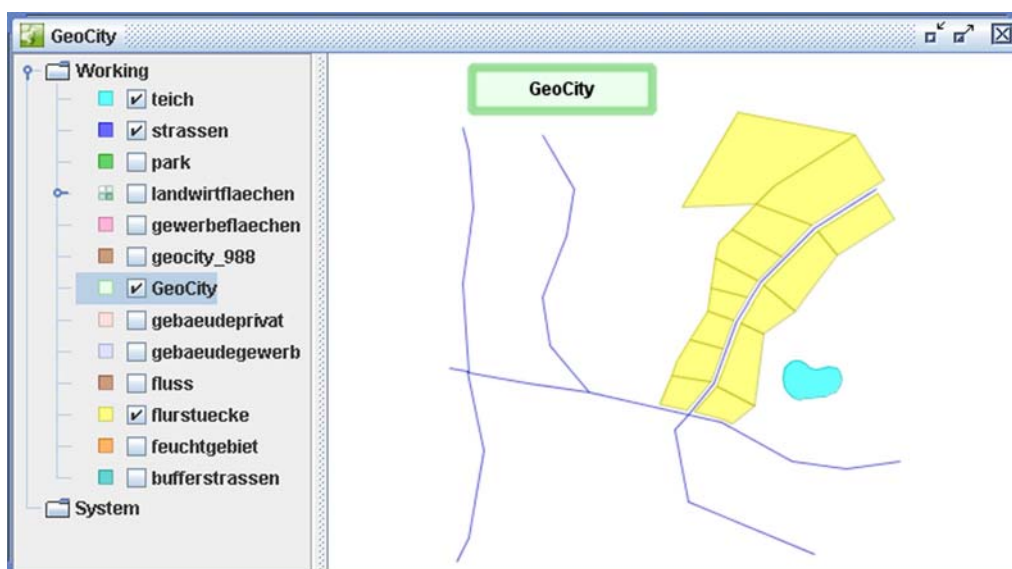
split two intersecting lines **on the intersection**

- select both lines
- left click on the intersection point

5 Editing Layers

5.1 Layer

Layers are used to display vector data (collection of features) or raster data. The left side of the project window shows an overview of all loaded layers in a project. Layer can be grouped by **Category** (e.g. „Working“ und „System“). By appropriate selection of layer names one can already establish a simple information system. In the **GeoCity** example below one can see a pond, roads, land parcels, and the label „GeoCity“. All other layers are turned off (set invisible).



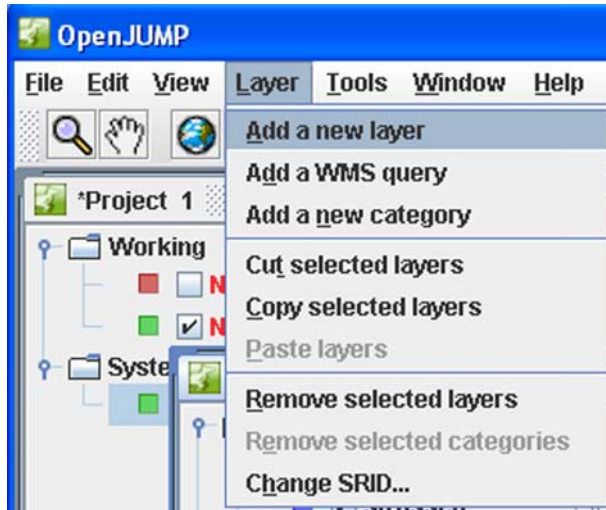
Overview on roads , parcels and pond of GeoCity

5.1.1 Layer Properties

Layer ...

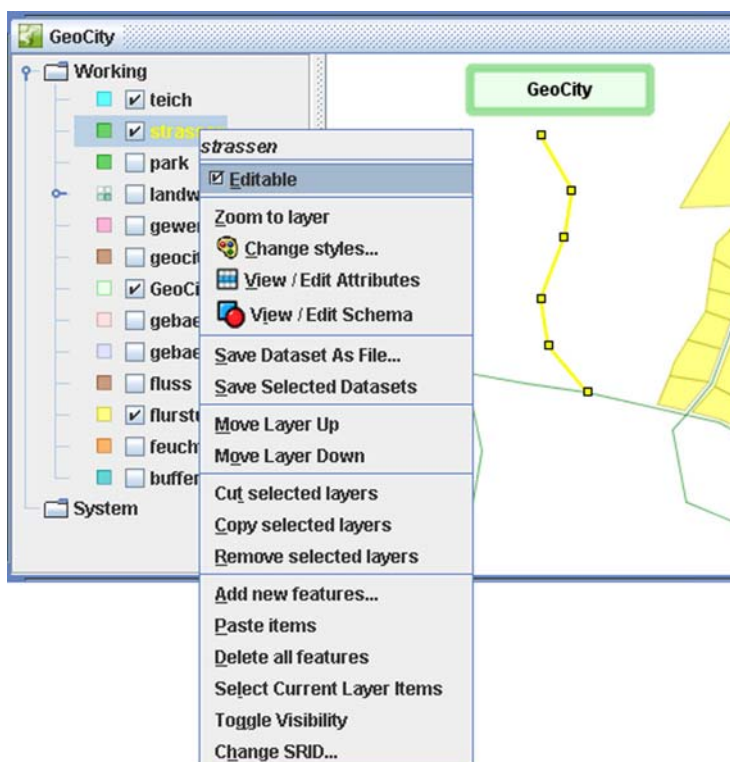
- ... have a name
- ... can be copied, cut, removed, inserted and moved in the hierarchy
- ... can be assigned to categories
- ... can display vector- and raster-data
- ... have to be saved in a file or database
- ... have a colour visualisation („**Rendering**“, page 20)
- ... can be visualised differently with respect to feature attributes
(„**Thematic Visualisation**“, page , „**Labeling**“, page)
- ... have a attribute schema (see „**Edit Schema**“ and „**Editing Attributes**“, page 18 and page)

In the „Layer“ menu one can load Layer, WMS-Layer (see „WMS-Layer“, page 16) and create layer categories.



Menu Layer for managing layers

With a right-mouse click on the layer name specific layer properties can be changed.



After right click on layer name

Editable: To edit the features in a layer, the layer must be editable!

Zoom to layer: Zooms to the layer extent (shows all features of the layer)

Change styles... : opens the menu to change the layer visualisation (page)

View / Edit Attributes: view and modify the attributes of features (page)

View / Edit Schema: View and edit the attribute schema (page 18)

Save Dataset As File... : to saves the current layer

Move Layer Up: Moving the layer up, give them a higher priority in the visualization (like „bring to front“)

Move Layer Down: Moving the layer down gives the layer a lower visualisation priority (like „send to back“)

Cut selected layers: ---

Copy selected layers: ---

Remove selected layers: ---

Add new features...: Add features in well-known text format

Paste items: inserts previously copied items (only the geometries, not the attributes)

Delete all features: Deletes the features in that layer (if the layer is editable)

Select Current Layer Items: selects all items in the layer

Change SRID...: changing the SRID for this layer (only for PostGIS database connections, page 30)

5.1.2 Saving Layers and Projects

Every modified layer has to be saved on its own! Several storage format are available like „**JUMP GML**“, ESRI Shape or the „**PostGIS**“ format.

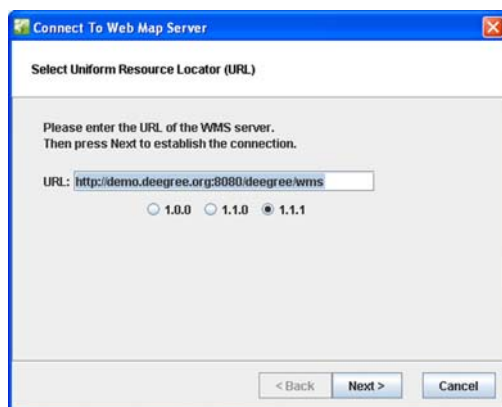
With „**Save Dataset As File...**“ the current layer can be saved in a file.

The menu option „**File>Save dataset as...**“ enables to save a layer also in a database (see „**Writing Tables**“, page). If all layer are saved one can save the project using „**File>Save Project**“ (not the other way around!).

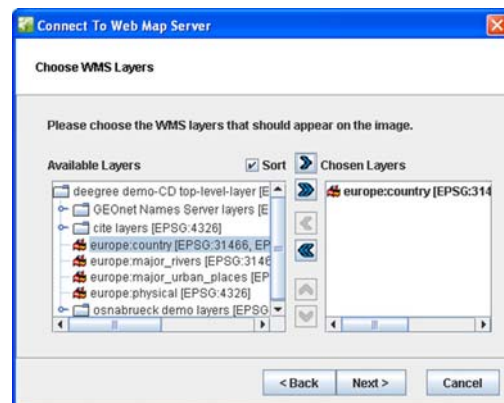
5.2 WMS-Layer

The data available from **WMS server** (Web Map Service) can be displayed in OpenJUMP as layers. Such a layer does only display the WMS data and does not have the same properties of usual layers. For instance one can not draw new data in that layer or modify the data. But such layer can be serve as base for extending the information system by vectorization / digitizing new geometries and linking them with attributes. With „**Layer>Add a WMS query**“ the URL (Uniform Resource Locator) of the WMS server can be added. The WMS server can deliver different **WMS layer**, which can be select using the button „Choose Layers“ .

Note: If one wants to display the WMS layer in OpenJUMP in different layers, then one needs to redo („**Layer>Add a WMS query**“) for every WMS layer!

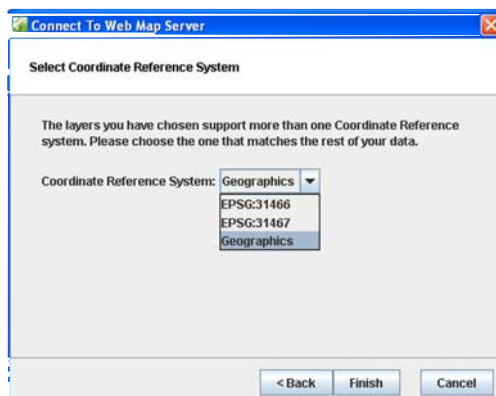


Adding the URL of the WMS server



Choosing the WMS layer

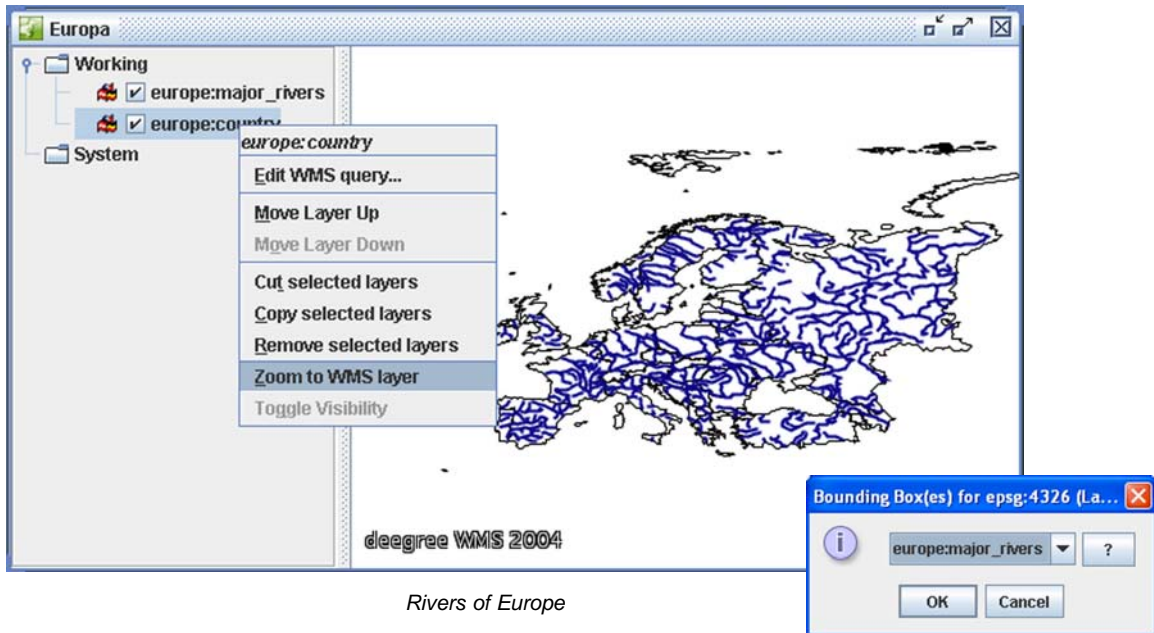
A WMS server sometimes provides the raster data in different Coordinate Reference Systems (CRS). To display the data the appropriate CRS has to be chosen. Here, the references systems are specified using the EPSG notation (European Petroleum Survey Group).



Choosing the coordinate reference system

After choosing the EPSG code from the list proceed with the „Finish“ button. It may happen that the layer view stays empty. Using „**View>Zoom to WMS layer**“ will zoom to the selected layer. Dependent on the network connection and the capabilities of the server it may take a few second until the raster image has been load.

In the **example** below two layers are displayed with images from two WMS-Layer. With right-click on the layer and choosing „**Zoom to WMS layer**“ one zooms to the bounding box of the WMS layer. Here, a **Bounding Box** defines the extent of a WMS layer.



If **several** WMS layer exists on **one** OpenJUMP layer, one can use the function „**Zoom to WMS layer**“ to *zoom to the adequate* bounding box of the WMS layer. Additionally one can obtain different information on the sent wms data by pressing the button . For Instance in the table below one can see title, name, SRS (Spatial Reference System) and the extent of the specific bounding boxes.

Title	Name	SRS	MinX	MinY	MaxX	MaxY
deegree demo-CD top-lev...	Unknown	epsg:4326	-179.9	-89.9	179.9	89.9
deegree demo-CD top-lev...	Unknown	latlon	-180	-90	180	90
deegree demo-CD top-lev...	Unknown	epsg:4326	-179.9	-89.9	179.9	89.9
cite layers	Unknown	epsg:4326	-179.9	-89.9	179.9	89.9
cite layers	Unknown	latlon	-180	-90	180	90
cite layers	Unknown	epsg:4326	-179.9	-89.9	179.9	89.9
cite BasicPolygons	cite BasicPolygons	epsg:4326	-10	-10	10	10
cite BasicPolygons	cite BasicPolygons	latlon	-10	-10	10	10
cite Bridges	cite Bridges	epsg:4326	-1	-1	1	1
cite Bridges	cite Bridges	latlon	-1	-1	1	1
cite BuildingCenters	cite BuildingCenters	epsg:4326	-1	-1	1	1
cite BuildingCenters	cite BuildingCenters	latlon	-1	-1	1	1
cite Buildings	cite Buildings	epsg:4326	-1	-1	1	1
cite Buildings	cite Buildings	latlon	-1	-1	1	1
cite DividedRoutes	cite DividedRoutes	epsg:4326	-1	-1	1	1
cite DividedRoutes	cite DividedRoutes	latlon	-1	-1	1	1
cite Forests	cite Forests	epsg:4326	-1	-1	1	1

Information on WMS layer of the WMS server

6 Schema editing


For every layer one can create a so called schema. With the schema the non-spatial attributes of the features are defined.


Example: One layer displays several land parcels. If one wants to capture the **name** of the owners, the **value** or the for example the **area** of the land parcels, one needs to define these attributes beforehand in the layer schema.


This schema is then applied to all features in the layer. Therefore for the modelling with GIS it is very important to define the layers and the schema respectively in such a way, that similar objects can be stored in the same layer. For instance it is not very useful to put land parcels and trees in the same layer since land parcels and trees have different properties and subsequently attributes. As we will see later, the term **schema** is adequate to the term **Relation-Schema** of a relational databases.


Every **attribute**, sometimes also called **field**, is assigned a specific **datatype**. OpenJUMP knows the following datatypes: **Integer** (whole numbers), **Double** (decimal numbers), **String** (text), **Date** and **Geometry**.


Note: The datatype „Object“ is not used here. It is only for customized applications!

After right click on the layer name we can select the option  to edit the schema. Therefore the layer has to be editable!

Add attribute 

Delete attribute 

Move up 

Move down 


Edit Schema: flurstuecke

	Field Name	Data Type
+	GEOMETRY	Geometry
X	vorname	String
X	nachname	String
^	flaeche	Integer
v		Object
		Integer
		Geometry
		Double
		Date
		String



Apply Changes
Revert Changes
 Force invalid conversions to null

Schema for the Layer „flurstuecke“ with the attributes GEOMETRY, vorname, nachname and flaeche

7 Editing Attributes

If a Schema has been set up one can add the attribute values for the features. Applying a right mouse click on the layer name one can access  **View / Edit Attributes** and edit attributes.

The layer has to be editable if attribute values should be changed, but viewing is always possible .

The toolbar of the attribute window provides functions for zooming and identification of features which have been selected in the table. For instance if one likes to know which land parcels are owned by Otto Fant one simply marks the row of Otto. By Left click on the magnifier icon  the parcel is shown and with a left click on the flashlight  the parcel outline is flashed.

To use the functions from the toolbar one or more rows in the table have to be selected.

scrolling und zooming

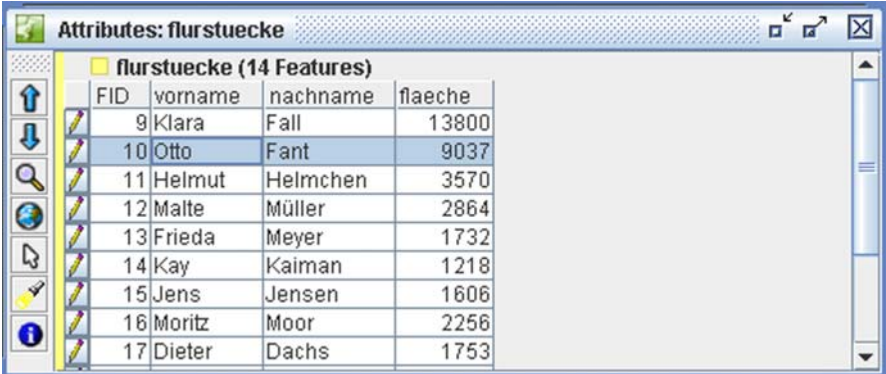
Zoom selected row

Zoom to all

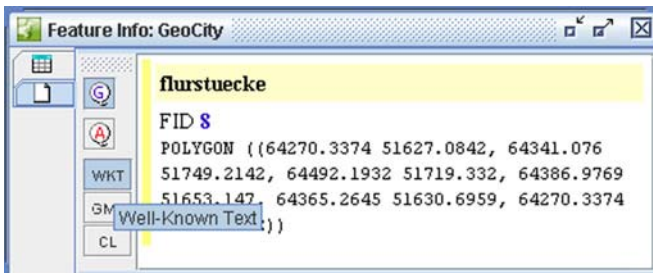
Show and select in layer view

Show in layer view


Show geometry data



Attribute table of the layer „flurstuecke“




Geometry data in well-known text format (WKT)

The geometry data of the selected are displayed if the info-button  of the attribute table is pressed. The display format can be switched between **WKT** (Well-Known Text), **GML** (Geography Mark-Up Language) and the **CL**-format (coordinate list).

8 Layer display (Styles)

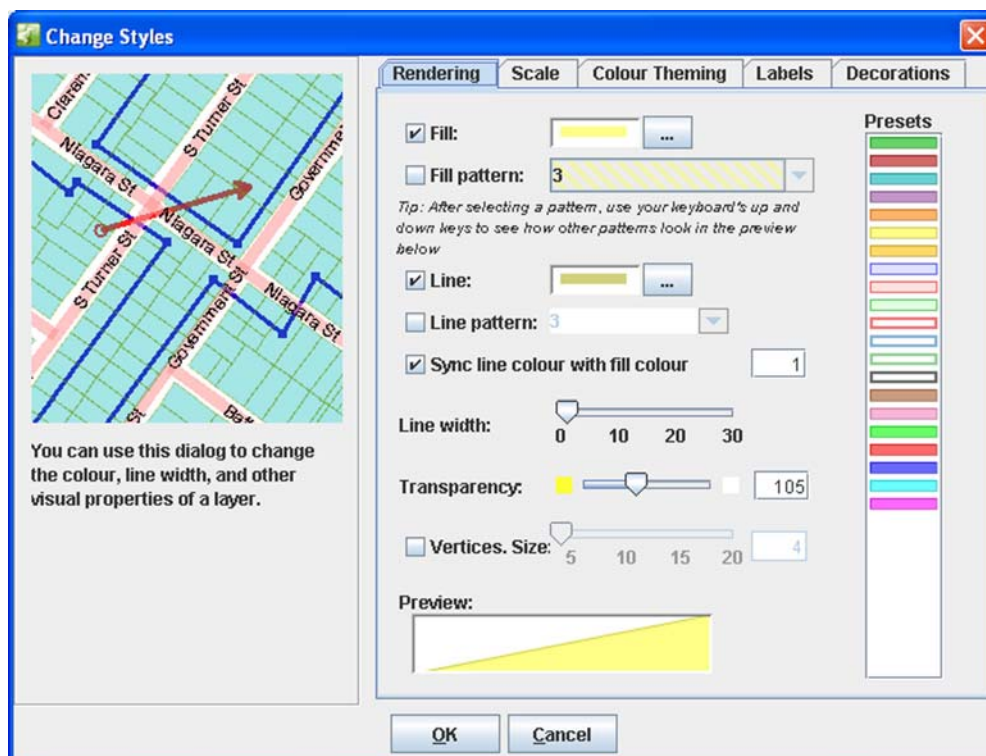


The display of layers, thus the display of the features in that layer can be changed using the „**Change Styles**“ window, which can be activated using the button:  Therefore the layer needs not to be editable. The window offers four possibilities to change the display: **Rendering** (normal styles), **Colour Theming** (thematic styling), **Labels** (text display) und **Decorations** (Start-/Endpoint style of lines).

8.1 Normal Styles (Rendering)

On the tab called „Rendering“ the following things can be changed:

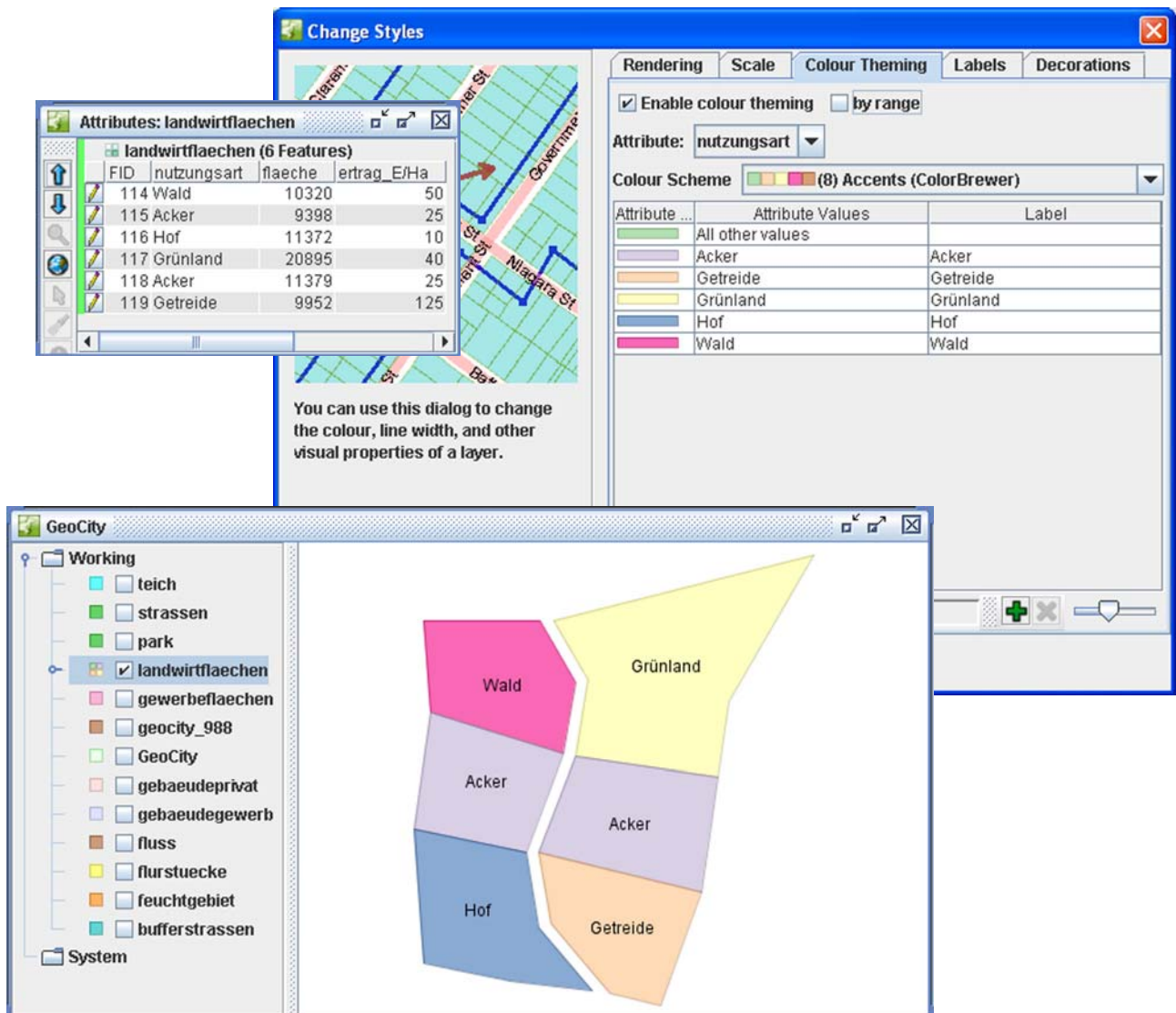
- **colour** of lines and areas
- **fill patterns** of areas (Fill pattern)
- **line colour style** (Line pattern)
- line width
- **transparency** of colours
- size of vertices



Window to change the normal styles of a layer

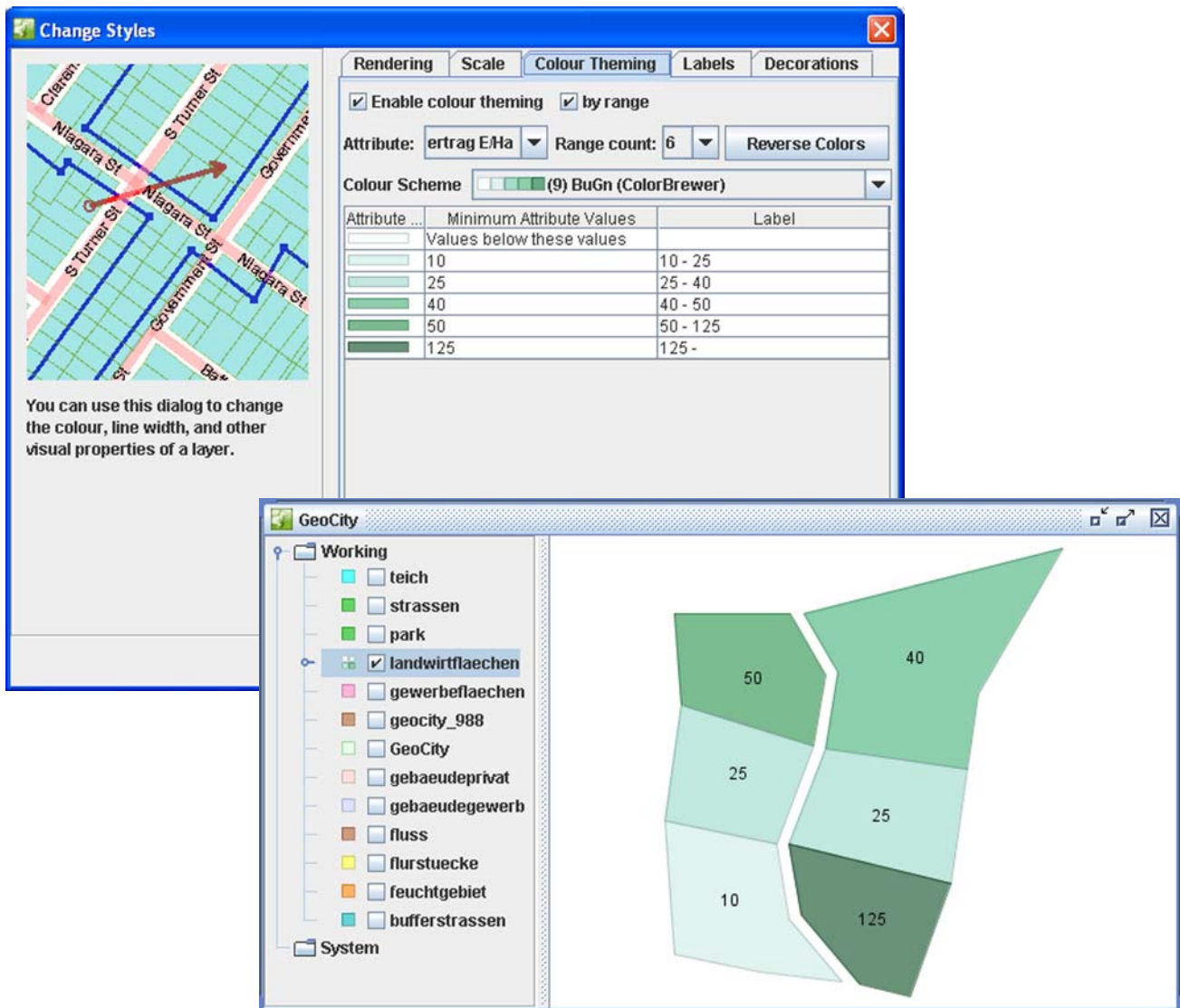
8.2 Colour Theming

If one likes to create a thematic map, the layer features must have attributes (see Editing Schema and Attributes). In the example below the features of the layer „landwirtflaechen“ have the attribute „nutzungsart“ with the possible **values**: Acker, Getreide, Grünland, Hof und Wald. The colours can be changed by using the „Colour Scheme“ list/box or by left-click on the attribute colour.



Thematic display of land-use with labeling

Are the attribute values numerical values (no text, e.g. Integer, Double) and the values should be grouped into intervals, then the checkbox „**by range**“ must be checked. If this is done, then depended on the setting of „**Range count**“ the ranges are calculated and their colours set. In the example below exists for every farming site an attribute earning in Euro per Hectare (ertrag E/Ha). Since „**Range count**“ is set to 6, one obtains six intervals which are coloured in a green scale fashion.



Thematic Display of Ertrag pro Hektar (Earning per Hektar) with labeling

8.3 Labeling

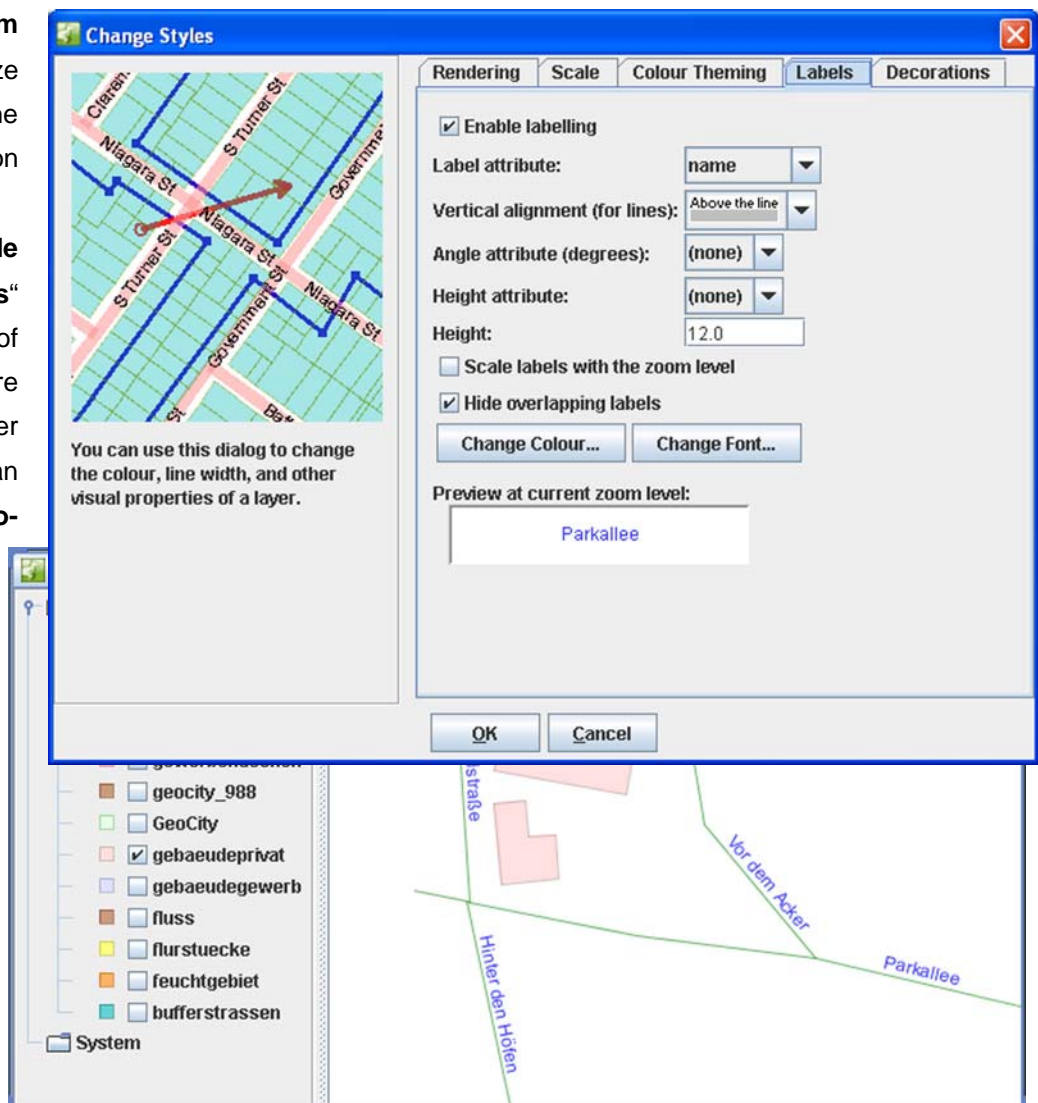
Attribute values of features can be displayed as text labels. In the shown **example** the streets of the layer „strassen“ are labelled using the attribute „name“. Therefore in the window of the style tab „Labels“ the box of „**Enable labelling**“ has to be checked and in the drop down list of „**Label attribute**“ the attribute name needs to be chosen. Additional possibilities for the text display are **Vertical alignment (for lines)**, the rotation angle (**Angle attribute**) and the font size (**Height attribute**), which can be set with respect to the attribute values. A fixed **font size** can be setup using the „**Height**“ option. In contrast, if one chooses the option „**Scale**



FID	name	laenge
260	Parkallee	601
261	Hafenstraße	683
262	Vor dem Acker	365
263	Landstraße	317
264	Hinter den Höfen	251

labels with the zoom level“ the font size will change on the screen depended on the zoom.

A further option: „**Hide overlapping labels**“ is useful to switch of labels which are overlapped by other labels. Finally on can use „**Change Colour...**“ and „**Change Font...**“ to obtain the preferred text style. The positioning of the labels is done by the program itself and can not be influenced by the user!



Change Styles

You can use this dialog to change the colour, line width, and other visual properties of a layer.

Rendering | **Scale** | **Colour Theming** | **Labels** | **Decorations**

Enable labelling

Label attribute: name

Vertical alignment (for lines): Above the line

Angle attribute (degrees): (none)

Height attribute: (none)

Height: 12.0

Scale labels with the zoom level

Hide overlapping labels

Change Colour... Change Font...

Preview at current zoom level:

Parkallee

OK Cancel

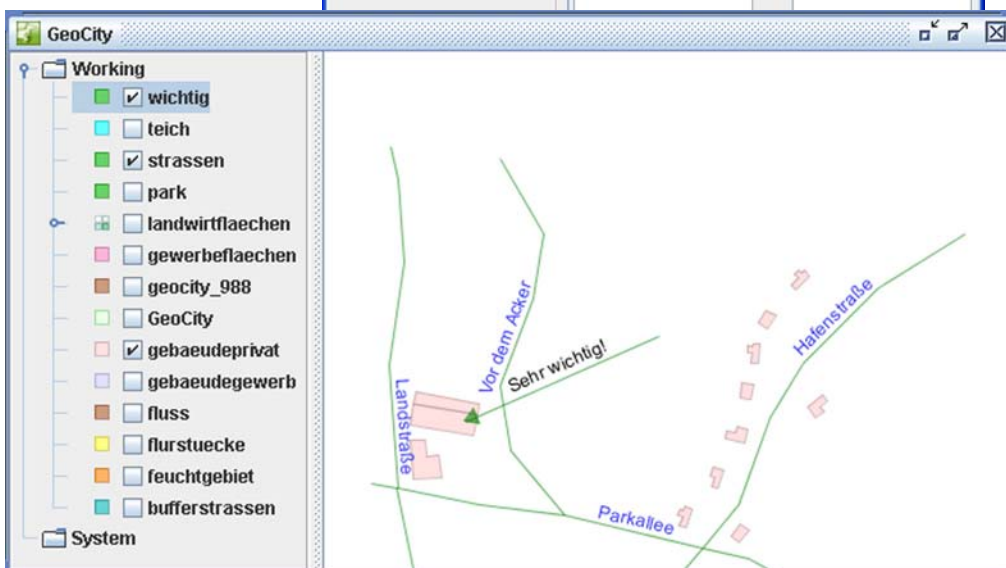
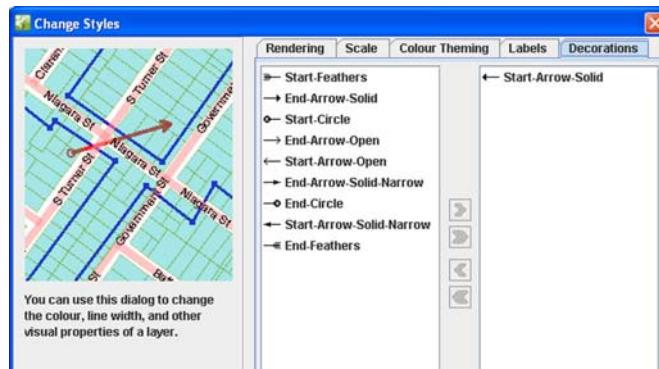
Legend:

- geocity_988
- GeoCity
- gebaeudeprivat
- gebaeudegewerb
- fluss
- flurstuecke
- feuchtgebiet
- bufferstrassen
- System

Labeling of Streets

8.4 Start- and End Point Display (Decorations)

Using the options provided by the „decorations“ window one can define the symbols for start and end points of lines. The selected symbols need to be transferred from the left window to the right window with the arrow buttons, to be applied on the lines. Thereby the symbols are applied to all features of a layer and not only to single feature.



The style „Start-Arrow-Solid“ applied to a line layer in combination with labeling

9 Analysis Tools

OpenJUMP delivers a number of spatial analysis tools, which can be accessed via the „*Tools>Analysis*“ menu. Some of them are

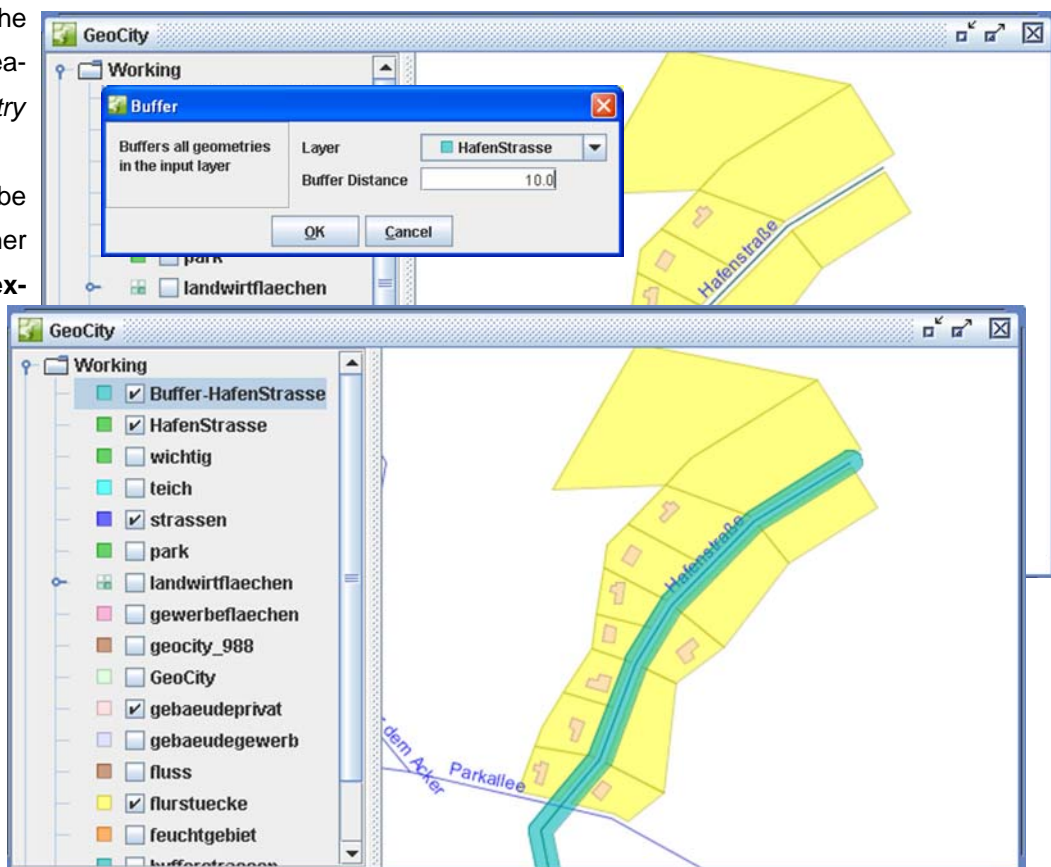
- buffer (similar to a fringe or distance zone)
- intersections (Intersect, Overlay)
- union
- difference sets (A-B), (B-A), symmetric

9.1 Buffer

If a buffer function is applied to a geometry one obtains the fringe region around it. The function delivers usually an **area** (polygon) in a **new layer**. This layer is automatically created. If one applies the buffer function on a point the result will be a circular area around the point. If a line is buffered by a defined distance then one will as well obtain a new area object with the input line as centre axis. Thereby the original input geometries are not modified! The buffer function is applied in OpenJUMP always on **all geometries**, part of a layer (note: to apply the buffer to a single feature use *Geometry Functions...*)


This new area can be used for further analysis. In our **ex-**

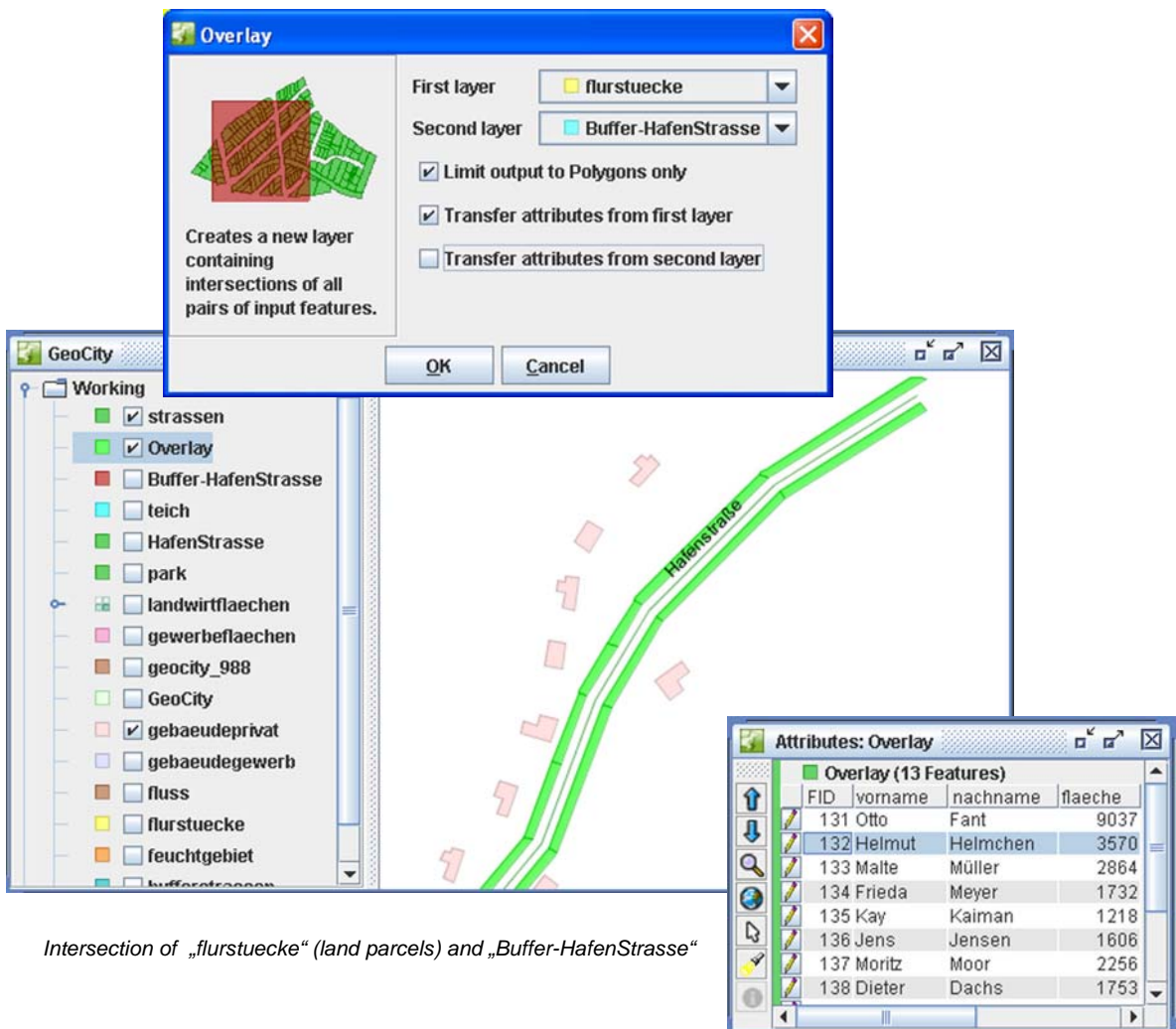
ample we would like to widen the street „Hafenstraße“ to 20m. Therefore we create a buffer of 10m distance around the centre line of the street. We obtain a new layer (Buffer-Hafen-Strasse), with the extended Hafenstrasse.



Extending the street Hafenstraße

9.2 Overlay

The overlay function is useful to calculate the intersections between two or more areas. In our **example** of the widening of the street (see Buffer) we can intersect the street buffer areas with the land parcels. The intersection areas are returned in a new layer and can be regarded as necessary area for the extension of the street. With help of the **View/Edit Attributes** button  we can see the list of owners of the intersections areas.

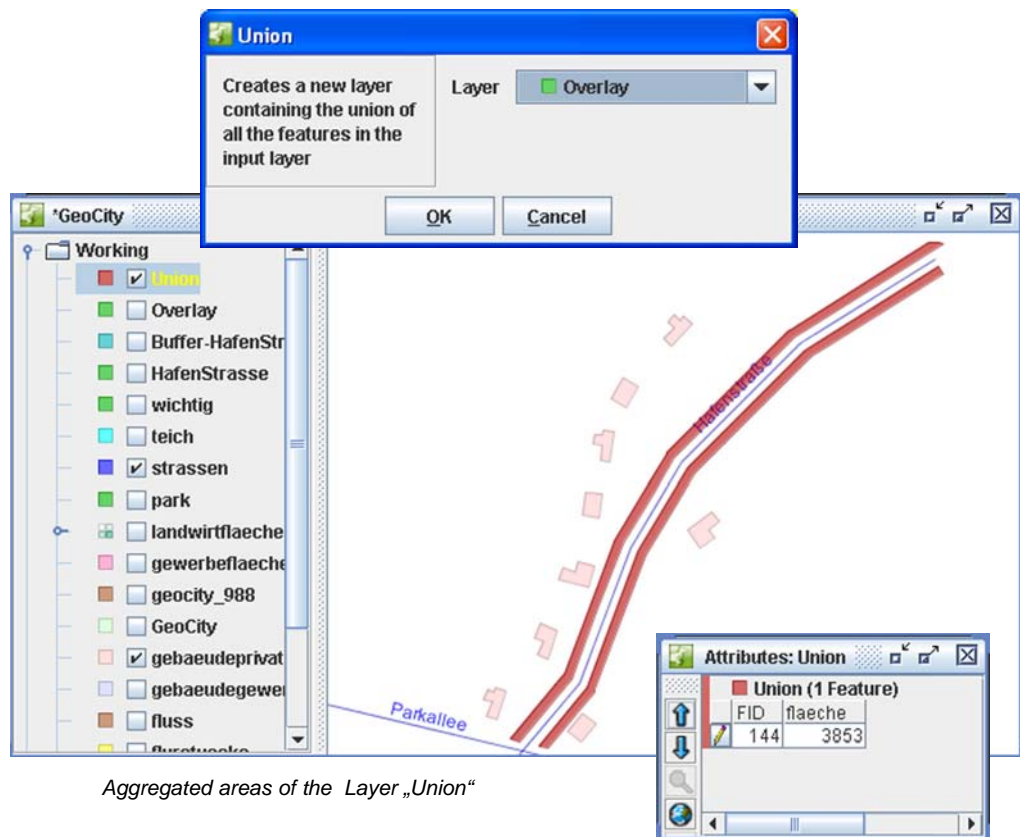


Intersection of „flurstuecke“ (land parcels) and „Buffer-HafenStrasse“

Affected land owners

9.3 Union

The union tool merges the areas of one layer to one single area. The new area is returned on a new layer. With respect to the **example** of the extension of the street „Hafenstraßen“ it is useful to do a cost approximation. Therefore we want to use the total area of the land parcels within the layer „Overlay“, containing the intersection result of. To obtain the area we first apply „Union“ to the layer „Overlay“, which results in a new layer called „Union.“ In the second step we extend the layer schema of the „Union“layer by the attribute „area“, which should be of type „double“. Finally the third step is to use the function „**Analysis>Calculate areas and lengths...**“ to calculate the total area. Note: therefore we need to have an editable layer.



Aggregated areas of the Layer „Union“

Calculated total area

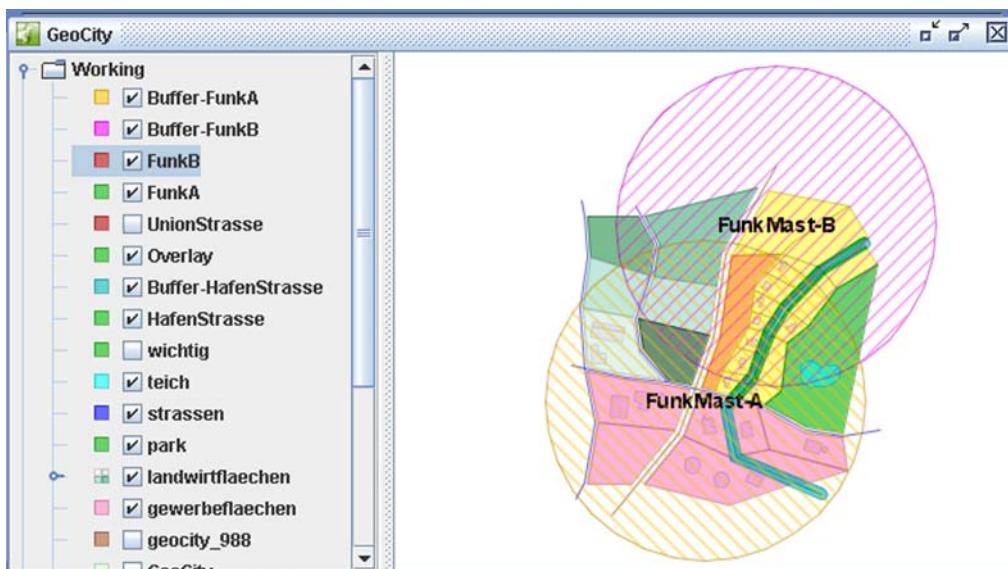
9.4 Geometry Functions

The geometric functions, which can be found under „**Tools>Analysis>Geometry functions...**“ provide different vector algebra methods, like **A-B**, **B-A** and **Symmetric Difference**. These functions account for **two layers** and consider only one single feature at a time.

The function **Difference A-B** creates a cookie cut region, whereby area B is cut out of area A and the resulting cookie cut is presented in a new layer. For instance if B is a circle area within the circle area A, then the result is a donut area. If you play a bit you will see that this hold similar for B-A. The function **Symmetric Difference** aggregates the area of layer A with the area of layer B (Union) and subtracts from it the overlapping area. Here as well, the resulting area is returned in a new layer.

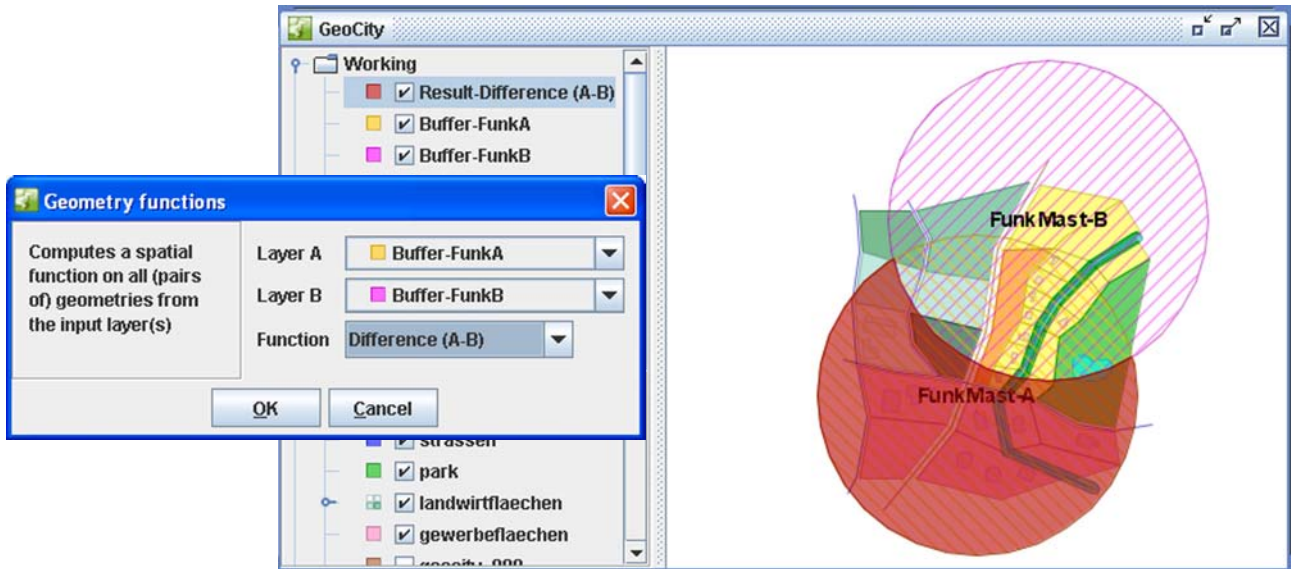
In our new second **example** got Radio-GeoCity two brand new antenna poles (FunkMast-A and FunkMast-B). But if the set them up we have to take care for interferences of radio waves, which can cause troubles in receiving the radio. Therefore we need to find out:

1. Which area is covered by mast A without radio reception problems?
2. Which area is covered by mast B without problems?
3. Which area has good radio reception?



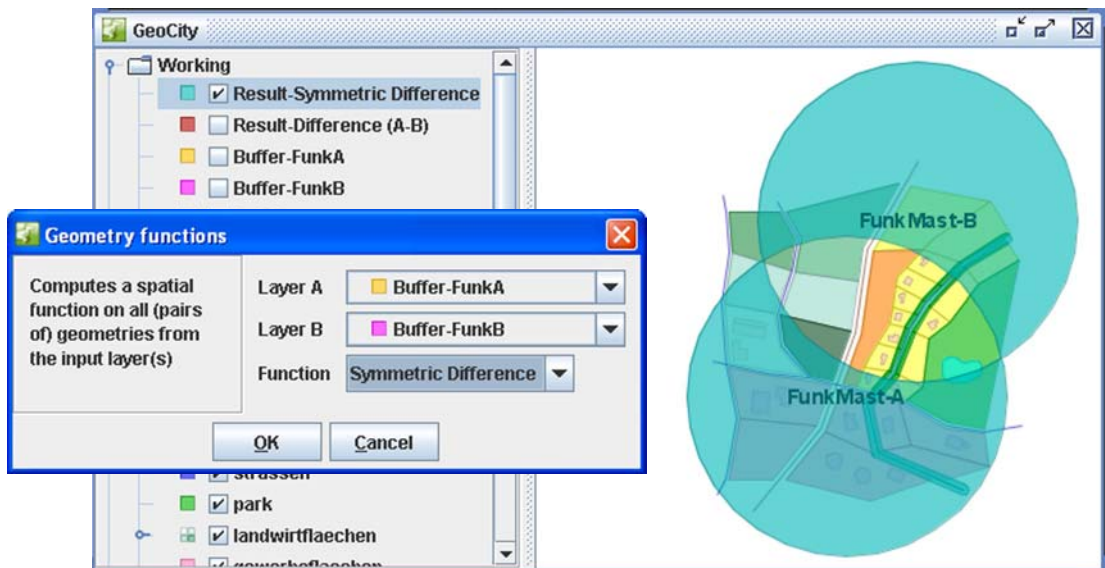
Radio-GeoCity's two new antenna poles

The **first question** can we answer, if we calculate the difference A-B using the layer Buffer-FunkA and Buffer-FunkB. The **second answer** can be obtained by just using the other difference function B-A.



Trouble -free reception via Mast-A (red; Difference A-B)

Finally the **third question** can be answered if we calculate the symmetric difference function of Buffer-FunkA and Buffer-FunkB. Unfortunately we obtain, that the residents of the area around the street „Hafenstraße“ will probably have difficulties to receive the radio station. If the engineers would have used OpenJUMP for the planning, the problem could have been avoided :-)



Good radio reception (blue; Symmetric Difference)

10 Connection to a PostgreSQL/PostGIS database

With an additional PlugIn it is possible to work in OpenJUMP with tables (relations) of a PostgreSQL/PostGIS database. **PostgreSQL** is a **object relational database management system** (ORDBMS), which can be adapted to geographic purposes using the so-called **PostGIS-Extension**. This extension allows to store and edit geometries and attributes in usual relations.

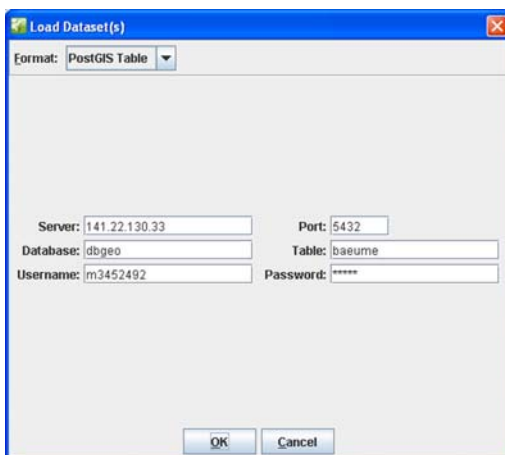
To use OpenJUMP with PostGIS, one needs to install a **JDBC driver** (Java Data Base Connectivity) for PostgreSQL and two OpenJUMP PlugIns. The newest JDBC-PostgreSQL driver can be obtained from the PostgreSQL-Homepage <http://www.postgresql.org/> (e.g. **postgresql-8.0.309.jdc3.jar**). To install the driver, copy the file into the folder `...\lib\ext` of the JavaRuntimeEnvironment (e.g. `C:\Programme\Java\jre1.5.0_04\lib\ext`).

The OpenJUMP PlugIns **JumpPostGISrzcj.jar** and **sridsupport.jar** must be placed into the folder `...\lib\ext` of the your OpenJUMP installation (e.g. `C:\Programme\OpenJUMP\lib\ext`). Thereby the PlugIn `JumpPostGISrzcj.jar` handles the connection to the database server and the reading and writing of the data. The second PlugIn `sridsupport.jar` is useful to assign a layer a **key** for the used **spatial reference system** (Spatial Reference System Identifier, **SRID**). To establish a connection with a PostgreSQL/PostGIS server one needs several information:

- the IP-address of the PostgreSQL server
- the port number (usually 5432)
- the database name
- the table name
- the user account name and the corresponding password

10.1 Reading Tables

If spatial tables do already exists in PostGIS format, then one can load them using „**File>Load Dataset(s)...**“.



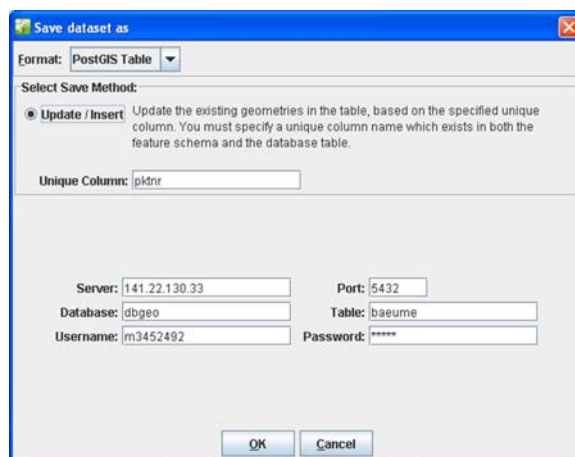
Reading PostGIS tables

Therefore one must chose in the „Format:“ field the option **PostGIS Table**. The table is afterwards displayed as a layer in OpenJUMP, which carries the same name like the table.

10.2 Writing Tables

With the menu option „**File>Save dataset as...**“ a **selected layer can** be stored in a data base. The features of this layer need to have an attribute which can be used a unique identifier (primary key), to have a unique assignment for the datasets. The name of this attribute has to be entered in the „**Unique Column:**“ field. Does the table not yet exist, then a new table is created. Every table in PostGIS is assigned a SRID value (Spatial Reference System Identifier). One condition for storing the datasets is, that they must have the same spatial reference system. The function „**Layer>Change SRID...**“ can be used to assign the currently selected layer a SRID value, assuming the corresponding plugin has been load (see above).

NOTE: The plugin does only enable to **edit** existing datasets or **add** new datasets! But if single features are deleted from a layer, then these datasets are **not deleted** on writing to the database! To delete datasets one need to either store the layer with a new table name or delete the table beforehand in PostgreSQL (which is unfortunately a bit cumbersome).

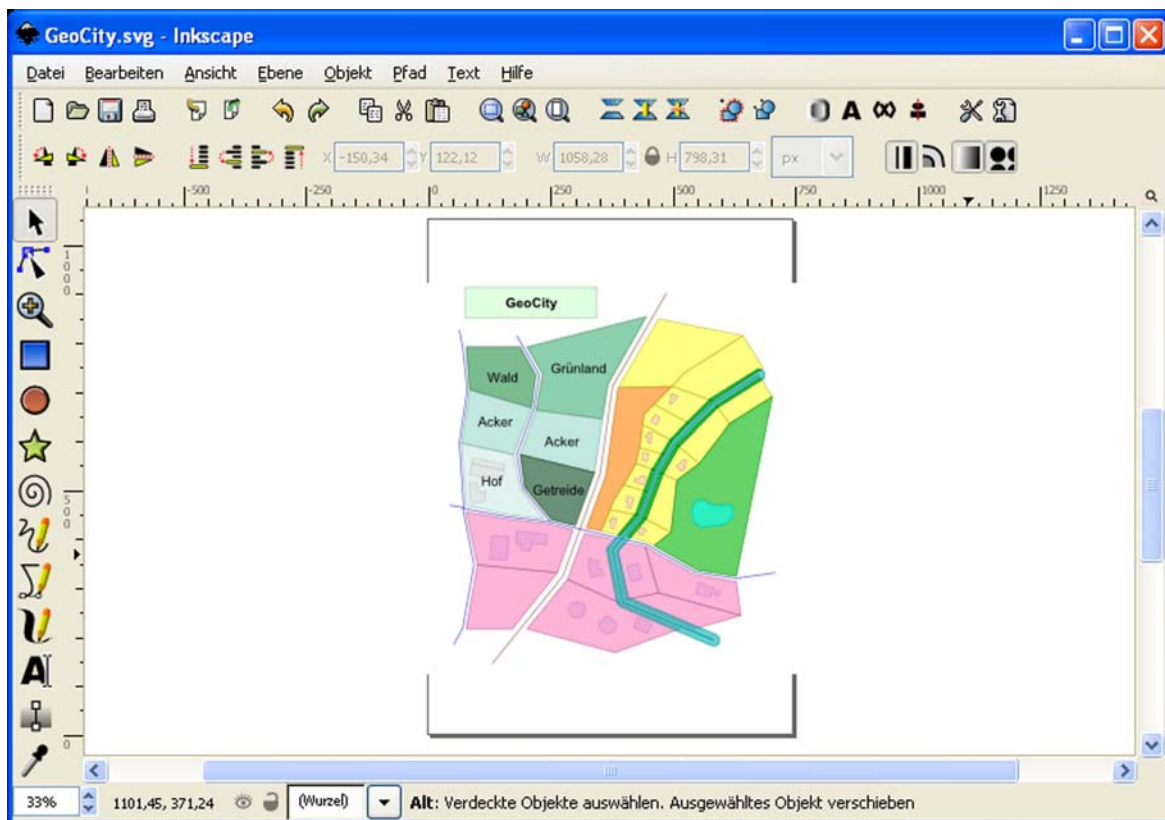


Writing PostGIS Tables

11 Printing

In OpenJUMP it is not possible to print the displayed graphics directly. However, graphics can be saved either as raster- (PNG, JPEG) or vector format (Scalable Vector Graphics, SVG) in a file – and afterwards printed with an appropriate (opensource :) program. E.g. For raster images PhotoFiltre <http://www.photofiltre.com> or GIMP and for vector graphics Inkscape <http://www.inkscape.org>. Thereby the vector format SVG is to prefer since it is better scalable and easier to edit than raster graphics.

With the function „**File>Save Image in SVG Format**“ one can store the current view of the project in SVG format. Thereby it will be stored only **Layer** which are **visible**. With the opensource program **Inkscape** the SVG files can be edited and printed.



GeoCity as graphics in Inkscape

12 Glossary

CRS: Coordinate Reference System

EPSG: European Petroleum Survey Group; nowadays **OGP** (Oil & Gas Producers) Surveying & Positioning Committee. www.epsg.org

The Oil & Gas Producers Surveying and Positioning Committee maintains and publishes parameters and descriptions for co-ordinate reference systems. Such parameters are give a key code, the so-called **Spatial Reference System Identifier (SRID)**. Such identifier numbers are for instance used by **OGC** compliant services (e.g. **WMS**) and in **PostGIS**.

(See also OGC: „Coordinate Transformation Services“).

Example:

- EPSG: 4326 = Geographic Co-ordinates in the WGS84 reference system
- EPSG: 31466 = Gauß-Krüger Germany, 2. Meridian Zone
- EPSG: 31467 = Gauß-Krüger Germany, 3. Meridian Zone
- EPSG: 31468 = Gauß-Krüger Germany, 4. Meridian Zone

The files containing the information on the references (**EPSG geodetic parameter dataset**) are available from the website <http://www.epsg.org/>.

Feature (Object):

- Features are abstracted objects of the real world. For instance streets are abstracted and displayed using Linestrings, buildings as areas or trees as points.
In OpenJUMP every feature has a spatial attribute (geometry) and none or several non-spatial attributes e.g. street name, owner, tree height.
- A group of spatial elements, which represents together a piece of the real world. Often synonymously used with the term object. A feature can also be aggregated to more complex features (objects), consisting of more than one group of spatial elements. (Lexikon der Geoinformatik, 2001)
- A geographic feature is „an abstraction of a real world phenomenon ... associated with a location relative to Earth“. A feature has spatial attributes (polygons, points, etc.) and non-spatial attributes (strings, dates, numbers). (JUMP Workbench User's Guide, 2004)

GeometryCollection: Collection of **several** features of one layer to **one** feature.

GML: Geography Mark-Up Language; see also on <http://www.opengeospatial.org/>

GNU General Public License: License type of free software; <http://www.fsf.org/licensing/licenses/gpl.html>

Inkscape: Editor for vector graphics (Open Source); <http://www.inkscape.org/>

Item: Graphic display of a features in OpenJUMP.

JPEG: Joint Photographic Experts Group; graphic format; <http://www.jpeg.org/>

JUMP: Unified Mapping Platform; A Geographical Information System; <http://www.jump-project.org/>

Mapserver: A development environment for the setup of internet applications with dynamic map content; <http://mapserver.gis.umn.edu/>

OGC: Open Geospatial Consortium; <http://www.opengeospatial.org/>
International society for the standardization of formats and interfaces for GIS and Location Based Services (LBS) applications. It is nowadays made up largely by companies and only to a smaller part by research institutes.

OGP: Oil & Gas Producer; <http://www.ogp.org.uk/>

OGP Surveying and Positioning Committee: former EPSG, <http://www.epsg.org/>

OpenGIS: see OGC; <http://www.opengeospatial.org/>

OpenJUMP: Geographical Information System; extended version of JUMP; <http://openjump.org/wiki/show/HomePage>

Open Source: the source code of a software is open to everybody; http://en.wikipedia.org/wiki/Open_source

PNG: Portable Network Graphics; graphic format for loss less storing of images

PostGIS: Extension of PostgreSQL for the work with geographic objects; <http://postgis.refrations.net/>

PostgreSQL: Object relational database management system; <http://www.postgresql.org/>

Refractions Research: Canadian Company which developed PostGIS and took part on the development of JUMP; <http://www.refractions.net/>

Spatial attributes: e.g. point, line, area.

Spatial information: geo-information, related to a place on the earth (or other stellar object)

SRID: Spatial Reference System Identifier; key code for specific (earth) co-ordinate systems

SRS: Spatial Reference System

SVG: Scalable Vector Graphics; recommended by the W3C to store vector graphics;
<http://www.w3.org/Graphics/SVG/>

URL: Uniform Resource Locator; e.g. a link in a browser.

Vertex, vertices: node, corner point.

Vivid Solutions: Canadian Company which developed JUMP in large parts; <http://www.vividsolutions.com/>

W3C: World Wide Web Consortium; <http://www.w3.org/>

Well-Known Binary (WKB): Binary representation for geometries, defined in the OpenGIS (OGC) document „*OpenGIS Simple Features Specification For SQL*“ .

Wiki: A **Wiki**, sometimes called **WikiWiki** or **WikiWeb**, is a collection of WebPages in the [World Wide Web](#), which can not only be read by the users, but also online edited. Thus, Wikis are alike [Content Management Systems](#). The name emerged from the Hawaiian word *wikiwiki*, which means "fast". (Wikipedia, 2005)

Well-Known Text (WKT): Textual representation of geometries, defined in the OpenGIS (OGC) document „OpenGIS Simple Features Specification For SQL“. Because of the textual representation they are easier to read for humans. For instance a point is represented by 'POINT (10 15)'.

Geometry Type	SQL Text Literal Representation	Comment
Point	'POINT (10 10)'	a Point
LineString	'LINESTRING (10 10, 20 20, 30 40)'	a LineString with 3 points
Polygon	'POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))'	a Polygon with 1 exterior ring and 0 interior rings
Multipoint	'MULTIPOINT (10 10, 20 20)'	a MultiPoint with 2 point
MultiLineString	'MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))'	a MultiLineString with 2 linestrings
MultiPolygon	'MULTIPOLYGON (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))'	a MultiPolygon with 2 polygons
GeomCollection	'GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))'	a GeometryCollection consisting of 2 Point values and a LineString value

Geometry types in WKT-Format (Source: OpenGIS Simple Features Specification for SQL)

WKB: see Well-Known Binary

WKT: see Well-Known Text

WMS: Web Map Service; An internet service which delivers on standardised requests standardised data for map like (raster) display of geodata. This services is specified as an OGC standard.

13 Literature References

Aquino, J., and M. Davis (2004):

JUMP Workbench User's Guide, Vivid Solutions

Aquino, J., and D. Kim (2003):

JUMP Developer's Guide, Vivid Solutions

Bill, R., and M. L. Zehner (2001):

Lexikon der Geoinformatik, Wichmann Verlag

Burrough, P., and R. A. McDonnel (1998): (*comprehensive and technical textbook*)

Principles of Geographical Information Systems. 2nd Edition, Oxford University Press

Jones, C. B. (1997): (*comprehensive and technical textbook*)

Geographical Information Systems and Computer Cartography. Addison-Wesley Longman.

Lake, R., D. S. Burggraf, M.Trninic and L. Rae (2004):

Geography Mark-Up Language (GML), John Wiley & Sons, Ltd

Goodchild, M. F., D. J. Maguire and D. W. Rhind (2005): (*reference book*)

Geographical Information Systems: Principles, Techniques, Management and Applications. 2nd Ed., Wiley John + Sons .

OGC (2003):

OpenGIS Geography Markup Language (GML) Implementation Specification, Open GIS Consortium

OGC (2001):

OpenGIS Implementation Specification: Coordinate Transformation Services, Open GIS Consortium

OGC (1999):

OpenGIS Simple Features Specification For SQL Revision 1.1, Open GIS Consortium

OGC (2004):

Web Map Service (WMS), Version: 1.3, Open GIS Consortium

Refractions Research (2005):

PostGIS Manual

The PostgreSQL Global Development Group (2005):

PostgreSQL 8.1.0 Documentation

14 Link Collection

Inkscape	http://www.inkscape.org/
JUMP	http://www.jump-project.org/
Jump Pilot Project	http://jump-pilot.sourceforge.net/
MapServer	http://mapserver.gis.umn.edu/
OGP Surveying & Positioning Committee	http://www.epsg.org/
Open Geospatial Consortium	http://www.opengeospatial.org/
OpenJUMP	http://openjump.org
PIROL, Fachhochschule Osnabrück	http://www.al.fh-osnabrueck.de/jump-download.html
PostGIS	http://postgis.refrations.net/
PostgreSQL	http://www.postgresql.org/

15 Index

Analysis.....	25	Layer View.....	5
Area.....	8, 10	Layer Display.....	20
Attributes.....	19	Layer Properties.....	13
Attribute Values.....	21	Layer Name.....	8
Bounding Box.....	17	Lines.....	8, 10, 12
Buffer.....	25	Line Width.....	20
Colour Theming.....	21	Linestring.....	10
Colour Scheme.....	21	Line Styles.....	20
Data.....	7	Link Collection.....	39
Data Types.....	18	Literature References.....	37
Decorations.....	24	Loading Data.....	7
Difference A-B.....	28	Node.....	12
Double.....	18	Open Source.....	4
Editing Toolbox.....	9	OpenJUMP.....	4
Enable Labelling.....	23	Options (Snap/Grid).....	9
End Symbol.....	24	Overlay.....	26
Features.....	5	PlugIn.....	30
Fill pattern.....	20	Point.....	8, 10
GeoCity.....	13	Polygon.....	10
Geometric Functions.....	28	Port Number.....	30
Geometry Collection.....	11	PostGIS.....	15, 30
Grid Lines.....	9	PostGIS-Extension.....	30
Glossary.....	33	PostgreSQL.....	30
GML.....	19	PostgreSQL/PostGIS Database	30
GNU General Public License.....	4	Printing.....	32
Graphical Editor.....	8	Project.....	5
Hole.....	11	Project View.....	5
Inkscape.....	32	Refractions Research.....	4
Integer.....	18	Rendering.....	20
IP-Address.....	30	Scalable Vector Graphics.....	32
JDBC-Driver	30	Schema.....	18
JUMP.....	4	Saving Layer and Projects.....	15
JUMP GML.....	15	Snapping.....	9
JumpPostGISrzcjn.jar	30	SRID.....	30
Label attribute.....	23	sridsupport.jar.....	30
Labels.....	23	Start- Endpoint Style.....	24
Layer.....	5, 13	String.....	18

Styles.....	20	Vertices.....	12
SVG.....	32	Vivid Solutions.....	4, 35
Symmetric Difference.....	28	Web Map Service.....	16
Thematic Styling.....	21	WKT.....	19
Transparency.....	20	WMS.....	16, 36
Union.....	27	WMS-Layer.....	16
Vertex.....	12	WMS-Server.....	16
Vertical alignment.....	23		