
OrbisGIS Manual Documentation

Release 5.1

G. Petit & E. Bocher

Aug 25, 2020

User documentation

1	Introduction	1
2	User documentation	3
3	Contributors documentation	41

Here you will find a documentation for both users and developers of OrbisGIS.

1.1 OrbisGIS

OrbisGIS is a free and open-source Geographic Information System, developed by researchers and mainly dedicated to research issues. The official website is <http://www.orbisgis.org/>.

All the sources of this project can be found on the [GitHub](#) account.

1.2 Licence

Except where otherwise noted, this documentation is licensed under a *Creative Commons Attribution-NonCommercial-ShareAlike 4.0 (CC-BY-NC-SA)* International License.

In this part, we present the basis of OrbisGIS (how to launch, to discover and to use the application).

2.1 Installation

2.1.1 Requirement

Before using OrbisGIS, you must ensure that you are running at least the **JAVA 7** virtual machine ([openJDK](#) or [Oracle](#)).

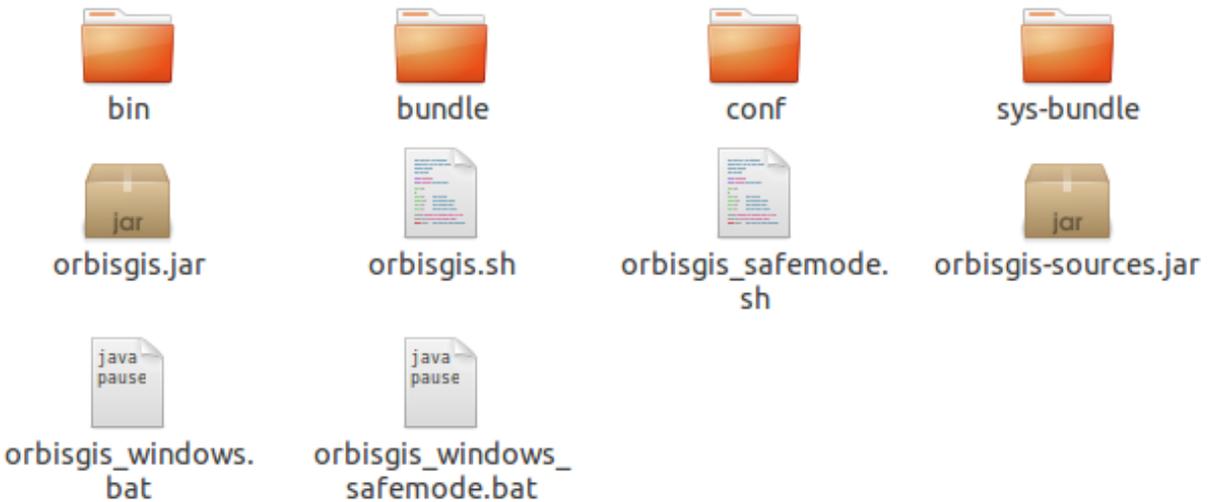
2.1.2 Download

To download OrbisGIS, just consult the official website [here](#).

2.1.3 Execute

OrbisGIS is a standalone application. It means that, if you want (see [Ubuntu PPA repository](#)), no installation is needed. In addition, it also means that you can play with OrbisGIS without any “administrator” rule !

Once the orbisgis-bin.zip file downloaded, just unzip the file.



In this folder, you can just focus on:

- orbisgis.sh : the Linux and iOS launcher,
- orbisgis_safemode.sh : the same, but in “safemode”,
- orbisgis_windows.bat : the Windows launcher,
- orbisgis_windows.bat : ... in “safemode”,
- orbisgis.jar : the application itself.

Note that “Safemode” is only useful when OrbisGIS does not start successfully. Running this file will uninstall all plugins and former OrbisGIS modules.

To launch OrbisGIS, you have the choice between:

- a double-click on the .jar file,
- for Windows, double-click on the .bat file,
- for Linux and iOS, double-click on the .sh file.

Remarks

- “execution rights” are sometimes needed.
- If you want to specify the jvm memory size, please consult this [page](#) on the OrbisGIS wiki.

2.1.4 Run the .sh file

To run the .sh file (*in Linux and iOS*) in command line, just follow these two steps:

1. open a terminal (*Ctrl+Alt+T*), then go in the unzipped folder (using the command `cd /your_url`)
2. run the file with the following command

```
./orbisgis.sh
```

```

gpetit@HP-gpetit: ~/orbisgis-dist-5.1.0-SNAPSHOT-2015-10-15
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
gpetit@HP-gpetit:~$ cd /home/gpetit/orbisgis-dist-5.1.0-SNAPSHOT-2015-10-15/
gpetit@HP-gpetit:~/orbisgis-dist-5.1.0-SNAPSHOT-2015-10-15$ ./orbisgis.sh

```

Note: You must have the rights to run this file. To do this, you can use the following command:

```
chmod ug+x ./orbisgis.sh
```

... and your file will be executable.

2.1.5 Ubuntu PPA repository

If you are using Ubuntu (or a Debian like distribution) you also have the possibility to connect to our PPA repository, available here : <https://launchpad.net/~orbisgis/+archive/ubuntu/orbisgis-unstable>

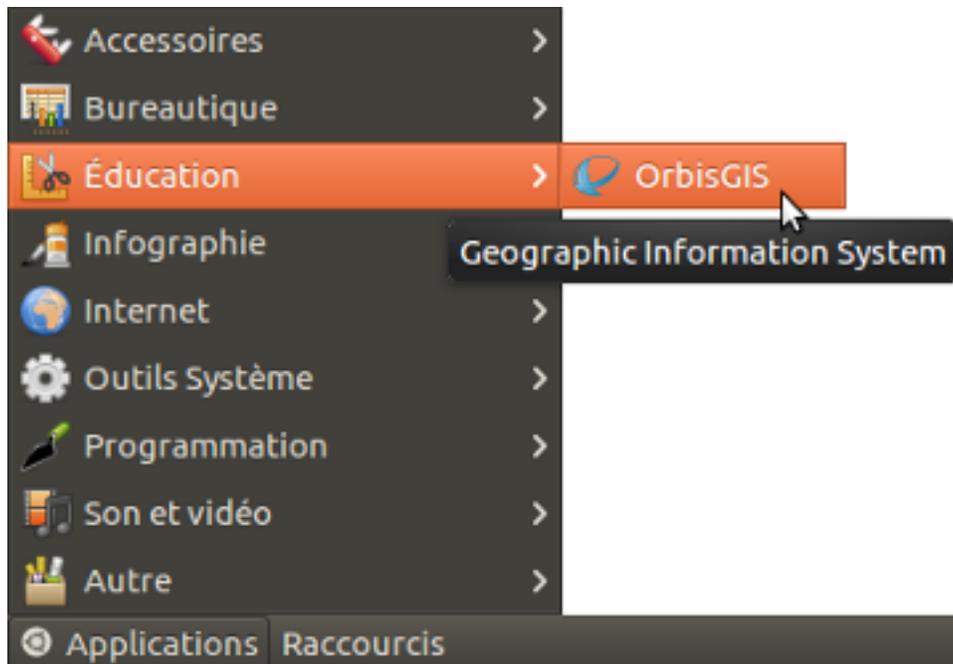
To add this repository and to install OrbisGIS (snapshot version), just execute the following command lines:

```

sudo add-apt-repository ppa:orbisgis/orbisgis-unstable
sudo apt-get update
sudo apt-get install orbisgis

```

Once done, you can launch OrbisGIS thanks to the shortcut in the “Applications/Education” menu



Or enter the following command in the terminal:

```
orbisgis
```

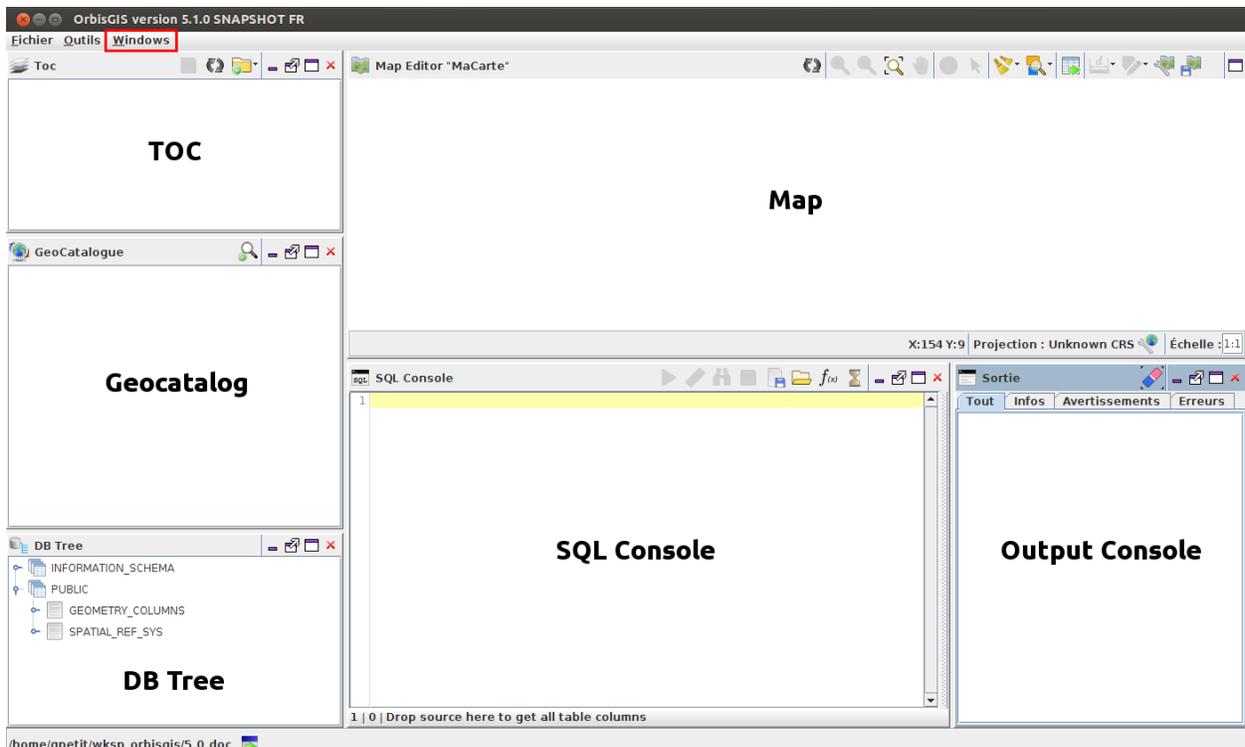
Note that you could also launch OrbisGIS in *safemode* with this command:

```
orbisgis -clean
```

2.2 Main Components

Here is a list of the main components of OrbisGIS. Except for the map, all these components are considered by OrbisGIS as plugins. So they can be:

- made visible or not (*just close the component to hide it*),
- resized (*using the cursor of your mouse*),
- moved (*drag & drop the component to a new place in the UI*),
- (un)docked (*pretty useful if you have a dual screen*),
- (un)installed (*thanks the plugin manager*).



To load a component, check the list in the “Windows” menu (*in red in the illustration above*) or the “Tools” menu (*for the SQL Console*).

2.2.1 Geocatalog

The Geocatalog () is the place where datasources are managed. Through a right-click in the blank area, the user can:

- Add a datasource (a file  or a folder ),
- Import a datasource (a file or a folder),

- Export layers into flat files or databases (),
- Show the attributes tables (),
- Remove a datasource (),
- Clear the Geocatalog : all the sources are removed from OrbisGIS,
- Refresh the list ().

2.2.2 Map

The map () is the place where geographic informations (vector or raster) are displayed. The user can navigate into the map (using zoom or pan tools), select and get informations on objects.

2.2.3 TOC

TOC () stands for “Table Of Content”. This component allows the user to:

- Make visible or not the layer with the checkbox (on the left),
- Change the layer name → make a double-click on the layer. Then change the name and press “Entrer”,
- Change the order of layers → drag & drop the layer upward or downward,
- Add WMS layers (),
- Add a layer group ().

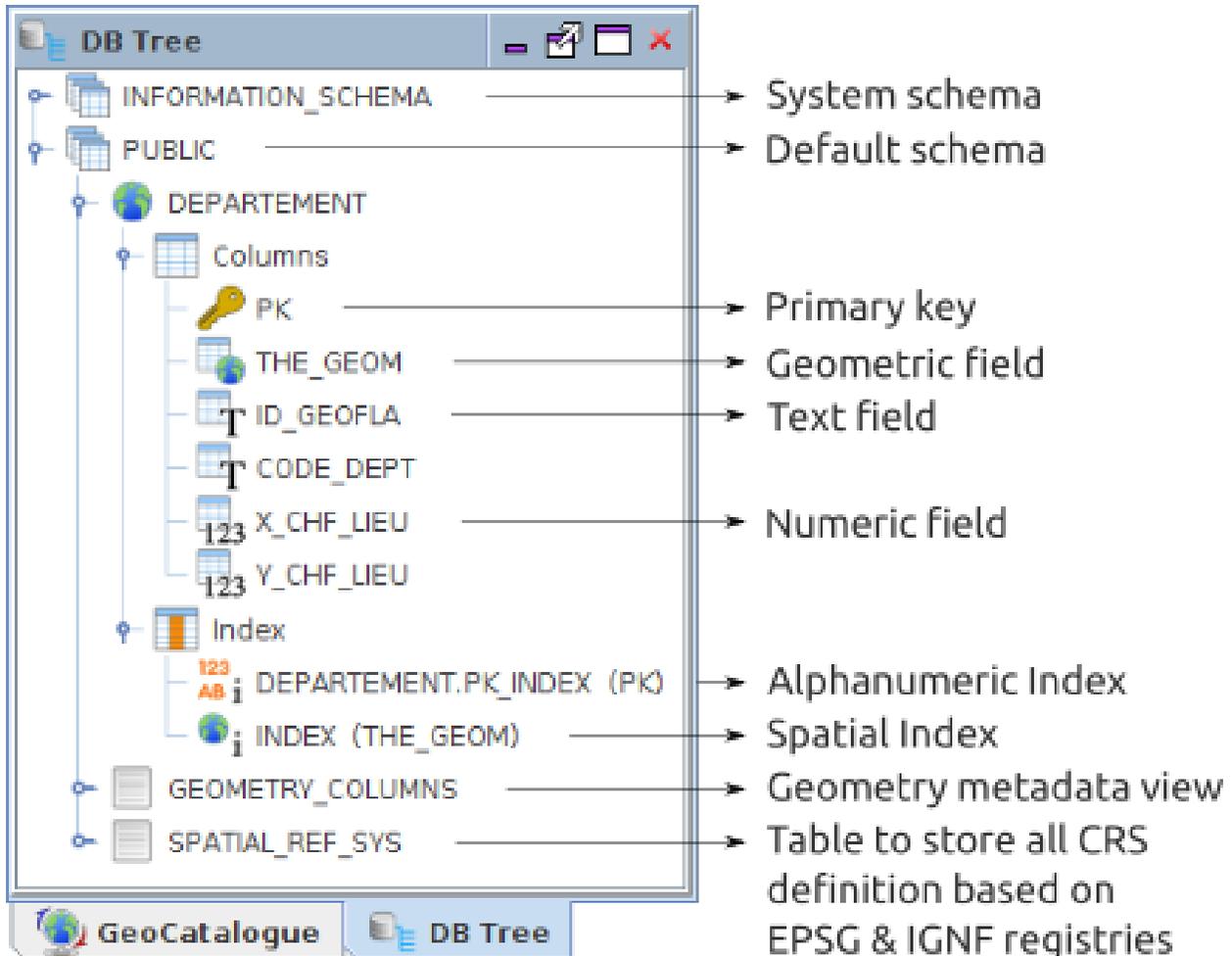
Moreover, several features are available via a right-click:

- Show attributes (): show the attributes’s table,
- Zoom to the layer (): Extended zoom on the layer,
- Import a style (), *
- Create a thematic analysis (), *
- Edit (), delete () or export () an existing style, *
- Remove the layer (): Remove the layer from the TOC,
- Start edition (): To edit a layer (e.g: *add a field, delete an object, ...*).

* see *Cartography*.

2.2.4 DB Tree

The DB Tree () is a tree representation of all the layers loaded in OrbisGIS. Layers (*tables*) are listed into their respective folder (*schema*).



Specific options allows the user to:

- Create spatial index (),
- Create alphanumeric index (),
- Remove columns ().

2.2.5 Output console

The output () console is the place where all messages (informations, warnings or errors) are displayed.

Logs can be copy and paste into a text editor.

2.2.6 SQL console

The SQL console () is the place where the user can process data thanks to SQL scripts. To know more about this component, read “*SQL Console*”.

2.3 Quickstart

2.3.1 The workspace

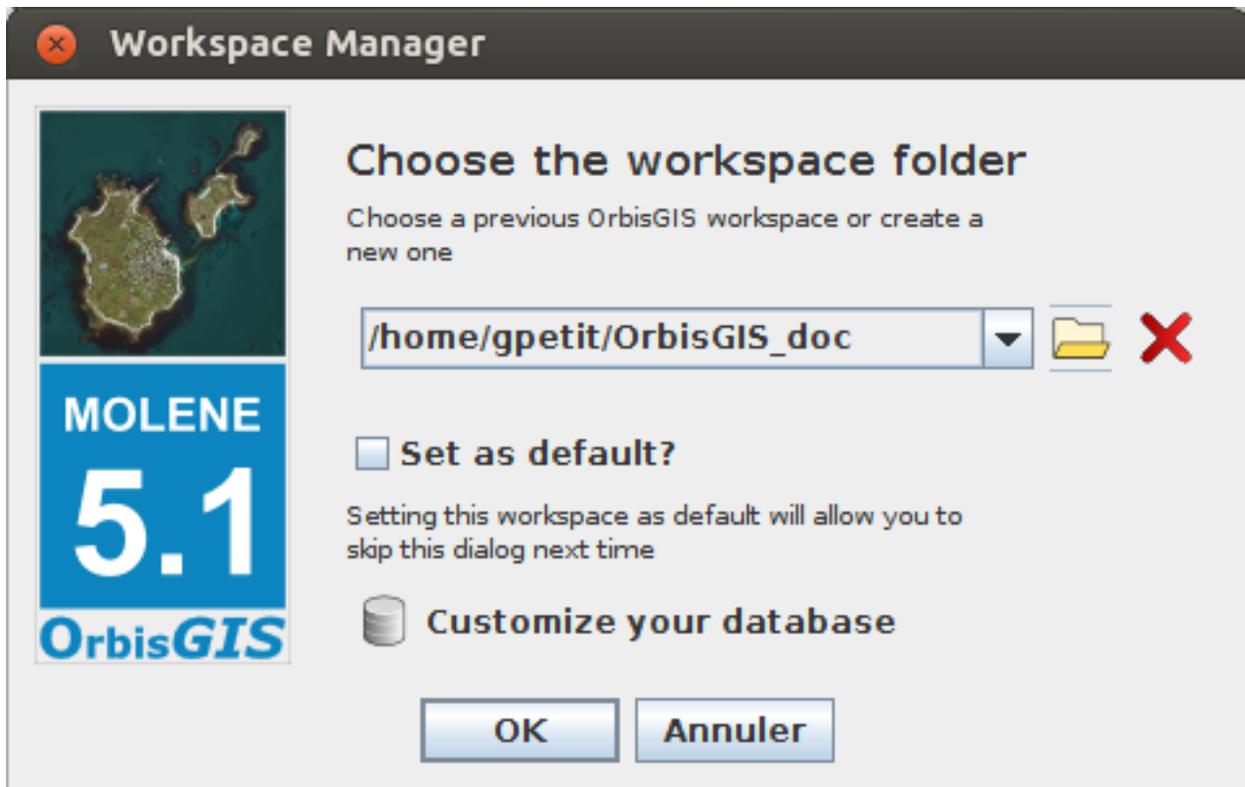
The workspace is the configuration of all your parameters and your User Interface (*UI*). You can create as many workspaces you want and switch them in a very quick way using *the workspace manager* .

The following items are saved in the workspace:

- Opened windows (with their size and place in the UI) (see *Main Components*),
- *Database* connection parameters,
- Map context file (loaded layers in the *TOC*),
- Toolbox parameters.

The workspace manager

When you launch OrbisGIS, the first panel displayed is the “Workspace Manager”. This interface, allows the user to create and choose workspaces.



- To create a new workspace, just click on  . In the new panel, create a new empty folder where you want, select it and press “OK”.
- You can select already created workspaces using the drop down list.
- If you check the “Set as default?” box, you will skip the Workspace Manager the next time you will launch OrbisGIS (not an irreversible action).
- If you want to configure your database, click on  (*Customize your database*) (see [Database](#)).
- If you **don’t** want to configure your database, just press “OK”. In this case, a default H2GIS db (named “database.mv.db”) will automatically be created in your workspace folder. The default login of this db is “sa” and there is no password.

2.3.2 Database

OrbisGIS is build on top of a relational spatial database. All data are managed and queried from a JDBC driver. By default OrbisGIS starts with the embedded spatial database [H2GIS](#). However, OrbisGIS could also be connected to a [PostgreSQL/PostGIS](#) database.

What does it mean ?

It mean that all the data you are using are interpreted as tables in OrbisGIS and can be processed thanks to the Spatial SQL langage. It also mean that if you choose to connect OrbisGIS to an existing PostGIS db, you will be able (1) to display and process your data, (2) to execute your PostgreSQL/PostGIS scripts directly into OrbisGIS, without any changes and (3) to use specific PostgreSQL/PostGIS functions.

Configure

To configure your database connection, click on  (*Customize your database*). The following panel will appear.

Database parameters

Saved connections: [dropdown] [refresh] [X]

Connection name: OrbisGIS_doc [save]

Database: H2GIS_EMBEDDED [dropdown]

Host: Not required

Port: Not required

Database name: /home/gpetit/OrbisGIS_doc/database

User name: sa

Require password:

Ok Cancel

Here you have:

- Saved connections: list of already existing connections,
- Connection name: the name of your connection (shortcut for all you parameters),
- Database: select the type of db you want to connect (*see below*),
- Host: only if the chosen db is not “H2GIS_EMBEDDED”,
- Port: *idem*,
- Database name: if “H2GIS_EMBEDDED”, URI of the local .db file. Else, the name of the remote db,
- User name: default value is “sa”,
- Require password: check if you want to specify a password to connect to your db.

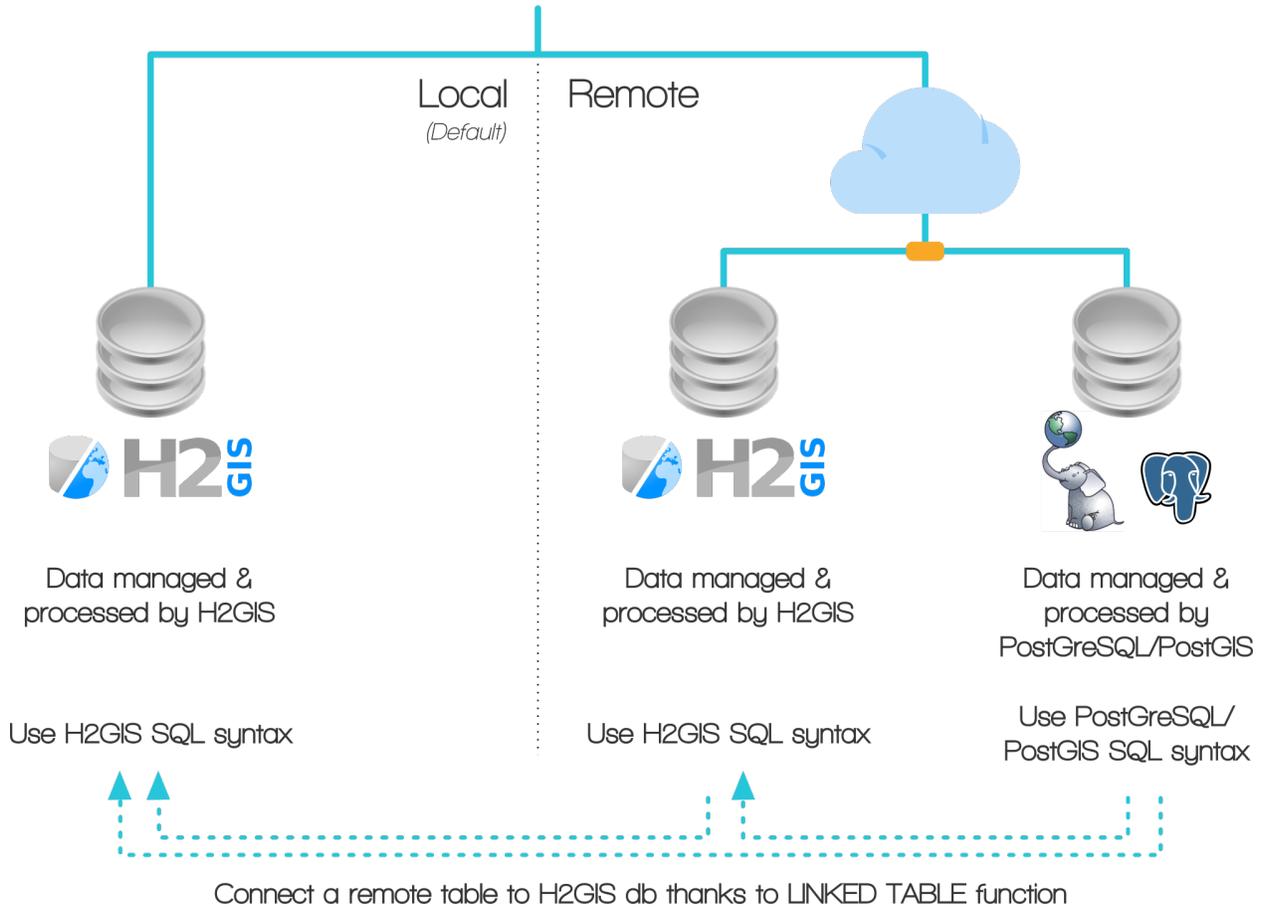
Database type:

- H2GIS_EMBEDDED: installed and deployed locally. By default the H2GIS db is stored in the workspace folder. This db may have been created by OrbisGIS, or by H2GIS,
- H2GIS_SERVER: remote H2GIS db,
- POSTGIS: remote PostGIS db.

The figure below summarizes this usage:



Load, display, process and style data coming from a database



Example with a PostGIS db

The screenshot shows a dialog box titled "Database parameters" with the following fields and controls:

- Saved connections:** A dropdown menu with a refresh icon and a red 'X' icon.
- Connection name:** Text input field containing "My_PostGIS_connection" with a save icon.
- Database:** Dropdown menu set to "POSTGIS".
- Host:** Text input field containing "my_host_uri".
- Port:** Text input field containing "5432".
- Database name:** Text input field containing "my_db_name".
- User name:** Text input field containing "my_login".
- Require password:** A checked checkbox.
- Buttons:** "Ok" and "Cancel" buttons at the bottom.

Once you open OrbisGIS with this db connection, you should see all your existing tables displayed in the *Geocatalog*.

2.3.3 Load data

To load data, just make a right-click in the *Geocatalog* and choose "Add" or "Import".

- **Add: Direct link to a data (vector, alphanumeric, ...),**
 - **Pro:**
 - * Immediate table creation,
 - * Does not take disk space,
 - * Can create spatial index.
 - **Cons:**
 - * Read-only table,
 - * Cannot create alpha-numeric index,
 - * Currently only compatible with h2 database.
- **Import: Copy the file content into a new database table,**
 - **Pro:**
 - * Create a native h2/PostGIS table,
 - * Read/Write table,
 - * Can create spatial index and alpha-numeric index.

– **Cons:**

- * Take time to create the table depending on file size,
- * Take more disk space.

For both options, you can choose:

- File: point one or many (using Ctrl + Click) files in a same folder,
- Folder: after selecting the type of file, point a folder. Here, selected type of file will be searched into the folder and potential sub-folders (*e.g.* open all .shp files present in the folder “GIS_Data”)

Formats

Below is the list of **readable** formats that you can choose in the UI, depending if you chose “Add” or “Import”.

Format	SHP	GPX	OSM	GeoJSON	CSV	DBF	TSV
Add	YES	NO	NO	NO	NO	YES	NO
Import	YES	YES	YES	YES	YES	YES	YES

Remarks:

- You can find a full description of readable and writable formats here : <http://www.h2gis.org/docs/dev/h2drivers/>
- Because OrbisGIS uses a RDBMS, a data is represented as a table and could be requested using the SQL language.

2.3.4 Display on the map

To display a geographic layer (vector or raster), just drag & drop it from the *Geocatalog* to the *TOC*.

2.3.5 Display attributes

To see how to display attributes, have a look at *Manage tables* page.

2.4 Manage data

OrbisGIS offers features to manipulate data.

2.4.1 Manage tables

To display the layer’s attributes, select it in the TOC (or in the Geocatalog) and make a right-click → “Open attributes”

().

A new tab appears in the UI with all the attributes.

The screenshot displays the OrbisGIS interface. At the top, the title bar reads "OrbisGIS version 5.1.0 SNAPSHOT FR". Below it, a menu bar contains "Fichier", "Outils", and "Windows". The main window is titled "COMMUNE - 1/36 595".

On the left, there is a "Toc" (Table of Contents) panel showing a tree view with "COMMUNE" selected. Below it is the "GeoCatalogue" panel, also showing "COMMUNE". At the bottom left, there is a "DB Tree" panel.

The central part of the interface is a table with the following columns: "PK" (Primary Key, marked with a key icon), "THE_GEOM", "ID_GEOFLA", and "CODE_". The table contains 7 rows of data, with row 4 selected. The data is as follows:

	PK	THE_GEOM	ID_GEOFLA	CODE_
1	1	MULTIPOLYGON (((499485 6265...	COMMUNE000000000000000001	216
2	2	MULTIPOLYGON (((514321 6384...	COMMUNE000000000000000002	033
3	3	MULTIPOLYGON (((474752 6280...	COMMUNE000000000000000003	009
4	4	MULTIPOLYGON (((902582 6453...	COMMUNE000000000000000004	225
5	5	MULTIPOLYGON (((639511 7027...	COMMUNE000000000000000005	890
6	6	MULTIPOLYGON (((825044 6910...	COMMUNE000000000000000006	018
7	7	MULTIPOLYGON (((462193 6302...	COMMUNE000000000000000007	113

Below the table is a search bar with "Trouver" and "Tout" buttons, and checkboxes for "Respecter la casse" and "Mots entiers".

The bottom part of the interface is the "Map Editor 'MaCarte' [Modifié]" panel. It shows a map with a yellow background and a blue polygon representing the selected commune. The map editor includes various tools like zoom, pan, and a toolbar. At the bottom of the map editor, there is a status bar showing coordinates "X:893893.936 Y:6456329.553", "Projection : Unknown CRS", and "Échelle : 1:231 597".

The bottom status bar of the application shows the file path: "/home/gpetit/wksp_orbisgis/5_0_doc".

Remarks

- On the top, the table name is displayed (in the illustration below "COMMUNE") with the number of (selected) objects,
- There is a dynamic link between a row in the table and the object displayed in the map. So, if you select a row in the table and make a right-click, you can zoom on the selection using the "Zoom to selection" tool (),
- Primary key is identified with a specific icon (),
- Numeric fields are aligned on the right. Other types of field (*string*, *boolean*, ...) are aligned on the left.

Operation on rows

With a right-click on a row, the user can access to several options:

- Filter selected rows () (non-selected lines will be hidden),
- Create a datasource from the selection (): export the selection into a new layer,
- Unselect (),
- Zoom to selection (),

- Reverse the selection (),
- Select rows with the same value: search all the lines that has the same value in the selected field ().

Operation on a column

With a right-click on a column (field), the user can access to several options:

- Sort ascending () / descending (),
- Unsort (),
- Show column information (),
- Show statistics (only for numeric fields) ().

Search engine

In the lower part of the window, there is a search engine that can be used in two ways:

- “Find” mode: enter a value, optionnaly choose a field and specify options,
- “SQL” mode: just write an SQL instruction and press the “Execute” () icon.

2.5 Cartography

OrbisGIS offers methods to produce maps using a cartographic language, which is managed in a style model. The style model is based on a research work that propose to extend the OGC [Symbology Encoding](#) standard. If you want to know more about this standard, have a look at this [website](#).

2.5.1 Symbology Encoding

In this standard, a style is defined with a set of elements:

- **Style: a style is composed of one or n rule(s)**
 - **Rule: a rule has one or n symbolizer(s)**
 - * Symbolizer: a way to render geometries

A style can be configured with:

- a name,
- a local langage (you can configure different languages in the same style),
- a title (*one per langage*),
- an abstract (*one per langage*).

A rule can be configured with:

- a name,
- a local langage (you can configure different languages in the same style),

- a title (*one per langage*),
- an abstract (*one per langage*),
- Min and/or Max scale of visibility.

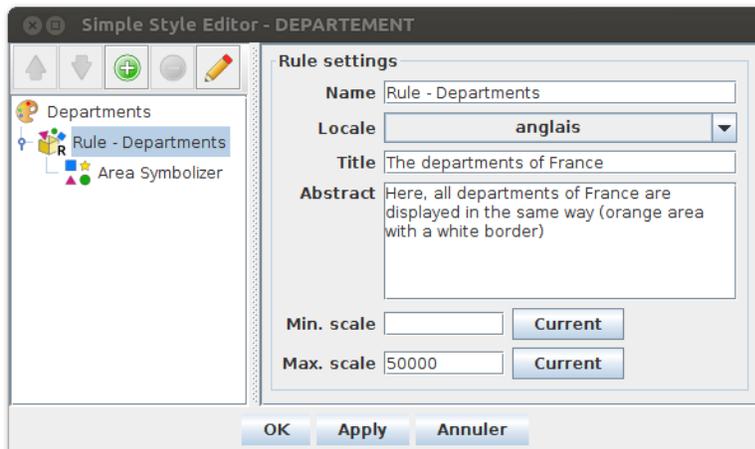
A symbolizer can be:

- a point,
- a line,
- a polygon,
- a text.

OrbisGIS proposes a set of pre-configured symbolizers to create maps:

- *Unique symbol*,
- *Value classification*,
- *Proportionnal symbol*,
- *Interval classification*.

Below is an example with a *Unique symbol* representation for the french departments. On the left side of the UI, we can see the three levels of the style (Style / Rule / Symbolizer). On the right side of the UI, we have the “rule settings” panel.



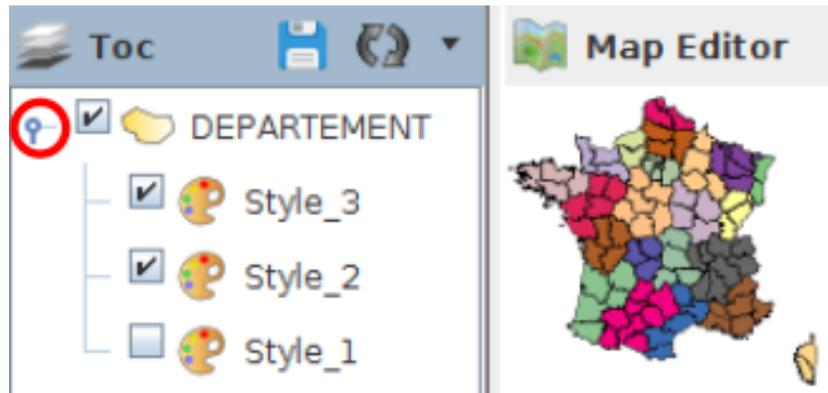
In OrbisGIS, styles can be exported (or imported) as an .se file, encoded in xml.

2.5.2 Manage styles

In the TOC () you have options to manage the style of your layer. Through a right-click on the layer, you can:

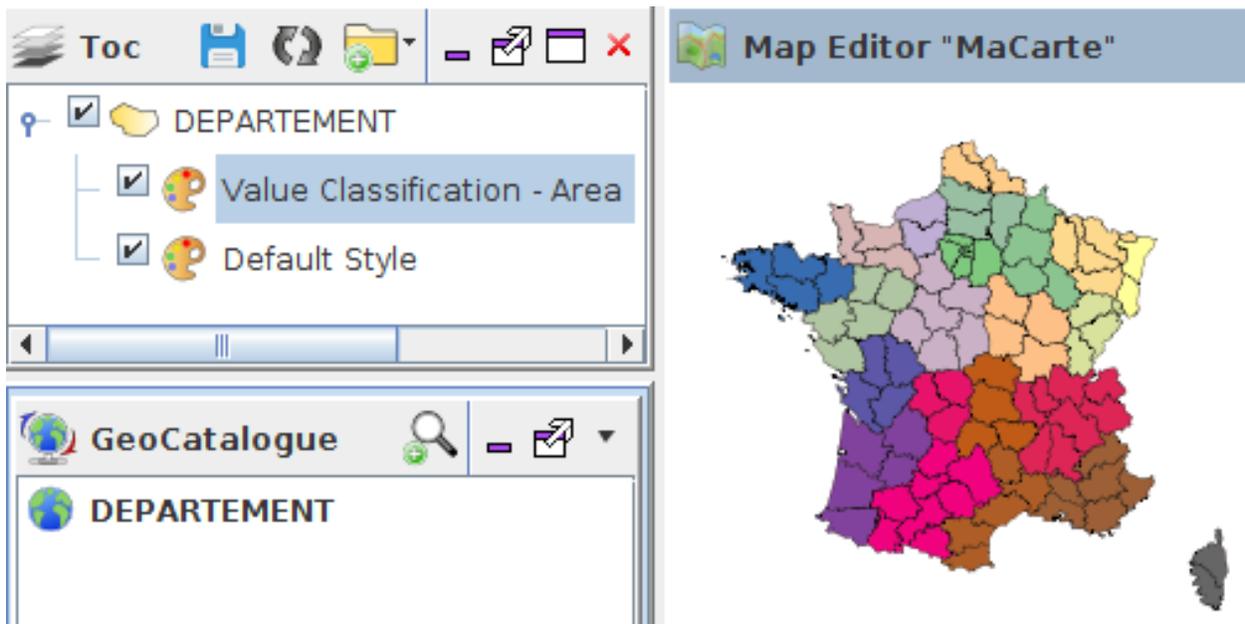
- Import a style (),
- Create a thematic analysis (): see below the list of available analysis (*Unique symbol*, *Value classification*, *Proportionnal symbol*, *Interval classification*),
- Edit (), delete () or export () an existing style,

You can also unfold (see red circle in the illustration below) a layer to display all the different styles that are associated (between 1 and n).



Remark : Each style can be (un)checked to be (un)visible.

By default, when loading the layer into the TOC, a *Unique symbol* style is applied. In the example below, two styles are associated to the DEPARTEMENT layer : one default and one *Value classification* (in the foreground).



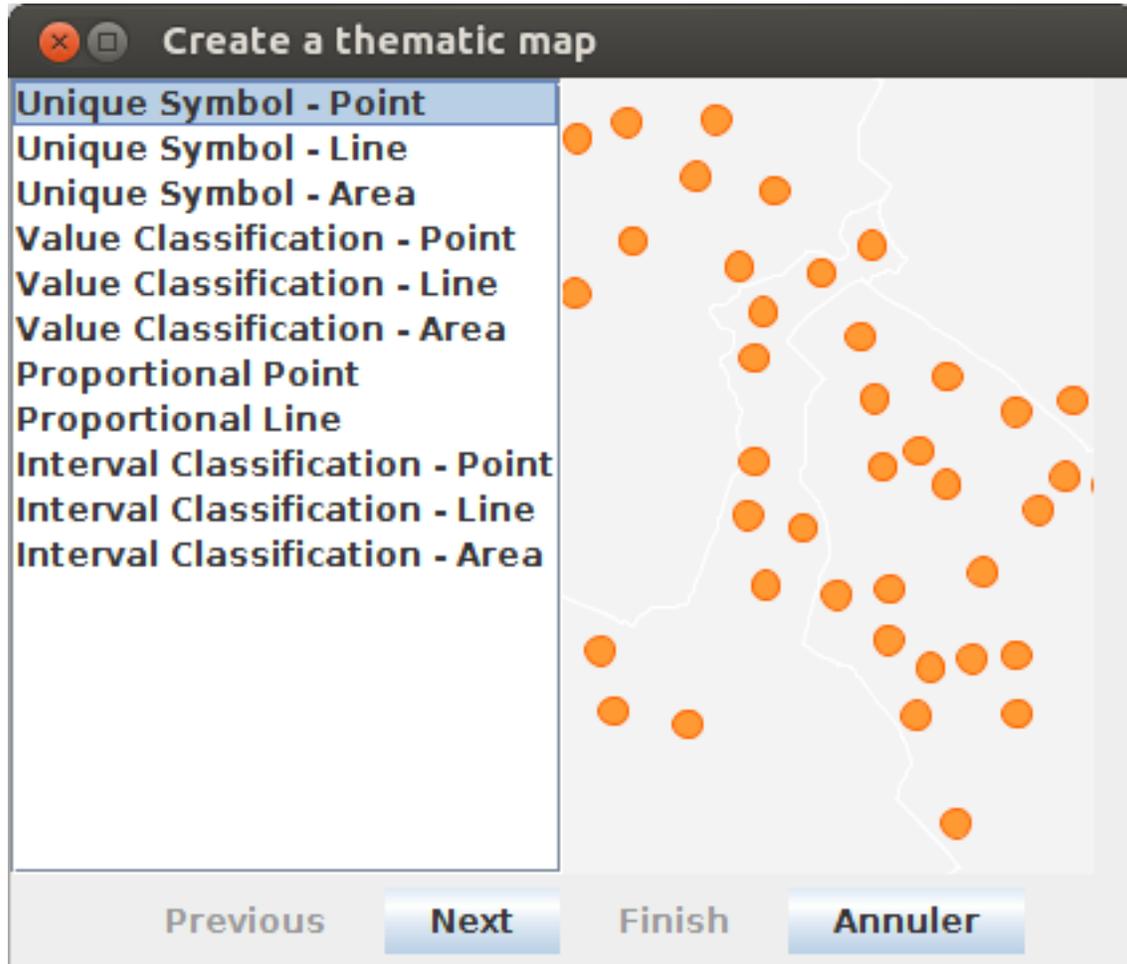
2.5.3 Unique symbol

In the layer, each objects have the same symbology (same border, fill, opacity, ...). Unique symbol works with point, line or area geometries.

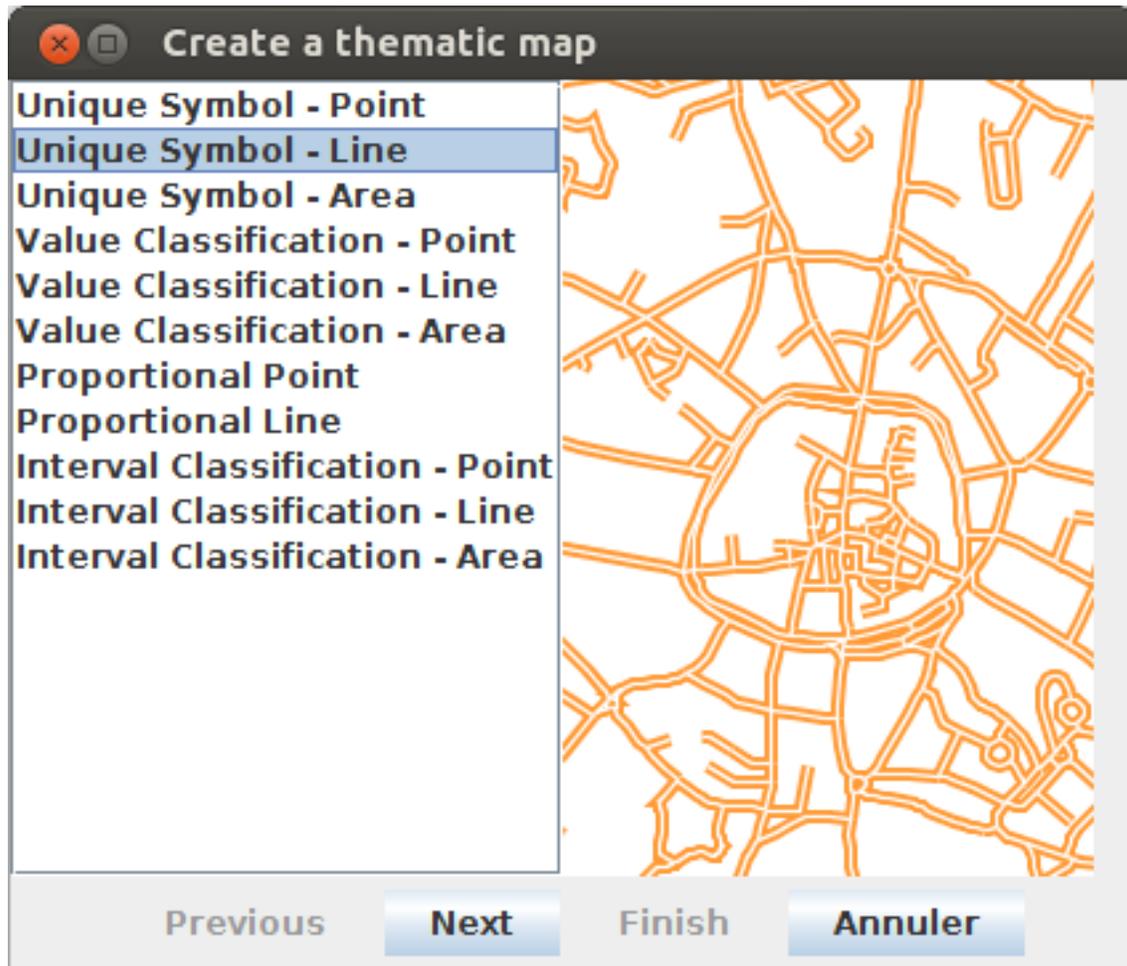
Depending on the geometry dimension, the user can manage:

- Fill settings: enable or not, color, opacity,
- Border settings: enable or not, color, width unit, width, opacity, dash array,
- Symbol settings (only for point): place (vertex or interior), shape (circle, half-circle, square, triangle, star, ...), size (width & height), size unit.

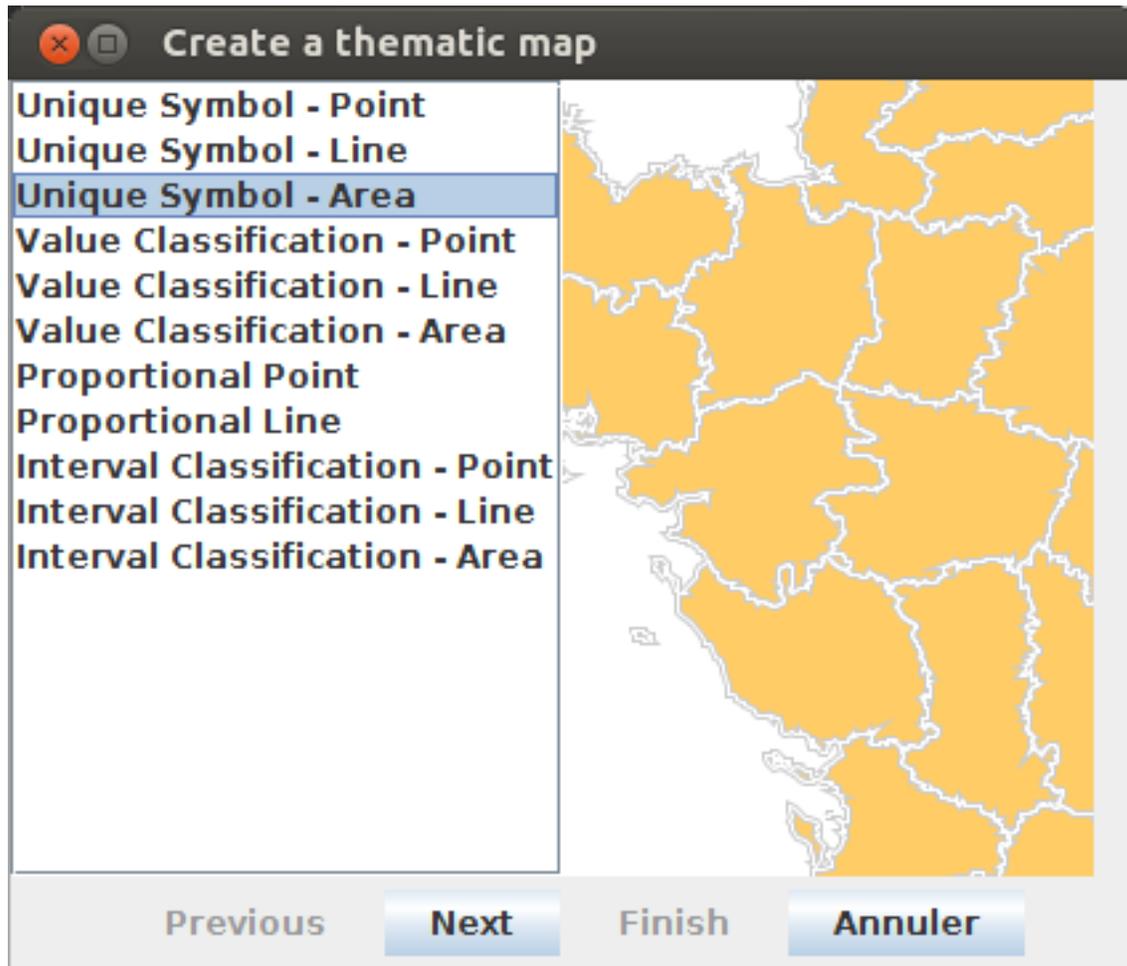
Unique symbol - Point



Unique symbol - Line



Unique symbol - Area

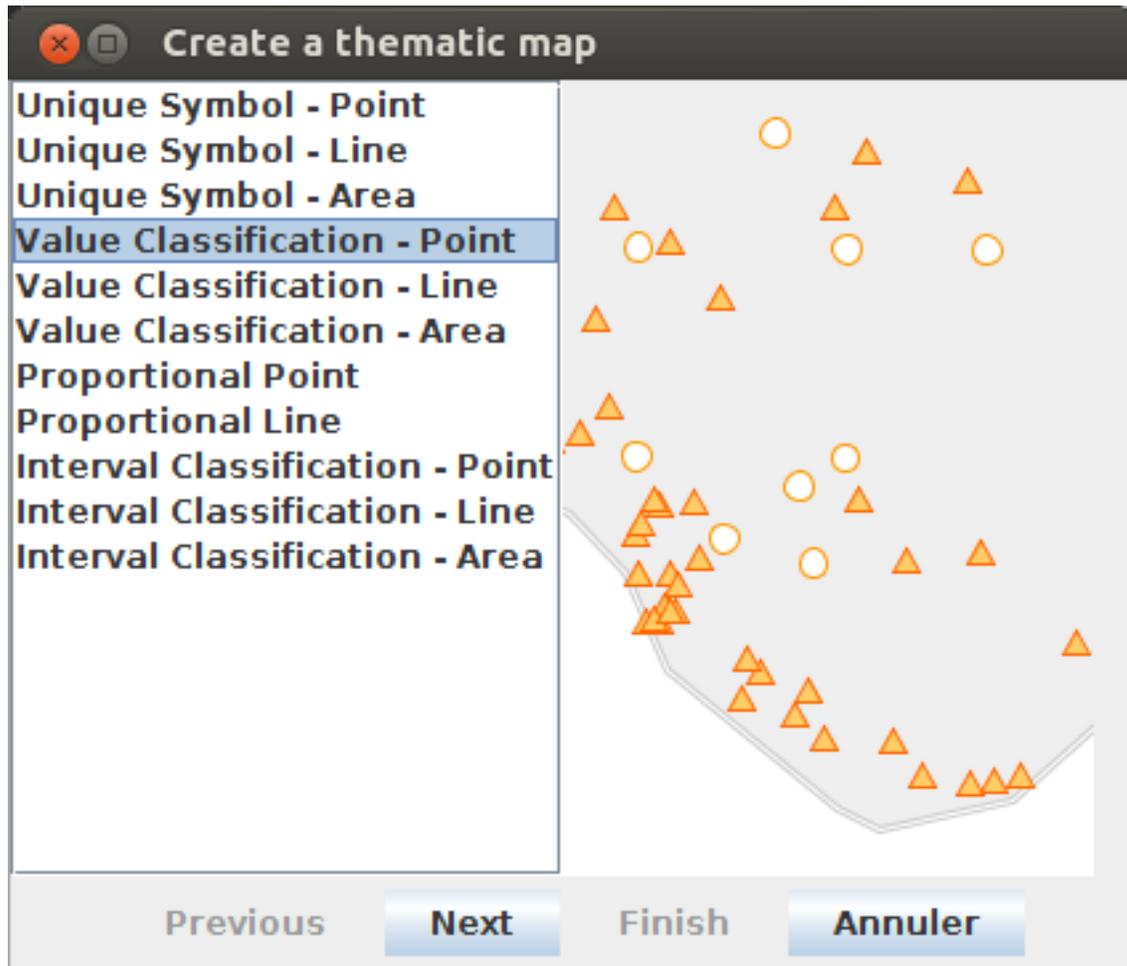


2.5.4 Value classification

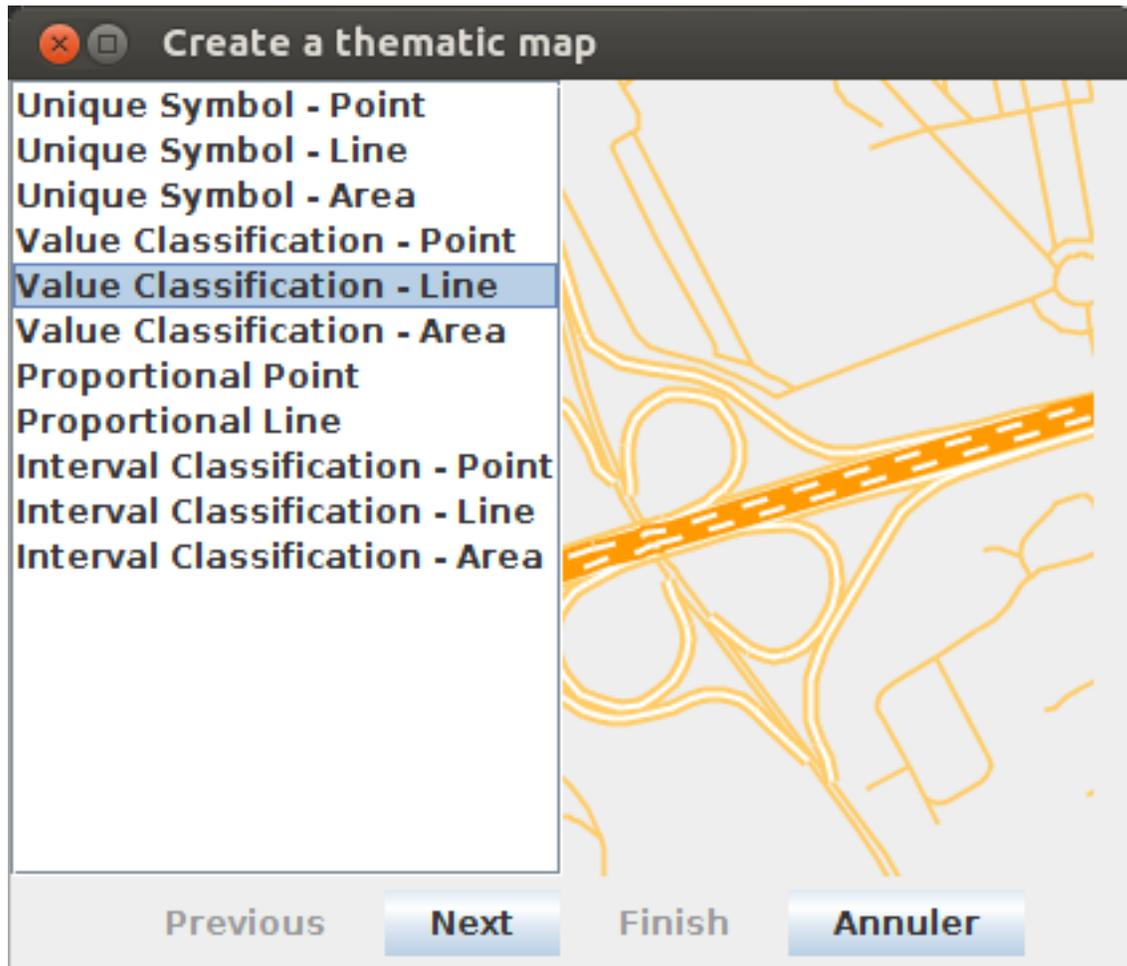
For one specific field, every objects that have the same value, have the same symbology. This classification is available for point, line or area geometries.

- The field used to make the classification can be numeric, text, boolean, ...
- For each classes, the user can configure the corresponding symbology (fill, border, ... see *Unique symbol*)

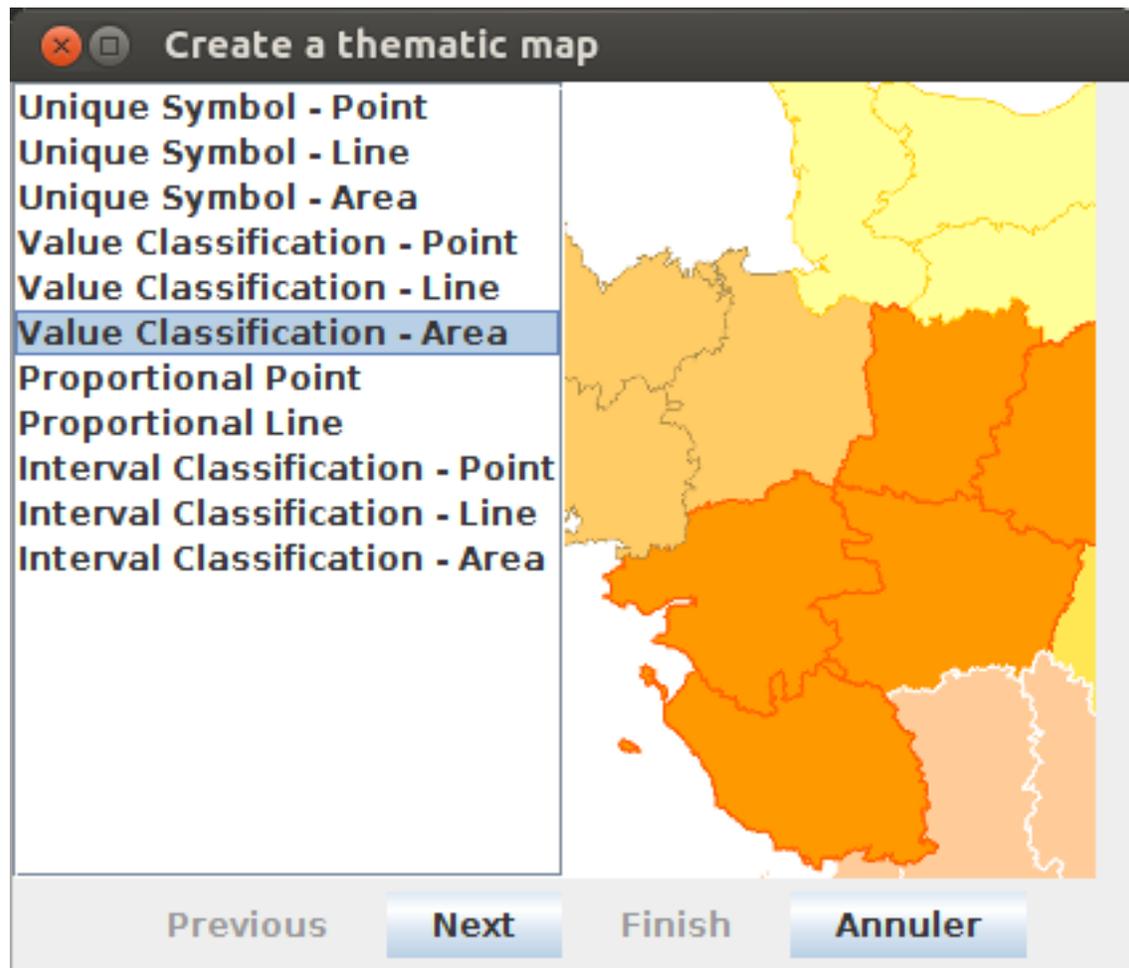
Value classification - Point



Value classification - Line



Value classification - Area



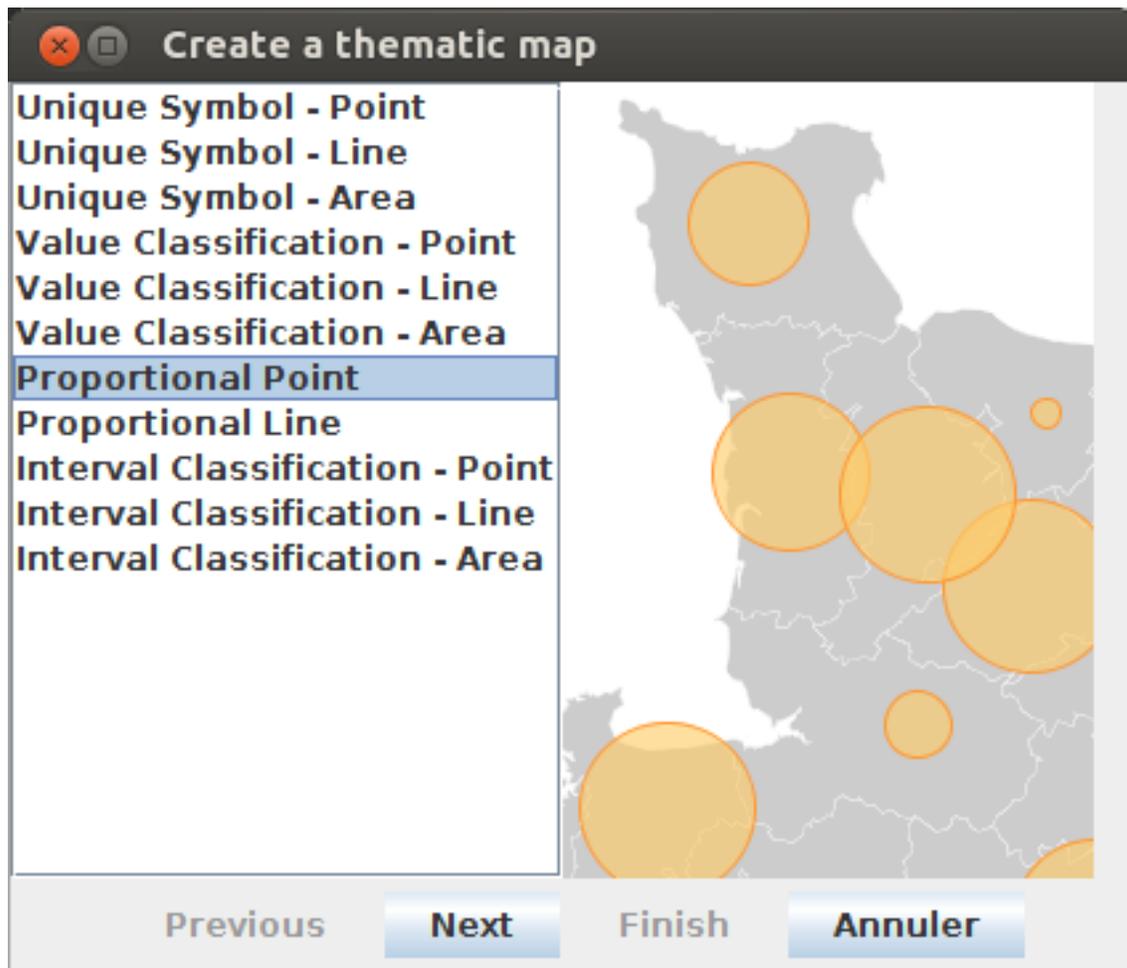
2.5.5 Proportionnal symbol

The size of an object depends on the value presented in a specific numeric field. This classification is available for point or line symbolizers.

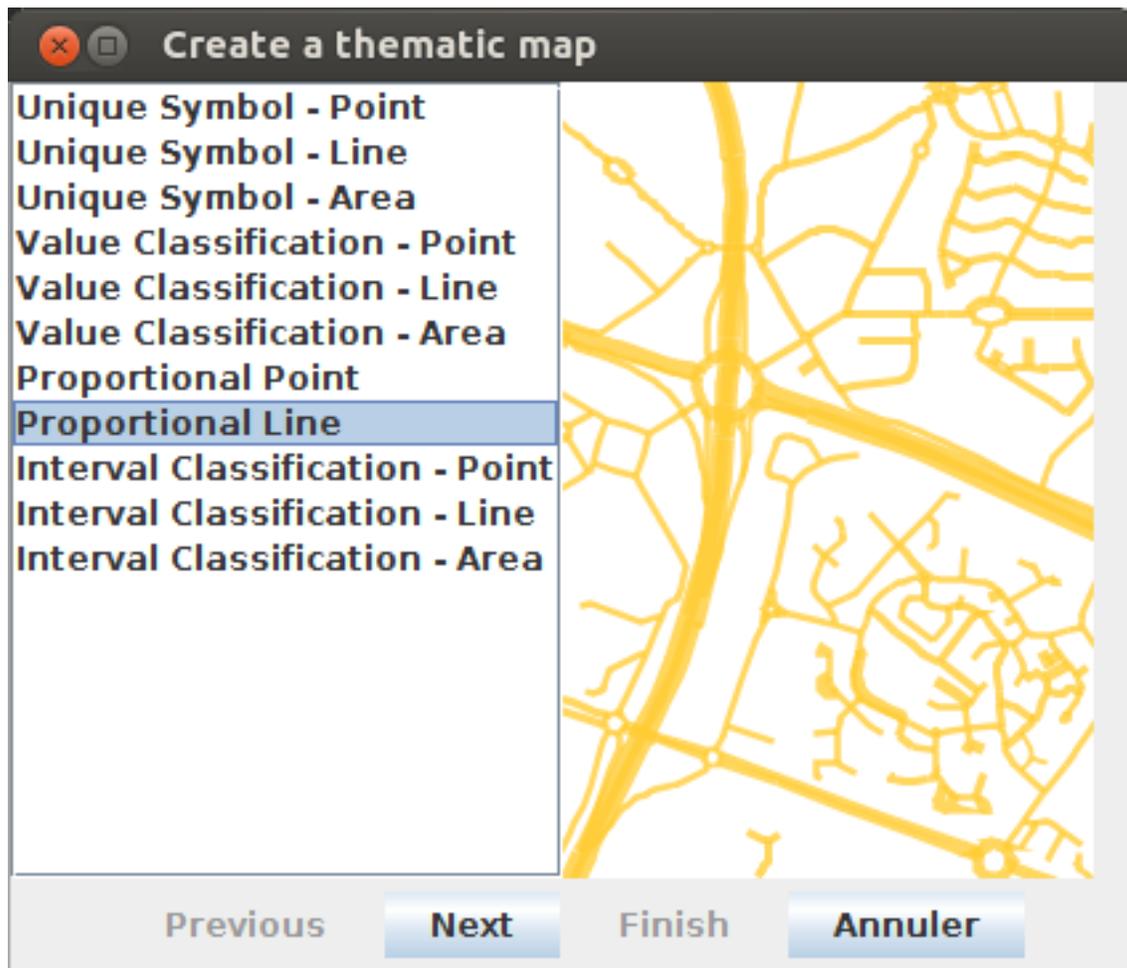
The user can manage:

- The field. Only numeric fields can be selected,
- The symbol: Shape, color, . . . , size (min & max) and place (only for point)

Proportionnal Point



Proportionnal Line

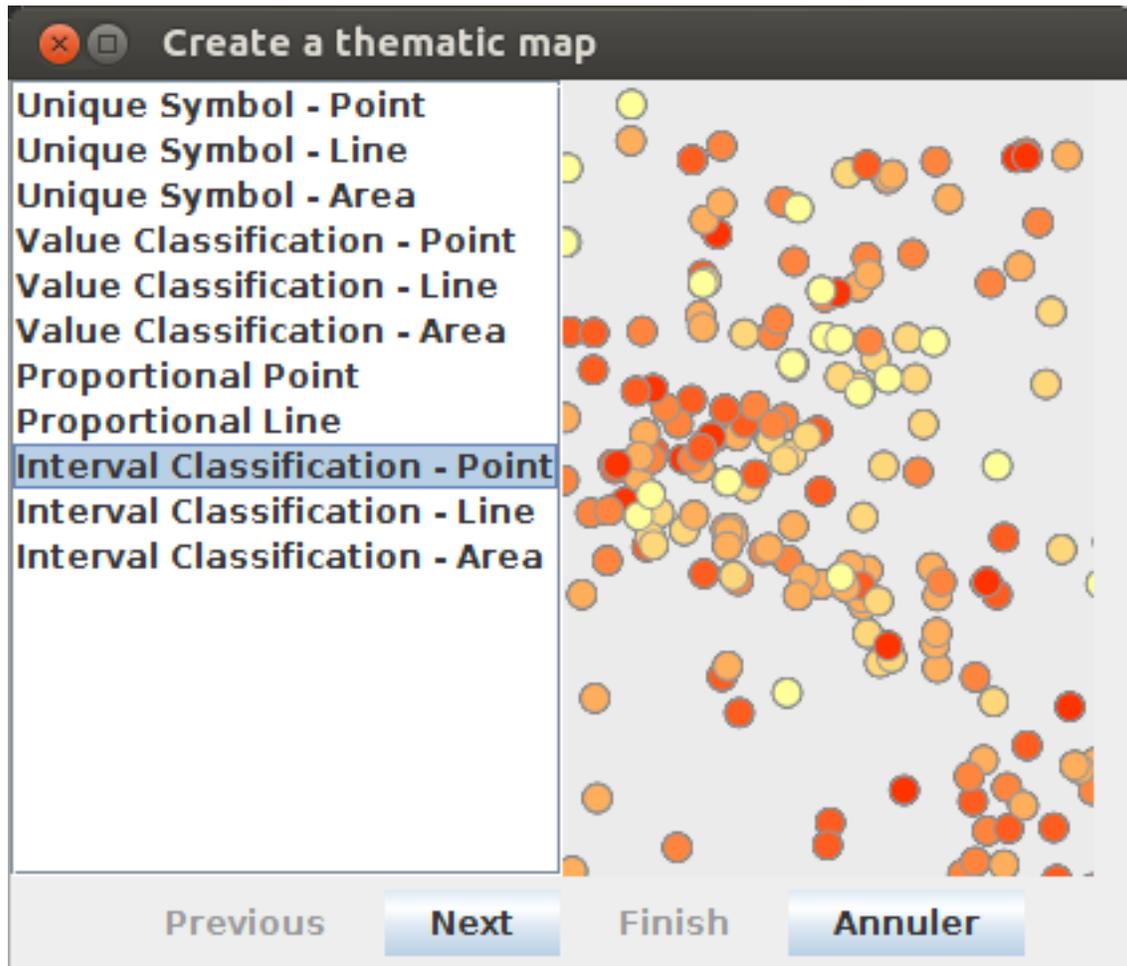


2.5.6 Interval classification

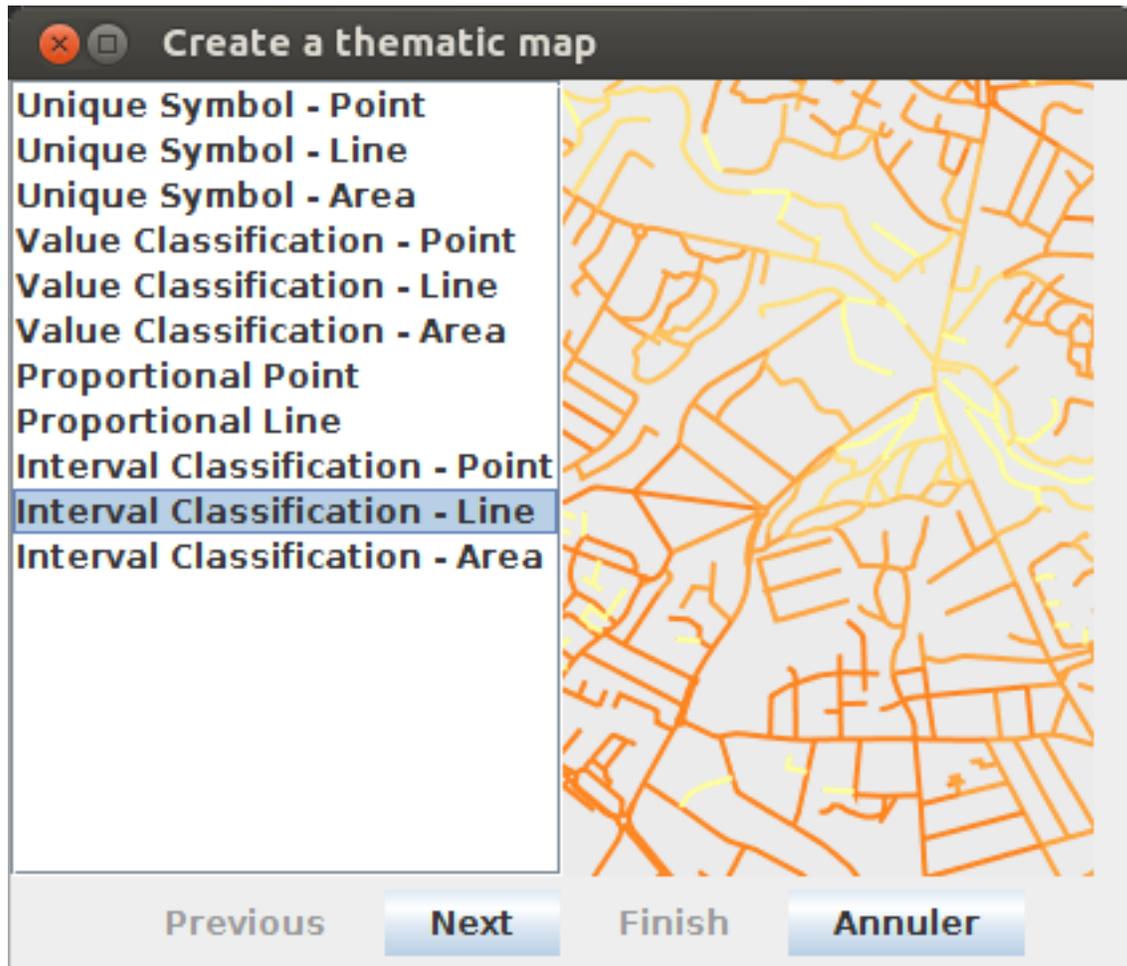
A numeric field is used to make a classification by intervals. The user is able to:

- manage the statistical method used to classify : Manual, Quantiles, Equal intervals, Mean - Standard deviation, Nested Averages, Jenks method,
- manage the desired number of classes,
- manage the color schema: choice between a gradient (from color x to color z) or a pallet,
- enable or not the fallback symbol,
- modify manually classes (thresholds, symbology, labels, add/remove classes).

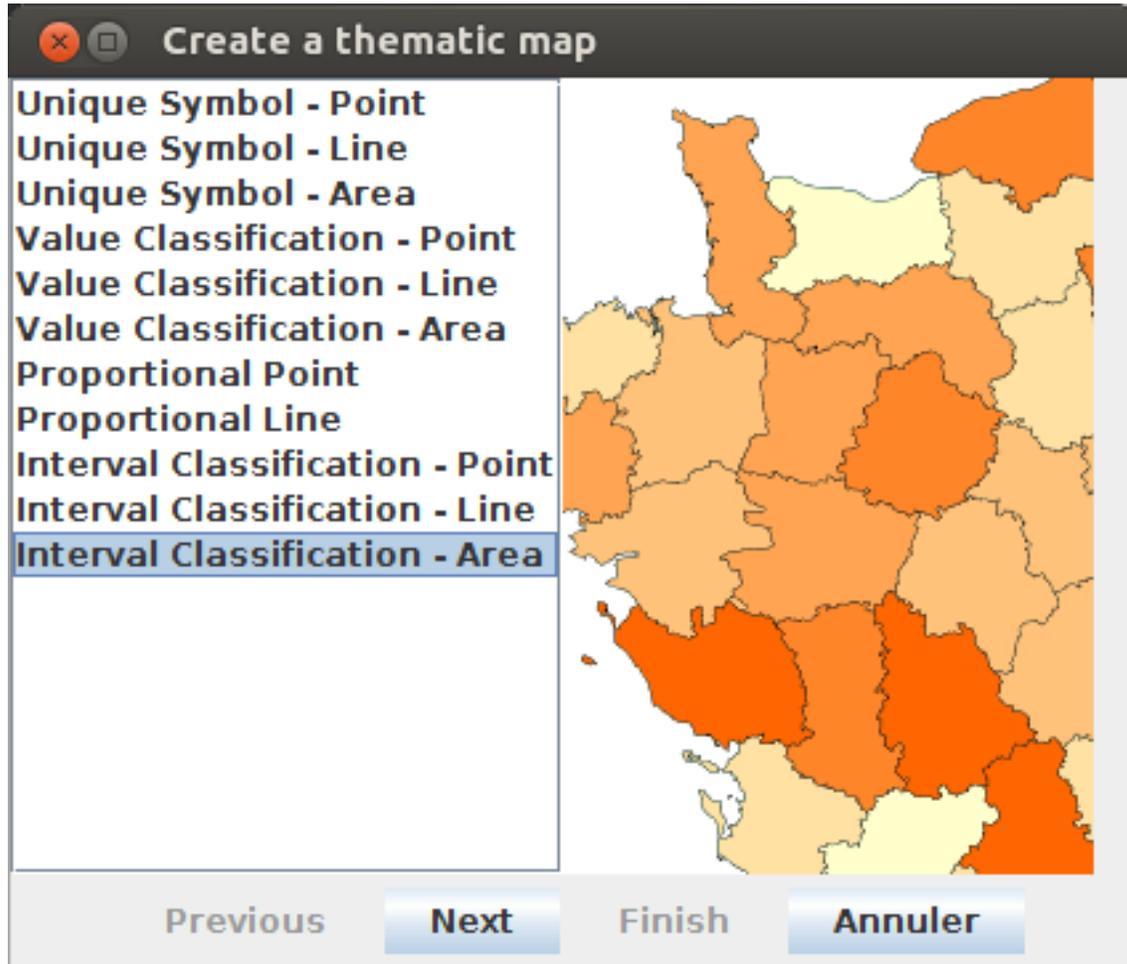
Interval Classification - Point



Interval Classification - Line



Interval Classification - Area



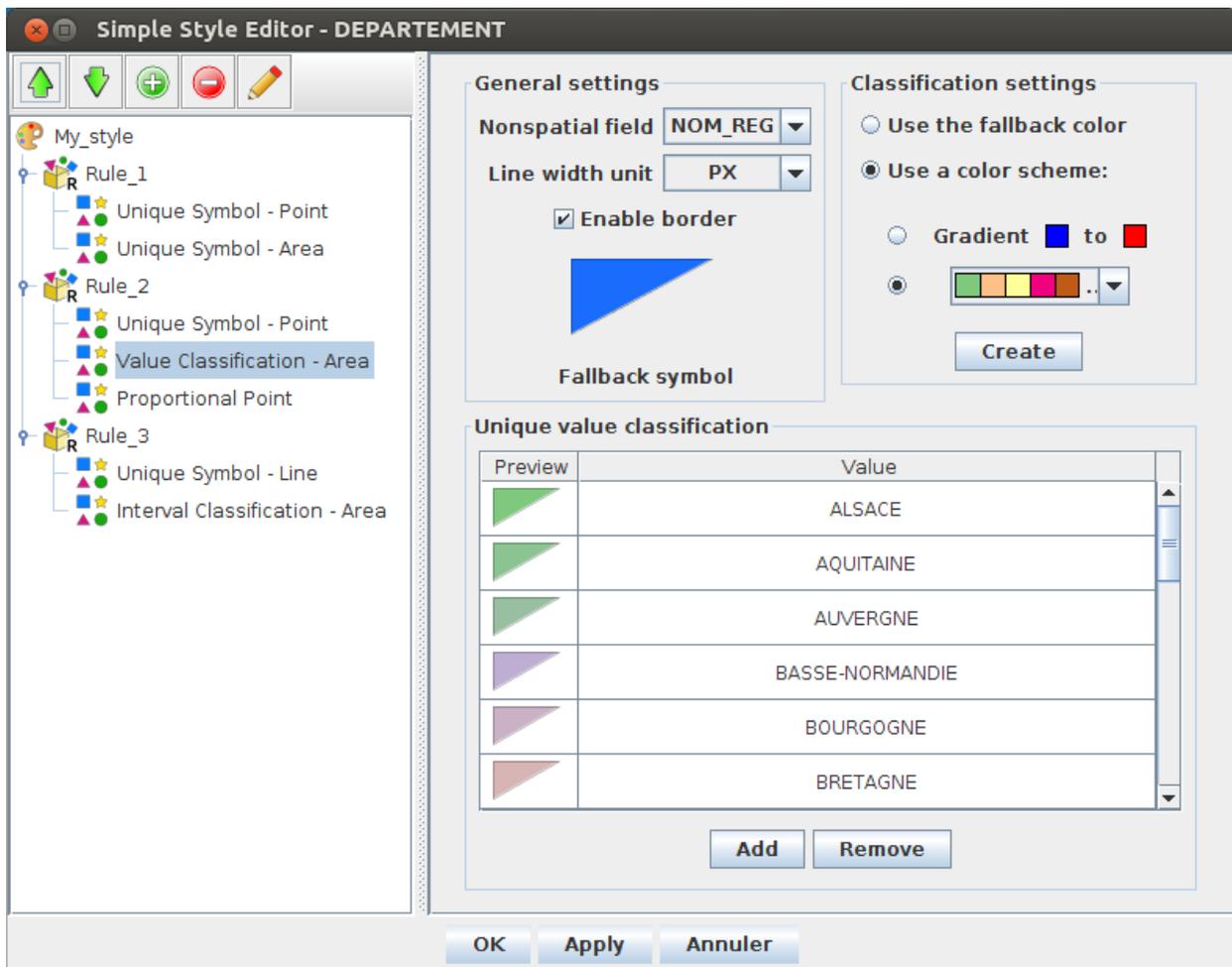
2.5.7 Complex styles

Thanks to the UI, the user is able to built “complex” styles, made with various rules and symbolizers.

In the example below, for one layer (DEPARTEMENT), we have one style composed of 3 rules in which there is different symbolizers:

- **Rule 1:**
 - Unique symbol - Point
 - Unique symbol - Area
- **Rule 2:**
 - Unique symbol - Point
 - Value classification - Area
 - Proportional point
- **Rule 3:**
 - Unique symbol - Line

– Interval classification - Area



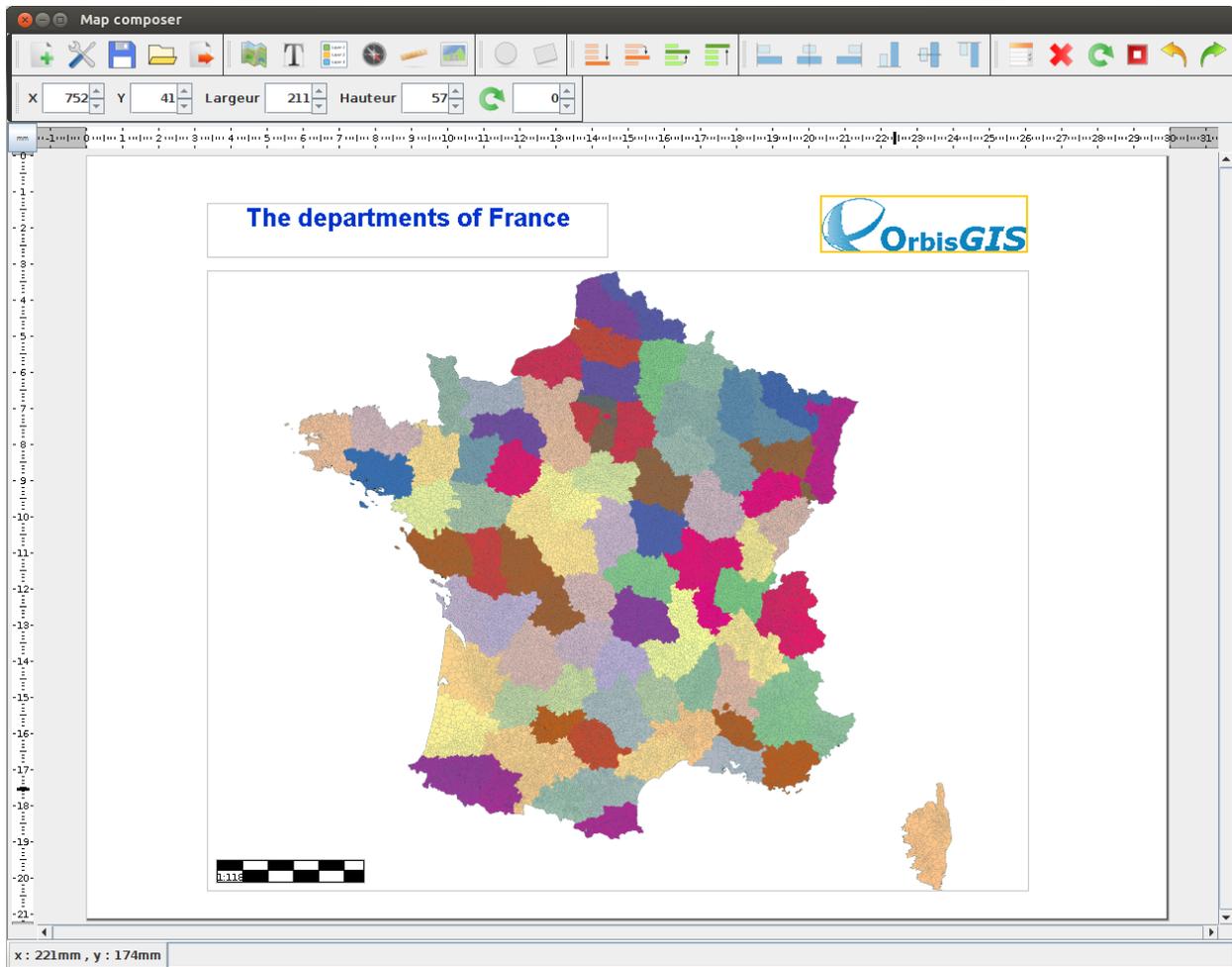
Using the toolbar, the user can:

- Move up () the symbolizer / rule,
- Move down () the symbolizer / rule,
- Add () a symbolizer / rule,
- Remove () the selected symbolizer / rule,
- Edit () the selected symbolizer / rule,

2.5.8 Compose & export maps

With the [Map Composer](#) plugin, user can create map layouts. A map layout is a collection of map elements organized on a page and designed for map printing. With The Map Composer you can add map elements such as text, orientation, scale. Once you have completed a map layout, you can print it or export it into .pdf or image files.

A detailed documentation is provided [HERE](#).



2.6 Plugins

OrbisGIS can be extended with new features thanks to its *Plugins manager*.

By default, some plugins are already installed (e.g Systems plugins, ...). In addition, user can optionally add some *Official plugins*.

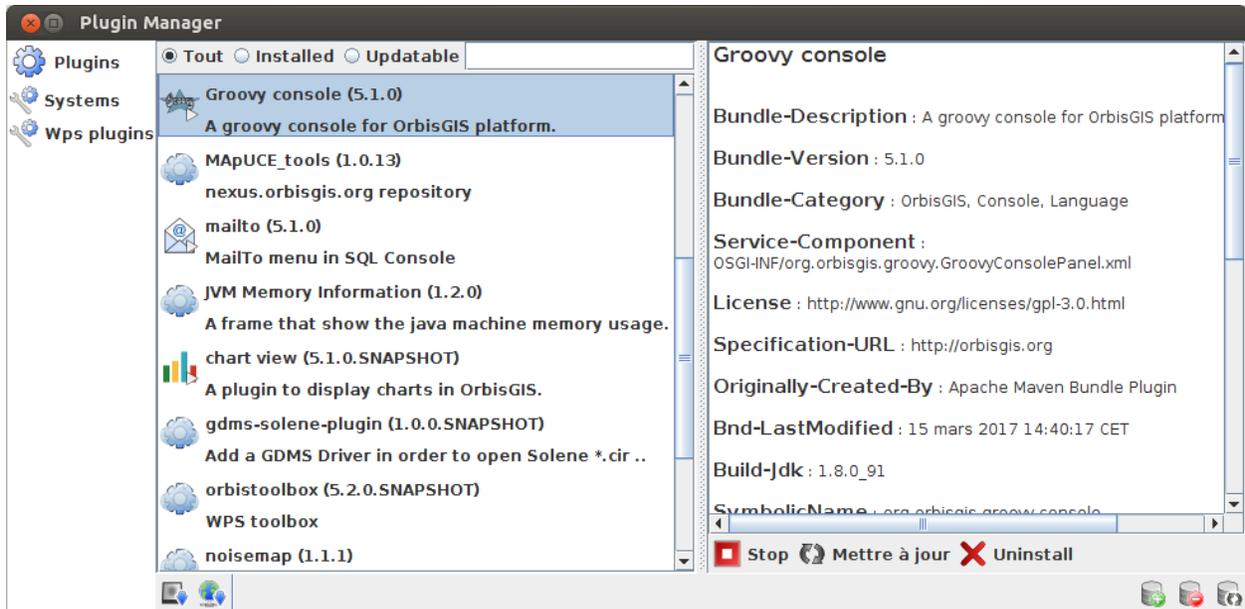
2.6.1 OSGi specification

Here, plugins are following the *OSGi* specification (*release n°4.3*). This systems allows developers to deploy their plugins in a simple and normalized way and to entrust the management of dependancies to OSGi.

To know more about this system, please consult the following documentation: [OrbisGIS plugin system](#).

2.6.2 Plugins manager

To install a plugin, you have to launch the “Plugin manager” which is in the “Tools” menu of OrbisGIS.



Here, users can:

- Search plugins thanks to the engine,
- Filter all, installed or updatable plugins,
- **Choose the type of plugins:**
 - Plugins: mainly dedicated to users,
 - Systems: mainly dedicated to developers,
 - WPS plugins: plugins following the OGC WPS standard and available in the “Toolbox”.
- **Add a plugin***
 - from the disk (): select an OSGI jar file,
 - from an url (): specify an url.
- Add () / remove () / update () a remote bundle repository,
- Download/start/stop/update/uninstall a plugin.

* in both cases, dependancies will not be automatically resolved.

When selecting a plugin, users can:

- Download () the plugin,
- Download and start () the plugin,
- Stop () the plugin,
- Refresh () the plugin,
- Uninstall () the plugin.

2.6.3 Official plugins

The following plugins are developed by the OrbisGIS team but are not deployed by default. To install them, just use the *Plugins manager*.

Those plugins are documented in their dedicated [GitHub](#) repository. Here, you can find:

- [ChartView](#)
- [Groovy Console](#) (see also *Scripting with Groovy*)
- [Mailto](#)
- [Map Composer](#)
- [JVM memory](#)
- [R Console](#) (see also *Scripting with R*)

... and probably more to come !

2.7 Scripting with SQL

2.7.1 SQL language

In OrbisGIS, you can process data using the Spatial SQL language. To do so, you have to execute your instructions into the *SQL Console*.

Here, all Spatial SQL functions and operators are based on the [H2GIS RDBMS*](#) and are mostly following the [OGC Simple Feature Access - SQL](#) standard.

* Remarks:

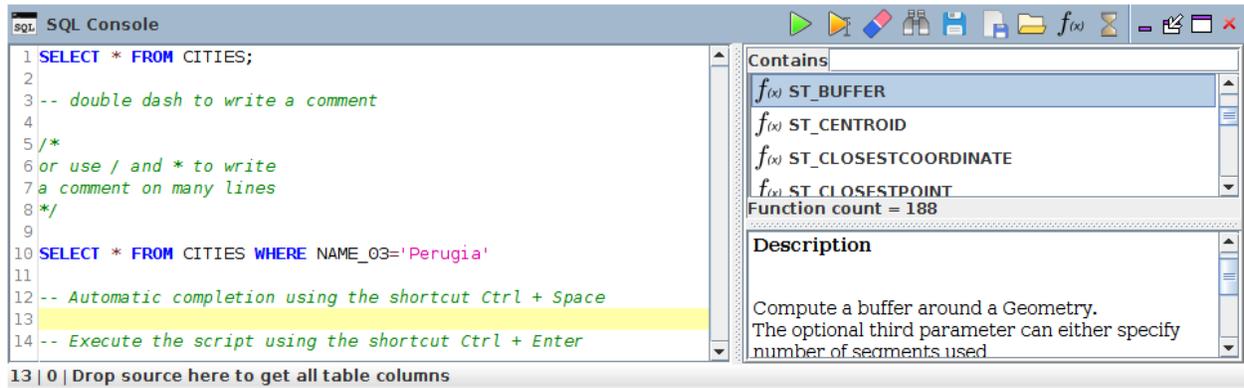
- *Only if you started OrbisGIS in H2GIS (local or remote) mode (see [Configure](#)).*
- *In this configuration, the “non-spatial” SQL functions and operators are executed by the [H2 Database Engine](#). Consult its quite complete documentation if you have any question about SQL.*

2.7.2 SQL Console

The SQL console () is the place where the user can execute SQL scripts. This component is available through the “Tools” menu.

Note that:

- You can open as many Console you want,
- The content of your Console can be automatically saved when you close OrbisGIS.



Here, the user is able to:

- Write & execute () instructions,
- Execute only selected instructions (),
- Load & Save .sql files,
- Search (spatial) SQL functions and operators into a predetermined list ($f(x)$) (the full list is available [here](#)),
- To drag & drop a function from the list to the text area, in order to see the default syntax of the chosen function,
- Search & Replace words (with advanced options) (),
- Share SQL instructions via email (need the “Mailto” plugin),
- Specify a query timeout (),
- To auto-complete instructions (see *Auto-completion*).

Remarks:

- You are able to open as many console you want,
- The content of the console is saved by default,
- There is syntax coloration that helps the user to read the instructions,
- You also have some additional functionalities thanks to a right-click in the text area.

Below are some other tooltips:

```

1 SELECT * FROM CITIES;
2
3 -- double dash to write a comment
4
5 /*
6 or use / and * to write
7 a comment on many lines
8 */
9
10 SELECT * FROM CITIES WHERE NAME='Perugia';
11
12 -- Automatic completion using the shortcut Ctrl + Space
13
14 -- Execute the script using the shortcut Ctrl + Enter

```

Auto-completion

To use the auto-completion, just press “Ctrl + Space”. For example, if you write “sel” and then press “Ctrl + Space”, OrbisGIS will recognize that you want to write “SELECT”. In the same spirit, if you start writing the name of a function, a table or field, a dropdown list will appear on the right to offer you the corresponding elements.

Below is an illustration video (*made for the V3.0 but works in the same spirit in the last version of OrbisGIS*).

2.7.3 Use cases

Below are some use cases to illustrate the power of Spatial SQL in OrbisGIS.

Detect holes into geometries

Here, we want to detect interior holes in a group of buildings (called *block*).

Method :

1. Merge buildings
2. Explode multi-geometries
3. Produce holes and keep only polygon information

SQL instructions:

```

1  DROP TABLE IF EXISTS BLOCK , BLOCK_EXPL, HOLES, BLOCK_HOLES ;
2  -- All buildings are merge into one, if there are 1cm far from the others
3  CREATE TABLE BLOCK AS
4      SELECT ST_UNION(ST_ACCUM(ST_BUFFER(THE_GEOM ,0.01))) as THE_GEOM
5      FROM BUILDINGS;
6
7  -- We explode this table in order to convert multipolygons into simple polygons
8  CREATE TABLE BLOCK_EXPL AS SELECT * FROM ST_EXPLODE ('BLOCK') ;
9
10 -- We remove the table BLOCK and we rename BLOCK_EXPL to be more clear
11 DROP TABLE BLOCK;
12 ALTER TABLE BLOCK_EXPL RENAME TO BLOCK ;
13 -- We add the "PK_BLOCK" id, which will be a primary key
14 ALTER TABLE BLOCK ADD COLUMN PK_BLOCK serial ;
15 CREATE PRIMARY KEY ON BLOCK (PK_BLOCK);
16 CREATE SPATIAL INDEX ON BLOCK (THE_GEOM);
17
18 -- We creates holes and we keep only surfacic geometries (dimension = 2)
19 DROP TABLE IF EXISTS HOLES, BLOCK_HOLES ;
20 CREATE TABLE HOLES AS SELECT ST_HOLES(THE_GEOM) as THE_GEOM, PK_BLOCK FROM BLOCK;
21 CREATE TABLE BLOCK_HOLES AS SELECT * FROM HOLES WHERE ST_DIMENSION(THE_GEOM)=2;

```

Result:



Spatialize GPS data

From a .csv file (see screenshot below), that contains GPS tracks of different peoples, recreate tracks and visualize the result in a map.

	A	B	C	D	E
1	Nom	Point	Latitude	Longitude	Altitude
2	Gwendall	1	47.218803	-1.533698	43.21855926513672
3	Gwendall	2	47.218803	-1.533698	42.467308044433594
4	Gwendall	3	47.218733	-1.534009	52.3936767578125
5	Gwendall	4	47.218733	-1.534009	66.9461441040039
6	Gwendall	5	47.218661	-1.534283	73.10115814208984
7	Gwendall	6	47.218651	-1.534375	77.83187103271484
8	Gwendall	7	47.218627	-1.534447	85.89417052222656

SQL instructions:

```

1  -- 1. Import .csv file into a table called MY_GPS_POINTS
2  DROP TABLE IF EXISTS MY_GPS_POINTS;
3  CREATE TABLE MY_GPS_POINTS AS SELECT *
4     FROM CSVRead('/my_url/gps_tracks.csv');
5
6  -- 2. Convert this table into a point layer (keeping all possible attributes)
7  DROP TABLE IF EXISTS MY_POINTS;
8  CREATE TABLE MY_POINTS AS
9     SELECT ST_MAKEPOINT(LONGITUDE,LATITUDE) as THE_GEOM, NAME, POINT, ALTITUDE
10    FROM MY_GPS_POINTS;
11
12 -- 3. Update Z value of points, using the Altitude field
13 UPDATE MY_POINTS SET THE_GEOM=ST_UPDATEZ(THE_GEOM, ALTITUDE);
14
15 -- 4. Generate a new layer with the different tracks stored as lines (one per people)

```

(continues on next page)

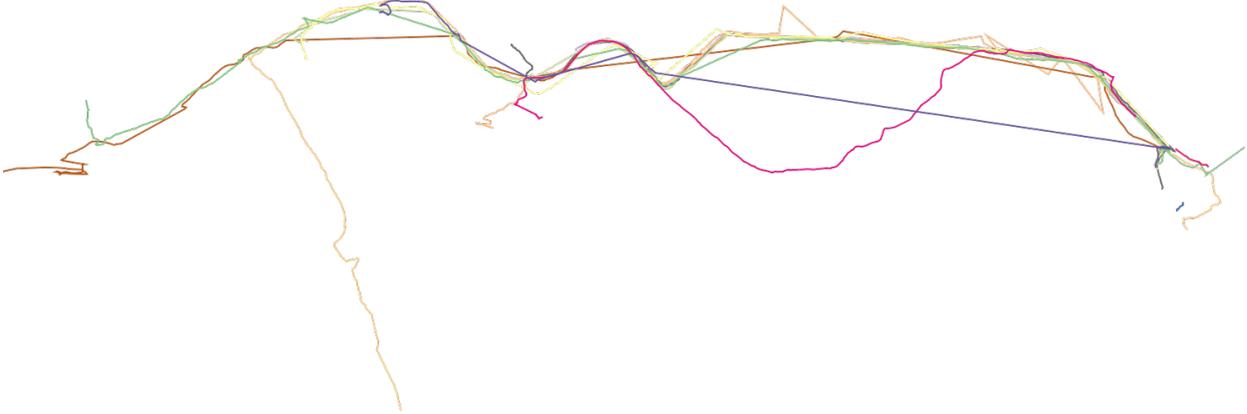
(continued from previous page)

```

16 DROP TABLE IF EXISTS LINES;
17 CREATE TABLE LINES AS SELECT ST_MAKELINE (ST_ACCUM (THE_GEOM)) as THE_GEOM, NAME
18 FROM MY_POINTS GROUP BY NAME;

```

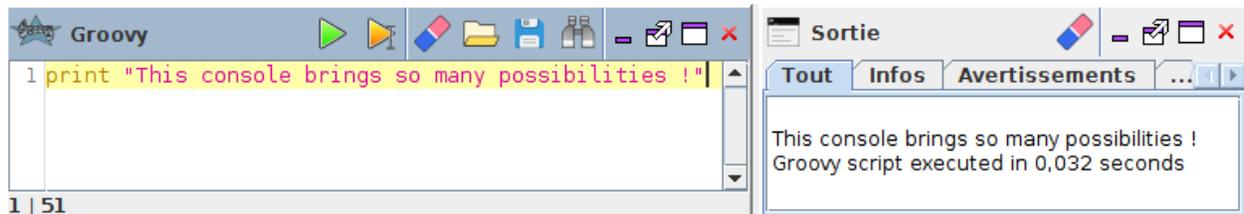
Result:



2.8 Scripting with Groovy

Thanks to its official “Groovy Console” plugin, OrbisGIS is able to execute instructions written in [Groovy](#) programming language.

To see how to install this plugin, you are invited to consult [this page](#) or just check for “Groovy console” in the *Plugins manager*.



Here, the user is able to:

- Write & execute () instructions,
- Execute only selected instructions (),
- Load & Save .groovy files,
- Search & Replace words (*with advanced options*) (),

2.8.1 Use cases

Include SQL into Groovy script

Below is an example of how to use SQL instructions into Groovy script. Here we just want to retrieve the coordinates of a point, which is stored in a layer called “TEST”.

```

1  import groovy.sql.Sql;
2
3  // Execute SQL request
4  sql.execute("DROP TABLE IF EXISTS TEST")
5  sql.execute("CREATE TABLE TEST AS SELECT 'POINT(1 3 10)>::geometry THE_GEOM")
6
7  // Fetch results and print point coordinates
8  // row["THE_GEOM"] is an instance of
9  // http://tsusiatsoftware.net/jts/javadoc/com/vividsolutions/jts/geom/Point.html
10 sql.eachRow("SELECT * FROM TEST") { row ->
11     pt = row["THE_GEOM"].getCoordinate()
12     print (pt.x+" "+pt.y+" "+pt.z)
13 }

```

The result is displayed in the Output Console:

```
1.0 3.0 10.0
```

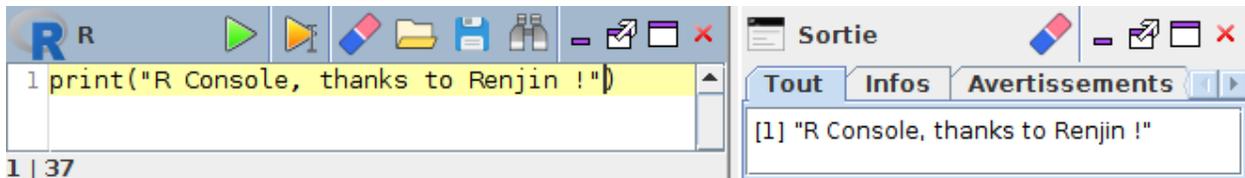
To see more about Groovy, you can consult this [tutorial](#) from the official Groovy documentation.

2.9 Scripting with R

Thanks to its official “R Console” plugin, OrbisGIS is able to execute instructions written in R language.

To see how to install this plugin, you are invited to consult [this page](#) or just check for “R console” in the *Plugins manager*.

The “R console” is based on [Renjin](#). Renjin is a JVM-based interpreter for the R language for statistical computing.



Here, the user is able to:

- Write & execute () instructions,
- Execute only selected instructions (),
- Load & Save .r files,
- Search & Replace words (*with advanced options*) (),

2.9.1 Use cases

List tables stored in the default H2GIS database

In this example, we are using R to list available tables imported in OrbisGIS (*assuming you are in H2GIS mode*).

```
1 library(RH2GIS)
2 print(dbListTables(con))
```

The result is displayed in the Output Console:

```
[1] "COMMUNE"           "SPATIAL_REF_SYS"   "TRONCON_HYDROGRAPHIQUE"
[4] "TRONCON_ROUTE"    "ZONE_OCCUPATION_SOL"
```

The screenshot displays the OrbisGIS software interface. At the top, there is a menu bar with 'Fichier', 'Outils', and 'Windows'. Below it is a toolbar with various icons. The main window is divided into several panes:

- R Console:** Shows the execution of the R script:


```
1 library(RH2GIS)
2 print(dbListTables(con))
```
- Sortie (Output Console):** Displays the results of the R script:


```
[1] "COMMUNE"           "SPATIAL_REF_SYS"   "TRONCON_HYDROGRAPHIQUE"
[4] "TRONCON_ROUTE"    "ZONE_OCCUPATION_SOL"
```

 Below the output, it states: "R script executed in 0,033 seconds".
- Map Editor "MaCarte" [Modifié]:** Shows a map of a region with several layers. The layers are:
 - TRONCON_ROUTE (green lines)
 - TRONCON_HYDROGRAPHIQUE (blue lines)
 - ZONE_OCCUPATION_SOL (yellow areas)
 - COMMUNE ("COMMUNE") (grey areas)
- GeoCatalogue:** A list of available tables:
 - COMMUNE
 - TRONCON_HYDROGRAPHIQUE
 - TRONCON_ROUTE
 - ZONE_OCCUPATION_SOL
- GeoCatalogue DB Tree:** A tree view of the database structure.
- Status Bar:** Shows coordinates (X: 224147.473, Y: 6762566.814), Projection: Unknown CRS, and Échelle: 1:465 830.
- Bottom Bar:** Shows the file path: /home/gpetit/OrbisGIS_doc_h2gis.

Contributors documentation

This part is mainly oriented to contributors (developers and translators). Some useful informations and examples are presented.

3.1 Develop

3.1.1 Source code

OrbisGIS is a free and open-source GIS software, released under the [GPL v3](#) licence. You can access to the source code on the dedicated [GitHub](#) repository.

Developers are cordially invited to contribute to OrbisGIS, posting:

- Issues: detect a bug, ask for enhancement ? [Create an issue](#) on GitHub,
- [Pull Request \(PR\)](#): want to contribute ? Please submit your PR and we will analyze it carefully.

3.1.2 Tools for developers

The following two “Continuous Integration” tools are used to build, control and deploy OrbisGIS source code:

- [Jenkins server](#)
- [Travis](#)

3.1.3 Releases & Snapshot

- Releases and pre-releases versions can be found [HERE](#)
- **Daily snapshot release (*for test purpose*) is provided via:**
 - a [.zip](#) file (*works on Linux, Windows and macOS*),

- a `.deb` file (for *amd64* architecture / also available for *i386*),
- the *Ubuntu PPA repository*.

3.1.4 Wiki documentation

Below are some useful links for developers, that are mainly provided on the [wiki](#) page of the OrbisGIS GitHub repository.

- [How to build OrbisGIS ?](#)
- [Debugging OrbisGIS with maven](#)
- [OrbisGIS and Nexus](#)
- [Release on Maven Central](#)
- [OrbisGIS internals](#)
- **OrbisGIS plugin system**
 - [Create your plugin](#)
 - [Resolve your plugin's dependencies](#)

3.1.5 Additional informations

- Made by a team from [CNRS](#)
- Registered date : 22/12/2007
- Langage of development : 100% Java
- Licence : [GPL v3](#)
- Supported OS : Linux, Windows and Mac OS
- Installation requirement : at least Java 7

3.2 Translate

The translation of OrbisGIS is managed thanks to the [Transifex](#) platform, a free-to-use website enabling users to collaboratively translate open-source projects. To do so, you just have to create a free account. Then, you can access to the Transifex OrbisGIS project [HERE](#).



3.2.1 Team

A language is managed by a team, which is composed of 3 roles:

- coordinator: manage the team (add/remove members),
- revisor: can translate and validate the result,
- translator: can translate.

Remark: note that one user can have the 3 roles in the same time.

New users are invited to:

- create a new team if their language is not already present,
- or to join an existing team.

3.2.2 Getting started (for translators)

Once you are team member, you should be allowed to translate the application. Here you are asked to translate words (also named keys) presents in .xml files called *ressources*.

Most of the translation keys are single words or sentences. In this case, the translation process is self-explanatory. However, you can discover some specific characters such as `{}` or `&`. In this case, please follow the instructions below.

1- Curly braces

In some cases, you will notice a number enclosed by curly braces. These strings refer to special parameters in the code and must be preserved. For example, the French translation of `{0}_selection_{1}` is `{0}_sélection_{1}`. Here `{0}` refers to the layer name and `{1}` refers to a text value.

Another example:

```
Cannot add a LayerCollection {0}.
```

becomes in French

```
Impossible d'ajouter la collection de couche {0}.
```

2- & symbol

The `&` symbol is used to create keyboard shortcuts. So they have to be preserved (if possible).

Example:

```
&Copy
```

becomes in French

```
&Copier
```

In this example, the shortcut will be placed on the “C”.

3.2.3 Contributors

Contributions are welcome! You can help out by translating, proofreading existing translations, or creating your own translation team (for a new language).

If your translation looks good, your work will be integrated into OrbisGIS.

Already done:

- English,
- French.

Languages in progress:

- Italian: Thanks to Simone Sandri (aka xseris), Team manager
- Portuguese (Brazil): Thanks to Enrico Nicoletto, Team manager,
- Arabic: Thanks to Ilyes Smadi (aka Smadi), Team manager,
- Chinese: Thanks to Quing-Xia Zhou, Team manager, Yinghao Li and Su Qi
- Spanish: Thanks to Emilio Plaza García (aka Dharth),
- Persian: Thanks to Mohammad Hosseini (aka PersianPolaris), Team manager and Mehrazin Omrani (aka Azin),
- Russian: Thanks to Assem, Team manager,
- Greek: Thanks to Ioannis Tsimpiris (aka gtsimp), Team manager,
- Breton

3.2.4 Transifex configuration (for developers)

The python file in tools will call `mvn gettext:gettext` in order to update all .pot files in the project tree.

To send to transifex new translation keys:

```
cd tools/  
python internat.py  
tx push -s
```

To retrieve updated translations from transifex.

```
tx pull -a --minimum-perc=1
```

This will update all .po files. A commit then a pull request has to be done in order to be published into the final binary.