

User Manual

# **MarushkaDesign**

Configuration and Publication of the  
Project



**GEOVAP**

# CONTENTS

<b>1</b>	<b>THE BASIC CONCEPT .....</b>	<b>1</b>
<b>2</b>	<b>VERSIONS OF MARUSHKADESIGN AND MARUSHKA SERVER 3.0.0.....</b>	<b>2</b>
2.1	THE CHANGE IN CONNECTION PARAMETERS OF DATABASE DATA STORES .....	2
2.2	UTILIZATION OF "IN MEMORY" DATABASE REDIS FOR STORING TEMPORARY FILES .....	2
<b>3</b>	<b>INSTALLATION AND PRODUCT LICENSING .....</b>	<b>3</b>
3.1	INSTALLATION .....	3
3.2	LICENSE .....	4
3.2.1	<i>A Trial License .....</i>	<i>4</i>
3.2.2	<i>Full License .....</i>	<i>5</i>
3.2.3	<i>How to Get the License Code .....</i>	<i>6</i>
3.2.4	<i>The Main License and Module Licensing .....</i>	<i>6</i>
3.2.5	<i>The License Manager .....</i>	<i>7</i>
3.2.6	<i>License of MarushkaServer .....</i>	<i>8</i>
3.2.7	<i>Setting the Web.config of MarushkaServer .....</i>	<i>8</i>
3.2.8	<i>Export of License Files .....</i>	<i>9</i>
3.2.9	<i>Licensing of Off-line Server .....</i>	<i>9</i>
<b>4</b>	<b>HOW TO DO IT? CONFIGURING PROJECT IN MARUSHKA DESIGN .....</b>	<b>11</b>
4.1	MENU OF MARUSHKADESIGN.....	11
4.1.1	<i>File .....</i>	<i>11</i>
4.1.2	<i>Edit .....</i>	<i>11</i>
4.1.3	<i>Draw.....</i>	<i>11</i>
4.1.4	<i>Modify .....</i>	<i>12</i>
4.1.5	<i>Tools .....</i>	<i>12</i>
4.1.6	<i>Settings .....</i>	<i>17</i>
4.1.7	<i>Modules .....</i>	<i>17</i>
4.1.8	<i>About.....</i>	<i>17</i>
4.2	TOOLS OVERVIEW IN MARUSHKADESIGN.....	18
4.3	MANIPULATION WITH DIALOG BOXES (WINDOWS) .....	19
4.3.1	<i>Dialog Box Object Properties .....</i>	<i>19</i>
4.3.2	<i>Map Surface .....</i>	<i>20</i>
<b>5</b>	<b>BASIS OBJECTS IN MARUSHKADESIGN ENVIRONMENT .....</b>	<b>22</b>
5.1	BASIC TERMINOLOGY .....	22
5.2	THE PROJECT .....	23
5.3	DATASTORE – DATE SOURCE.....	23
5.3.1	<i>Cartographic Coordinate Systems.....</i>	<i>24</i>
5.3.2	<i>Form Layer – The Requirement for Spatial Data .....</i>	<i>25</i>
5.3.3	<i>Symbology .....</i>	<i>27</i>
5.3.4	<i>Physical Layer – Physical Data Storage.....</i>	<i>28</i>
5.3.5	<i>Publish Layer .....</i>	<i>29</i>
5.3.6	<i>Annotation Layer.....</i>	<i>31</i>
<b>6</b>	<b>FILE DATA SOURCES PROPERTIES.....</b>	<b>32</b>
6.1	WKB.....	32
6.2	SHP.....	32
6.3	DGN .....	32
6.4	RASTER FILES .....	32
6.5	WMS .....	32
6.6	REMOTE DATABASES .....	32

<b>7</b>	<b>PROPERTIES AND POSSIBILITIES OF DATABASE DATA STORES .....</b>	<b>33</b>
7.1	SEQUENCES AND SYSTEM OBJECTS FOR OGC WKB FORMAT .....	33
7.2	DYNAMIC COLUMNS AND COLUMNS SET_PARS_.....	33
7.3	DYNAMIC PARAMETERS OF SET_ENV_ ENVIRONMENT .....	35
7.4	QUERY LIBRARY.....	36
7.4.1	<i>Interconnection with Database Table WWWQUERY.....</i>	37
7.4.2	<i>Information Query on Graphic Element.....</i>	37
7.4.3	<i>Common Info .....</i>	38
7.4.4	<i>Query Type Update .....</i>	39
7.4.5	<i>Document Query.....</i>	39
7.4.6	<i>Auxiliary Query for Selection of Document from DB .....</i>	40
7.4.7	<i>Localization Query.....</i>	40
7.4.8	<i>Localize Group.....</i>	41
7.4.9	<i>A List of Offered Values (Code List).....</i>	41
7.4.10	<i>A List of Static Values.....</i>	41
7.4.11	<i>Redirection.....</i>	42
7.4.12	<i>Draw by Etalon .....</i>	43
7.4.13	<i>Delete Element .....</i>	43
7.4.14	<i>File Browser.....</i>	43
7.4.15	<i>Draw and Locate WKT String .....</i>	43
7.4.16	<i>Multi Select .....</i>	43
7.4.17	<i>Localize Target Coordinates.....</i>	44
7.4.18	<i>External Queries.....</i>	45
7.5	CELL LIBRARY.....	45
7.5.1	<i>Legend.....</i>	46
7.6	USER STYLE LIBRARY .....	48
7.6.1	<i>Convention in Takeover Symbolology.....</i>	49
7.6.2	<i>Styles sy10 and sy17.....</i>	49
7.6.3	<i>Styles sy11 and sy12.....</i>	49
7.6.4	<i>Styles sy21, sy22 and sy23.....</i>	50
7.6.5	<i>Style sy24 .....</i>	51
7.6.6	<i>Style sy13 .....</i>	51
7.6.7	<i>Styles sy14 and sy15.....</i>	52
7.6.8	<i>Style sy16 .....</i>	52
7.6.9	<i>Style 10ws .....</i>	53
7.6.10	<i>Style 11ws .....</i>	54
7.7	ETALON LIBRARY .....	55
<b>8</b>	<b>PUBLISH SERVER MARUSHKA®.....</b>	<b>57</b>
8.1	REQUIREMENTS .....	58
8.2	APPLICATION POOL.....	58
8.3	CREATING AN APPLICATION ON IIS 7.....	58
8.4	SETTING UP APPLICATION DIRECTORIES OF MARUSHKA.....	59
8.5	COPYING THE CONFIGURATION XML.....	59
8.6	EDITING THE CONFIGURATION FILE WEB.CONFIG.....	59
8.6.1	<i>Authorization Settings.....</i>	60
8.6.2	<i>Additional Settings.....</i>	63
8.7	EXTERNAL CALL OF MAP CLIENT MARUSHKA.....	63
	<b>ANNEX – UML DIAGRAM OF OBJECTS GSFRAMEWORK .....</b>	<b>70</b>



# 1 The Basic Concept

The basic concept of GSFrameWork technology is using uniform core, which provides approach and unified graphical presentation of geographical data. The core is being developed on component object principle; there is no limitation in terms of possible integration into other applications (.NET) that require processing / access to geographically referenced data.

On the unified core are based applications of following type:

- The strong client **Marushka Design** is a visual administration tool for creating configuration project for server Marushka. Project is a collection of different data stores, queries, symbols, and their settings. MarushkaDesign offers a convenient and intuitive configuration of all these items of the project and also offers a full set of features for transformation, editing or creation of geographically relate data.
- The application server **GSFWWS (GS FrameWork Web Services)** – remote connection for a strong client / publish server
- Publish server **Marushka®** is a powerful application server. In combination with HTML client (web browser) is used to provide geographic data on the Internet (HTML, OGC WMS). Publication server also provides data via OGC Web Services standards, specifically WMS, WFS or Web Services in platform .NET (SOAP) environment. The advantages of HTML client include its runtime environment, which is a standard Web browser such as Internet Explorer, FireFox and other. So HTML client does not need any other installation to run.

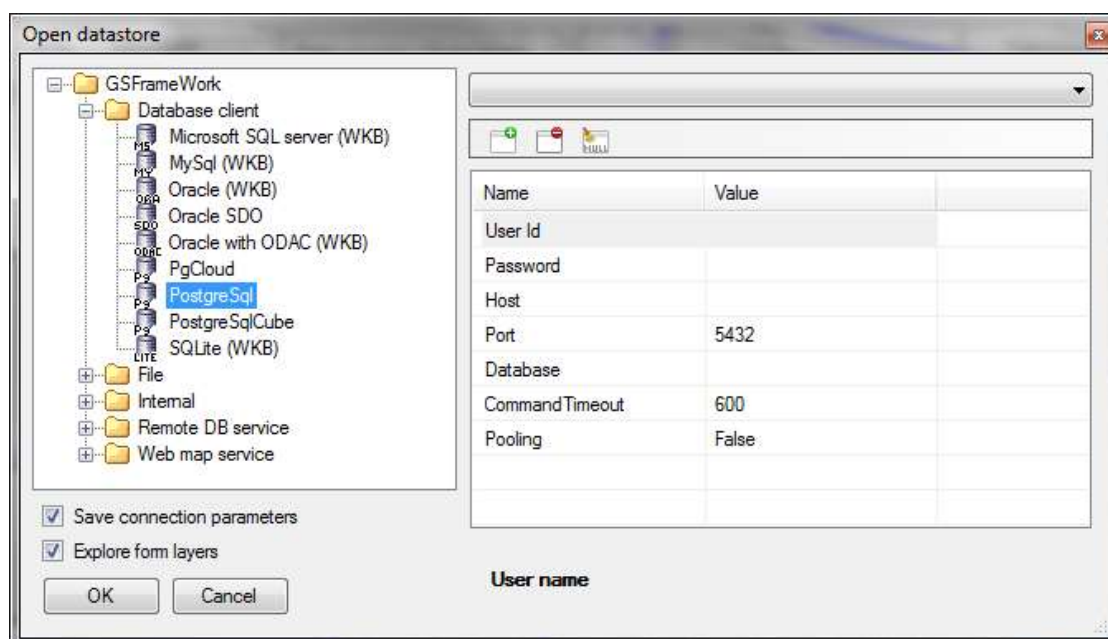
## 2 Versions of MarushkaDesign and Marushka Server 4.0.0

With the transition to the version of software products MarushkaDesign and Marushka server comes changes that can be divided into the following groups:

- License policy of GEOVAP Company (chapter [Installation and product licensing](#))
- Automatic versioning of software including packages, online updates (chapter [Installation and product licensing](#))
- Change in the connection parameters of database data stores
- The use of "in-memory" REDIS database for storing temporary files

### 2.1 The Change in Connection Parameters of Database Data Stores

With the expanding repertoire of database resources we were forced to generalize generating of their connection strings. There is no uniform standard for all databases. Each database store provides different options. Login dialog to the database thus newly look like this:



In the column name is the key in the login string or it is filled with its normal value. You can also add additional keys using the "Add a foreign key to connection string". Caution, from the database data stores were removed properties that are related to the connection strings (e.g. ConnectionPooling). Please, fix them in your projects to correct values. A summary of options, keys and connection values for different databases can be found at e.g.: <http://www.connectionstrings.com/>

### 2.2 Utilization of "in Memory" Database REDIS for Storing Temporary Files

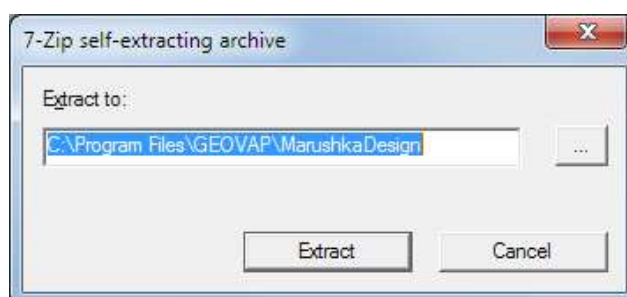
It was motivated both by increasing the performance of web applications (saving disk operation memory), and allowing the use of application servers in a "farm" – stores in it even necessary context of web session. In case, that you have available REDIS server (free for WIN and LINUX), then you can immediately take advantage of a simple change in the `web.config` file (search "REDIS").

## 3 Installation and Product Licensing

### 3.1 Installation

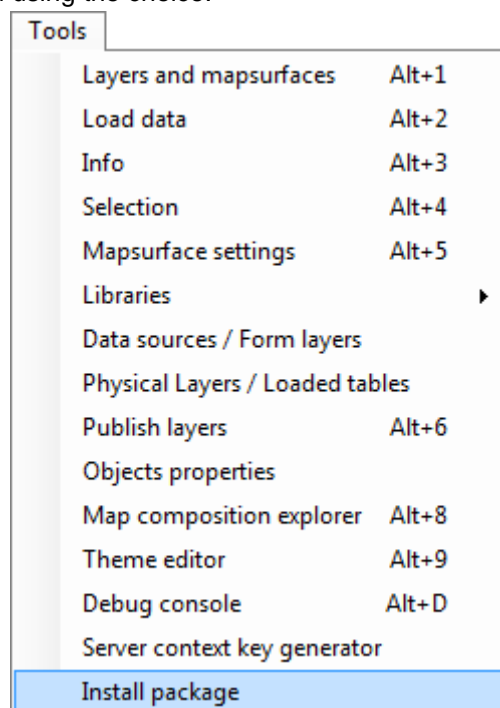
The first installation is performed so that you download installation package from URL <http://install.geovap.cz/GsInstall/GetLatestHandler.ashx?package=marushkadesign&sfx=yes> containing the latest version of the software Marushka. It is a self-extracting zip, run it and enter the root directory, typically:

C:\Program Files\GEOVAP\MarushkaDesign

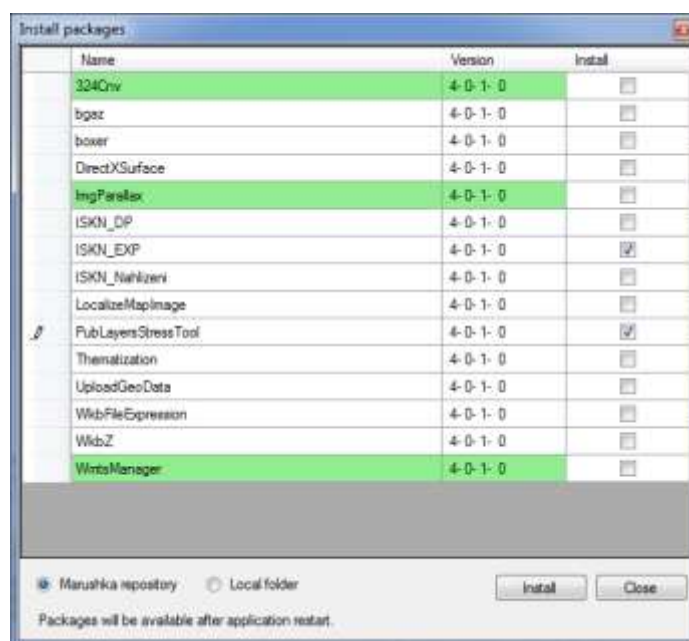


The next installation will be offered when the new version is available and the installation package is downloaded to a local directory. Installation packages are electronically signed by GEOVAP Company, and it is therefore precluded, that the application launches any other executable code. By this is online updating sufficiently protected against possible attacks. Downloading new version happens asynchronously in another process, the user during his work do not even notice that a new version is being downloaded.

Packages are installed using the choice:



In the menu of available packages choose the required one and click on "Install package".



Packages are installed to the folder of the version of the appropriate application. If the package is working with the web service, then the web components are also installed into the directory of the version of application in the structure, which a web service supports (usually a subdirectory of `service`). Modules included in the package are available after restart of the application.

## 3.2 License

MarushkaDesign can be licensed in two different modes. In mode full license "FULL" or in trial version "TRIAL". The trial mode verifies license information on each startup by the licensing service. MarushkaDesign thus cannot run in off-line mode in trial version.

A valid license is defined by a file MarushkaDesign.lic; this file is located in the same directory as the executable file MarushkaDesign.exe.

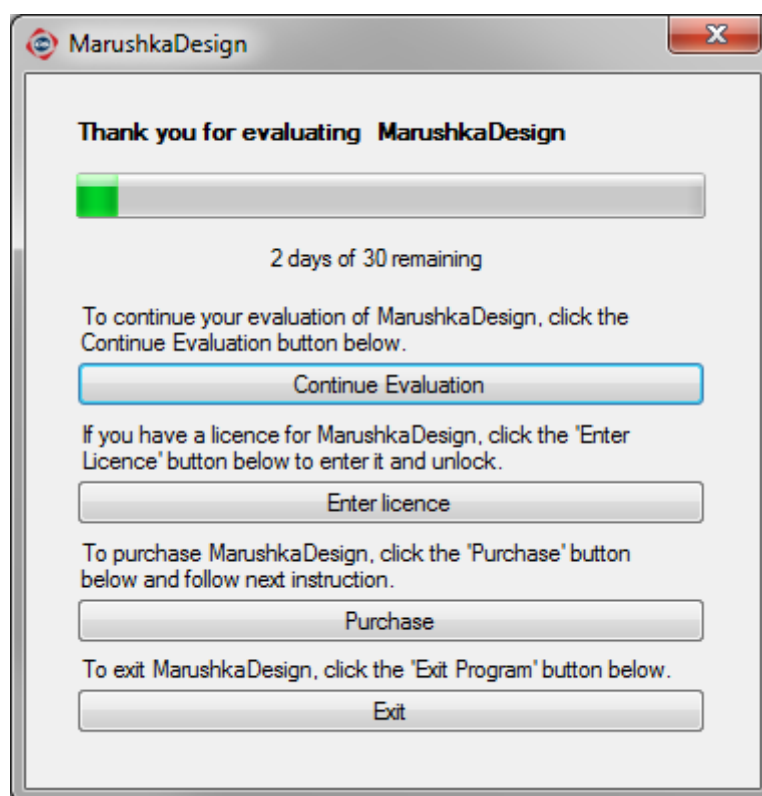
### 3.2.1 A Trial License

This license is always available and enables full use of the product MarushkaDesign for 30 days.

After the first start it automatically creates license file and the user is informed by the license dialogue about the elapsed time. Then it is necessary to choose one of the available options: continue in trial version – if possible, insert your license code, in a web browser open the order form or quit the application. MarushkaDesign in this mode always verifies the license by the license service by GEOVAP Company, so you must be online and arrange the service availability

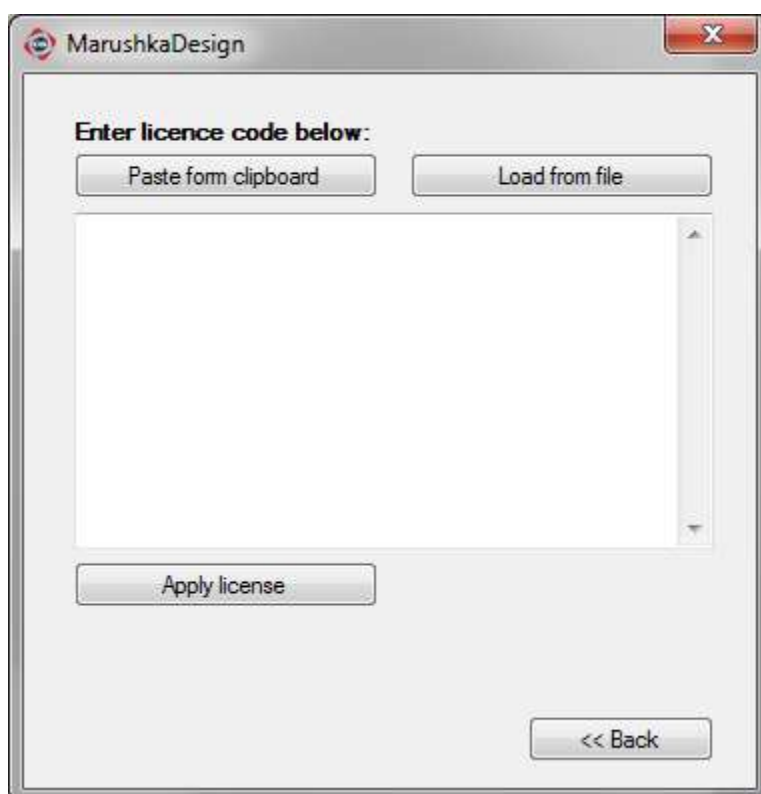
<http://install.geovap.cz/licenseservice/LicenceService.asmx>.



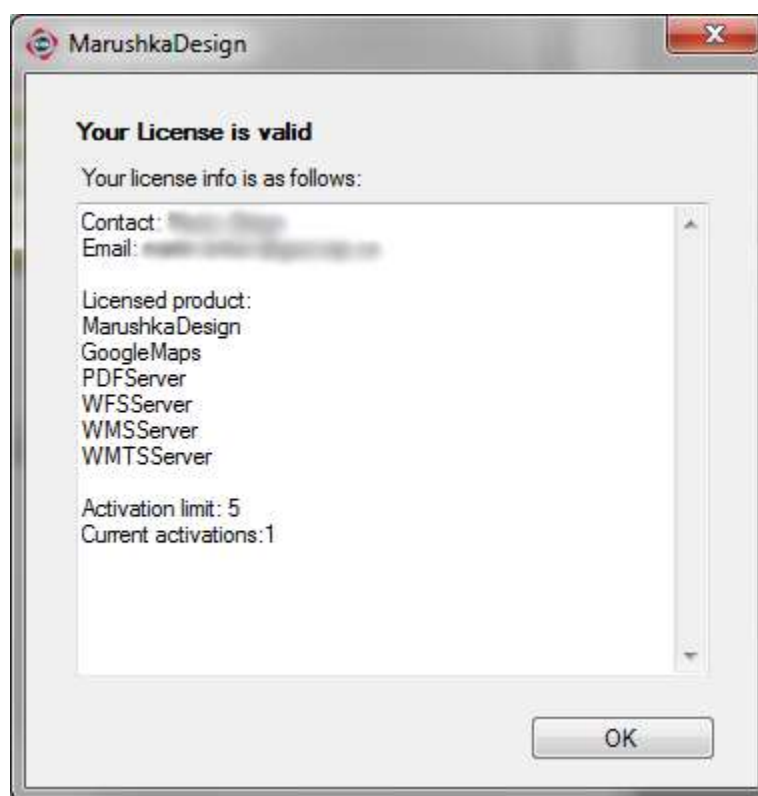


### 3.2.2 Full License

The full license is created by inserting the license code to the license dialog. The license code is verified through licensing service of GEOVAP Company (<http://install.geovap.cz/licenseservice/LicenceService.aspx>), and in this case it is also necessary to be online and provide access to the licensing service. The following chapter also describes how to license server, which is e.g. behind firewall and it is not possible to ensure the possibility of licensing services.



After entering a valid license code, a new license file will generate and display the detailed information about the license. The detailed information includes: name of the license owner, products that the license include, activation limit and current activations. After the successful activation is product ready to be used also in off-line mode.



### 3.2.3 How to Get the License Code

If you want to obtain the license code, you have to visit the web page <http://marushka.geostore.cz/en/install/>.

Then you have to register and order the product. The GEOVAP Company will contact you as soon as possible and will send you a price offer. After confirming the offer, you will receive a valid license code. This code should be safely stored to prevent fraudulent use. The button "Buy" also refers to the same address.

### 3.2.4 The Main License and Module Licensing

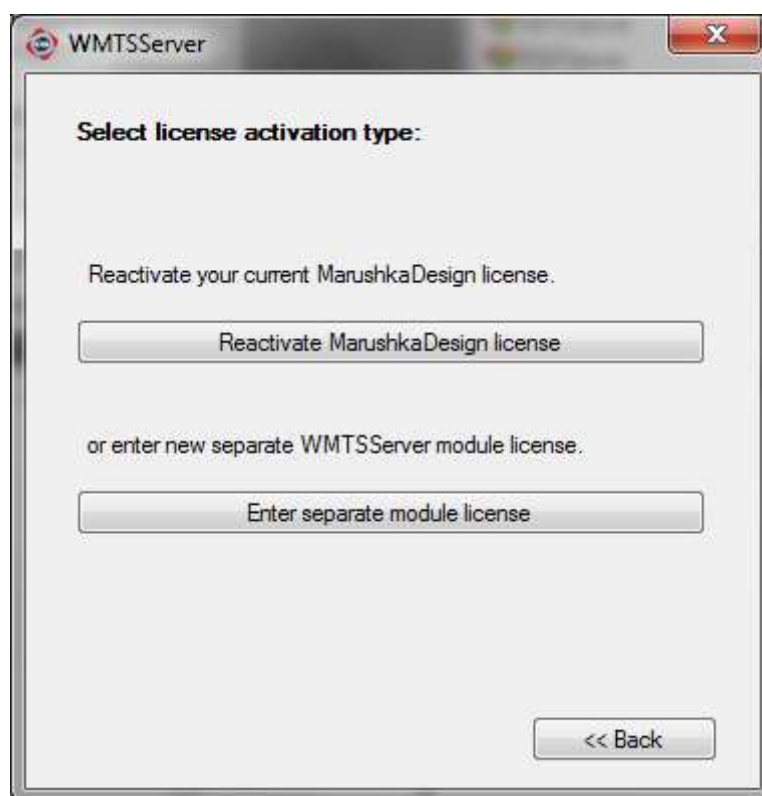
Marushka technology is based on component architecture. It can thus freely add modules. These modules are loaded after their first startup and can be also licensed separately. Single modules may or may not include a trial license.

The main license may include a list of licensed modules. If the main license of MarushkaDesign contain in the list a licensed module "A", then the module can be used without limitation. If the module "A" isn't included in the list of licensed modules and module "A" contains a trial license, so during the first use of module "A" a license dialog will appear. Then using the license service will be created a license file A.lic and the module can be used for 30 days. In this case is also required to be online, because during the first start of module "A" is the trial license authenticated against the licensing service. If the licensing service is not available, module "A" will end its run and won't be available, as well as after the trial period expires.

Module "A" can be then licensed in two ways:

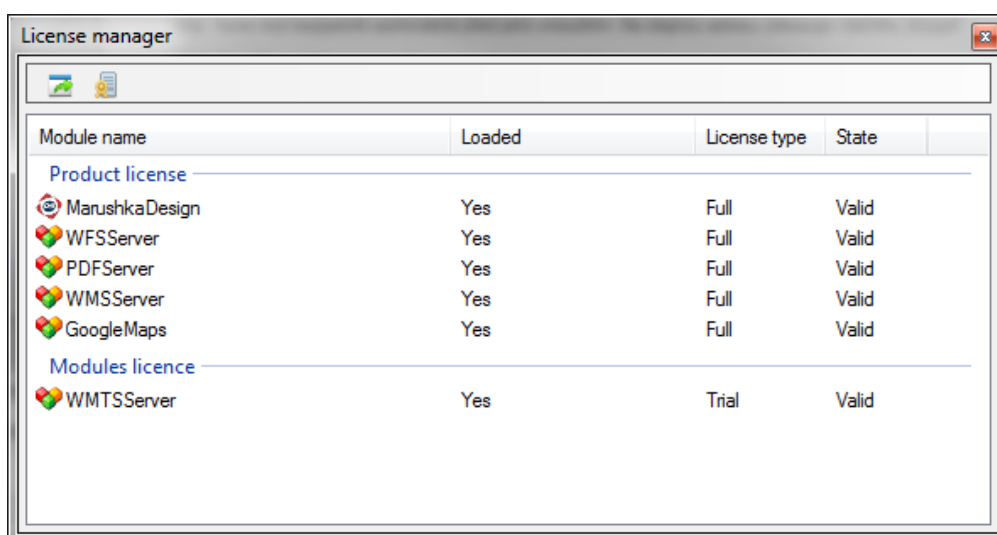
- a) By reactivation of the main license. In case, that module was added to the main license. Maximum number of the activations of the module is the same as maximum number of activations of the main license. Module "A" will then appear in the list of licensed modules of the main license.

- b) Inserting a separate license of the module "A". This will update the license file A.lic and after this action, the module "A" can be used off-line.

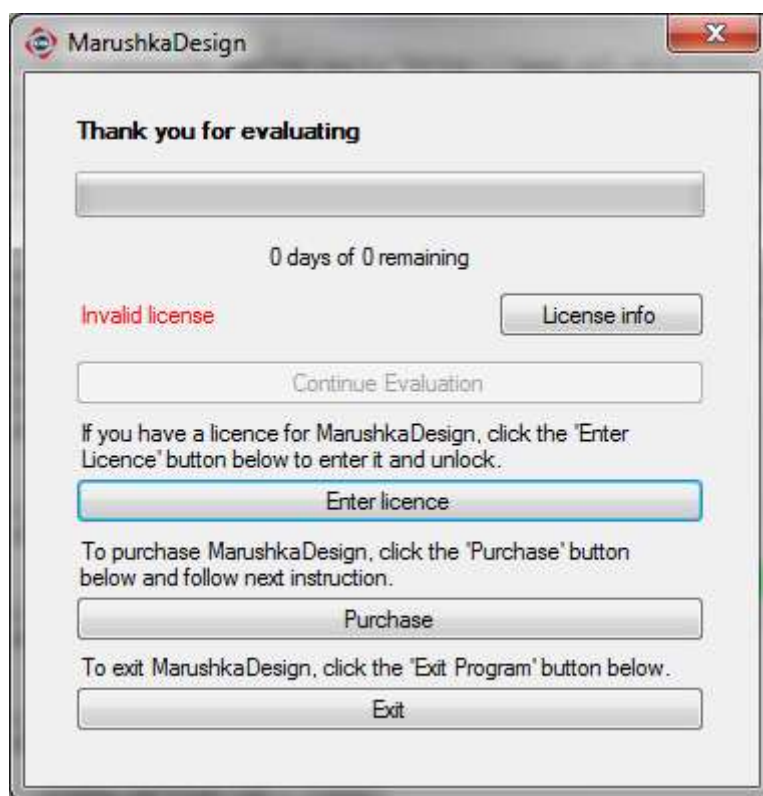


### 3.2.5 The License Manager

MarushkaDesign application includes a component for license management. Run it from the main menu *About / License manager*. "License Manager" displays a list of all modules available in the licensed product assembly of MarushkaDesign and displays their current status – valid / invalid, license type – trial / full and if the module is loaded or not (modules are loaded after their first use). The list is divided into two groups *Product license* and *Module license*.



Double click activates a selected item. In case that license is loaded and activated, the license details display. If not, then the license loads and activates and is subsequently validated through licensing service. If the license is trial or invalid, the component will offer possibility to insert license code or reactivate the main license.



Button "information" will display more detailed information about error of the license:



### 3.2.6 License of MarushkaServer

*MarushkaServer* and *MarushkaDesign* share the same license. So on one machine it is possible to operate with both *MarushkaServer* and *MarushkaDesign* from one license. *MarushkaServer* cannot perform activation of the full or trial license, so it is necessary to perform this activation using *MarushkaDesign*. In case that server is running under the trial license, you must arrange so that ASP.Net application has access to the address of licensing services, server therefore cannot be operated off-line.

### 3.2.7 Setting the Web.config of MarushkaServer

For the operation of the server must be defined in the configuration files of web.config path to the license files using key **LicenseDir**

```
<!-- ***** License settings ***** -->
<!-- Path to license files dir -->
<add key="LicenseDir" value="c:\MarushkaLicense\"/>
<!-- ***** License settings end ***** -->
```

Then arrange rights to read in this directory to ASP.net application. It is appropriate to place license files outside the root (*wwwroot*) of web directory.

In the default installation of MarushkaDesign is server operated on an integrated webserver MarushkaDesign. **LicenseDir** key is then defined in the installation directory MarushkaDesign.

### 3.2.8 Export of License Files

To export the license file into the selected directory is possible to use "License Manager" in the menu of the component by "Export all licenses to directory" button. This feature will copy all the "\*.lic" files into selected directory.

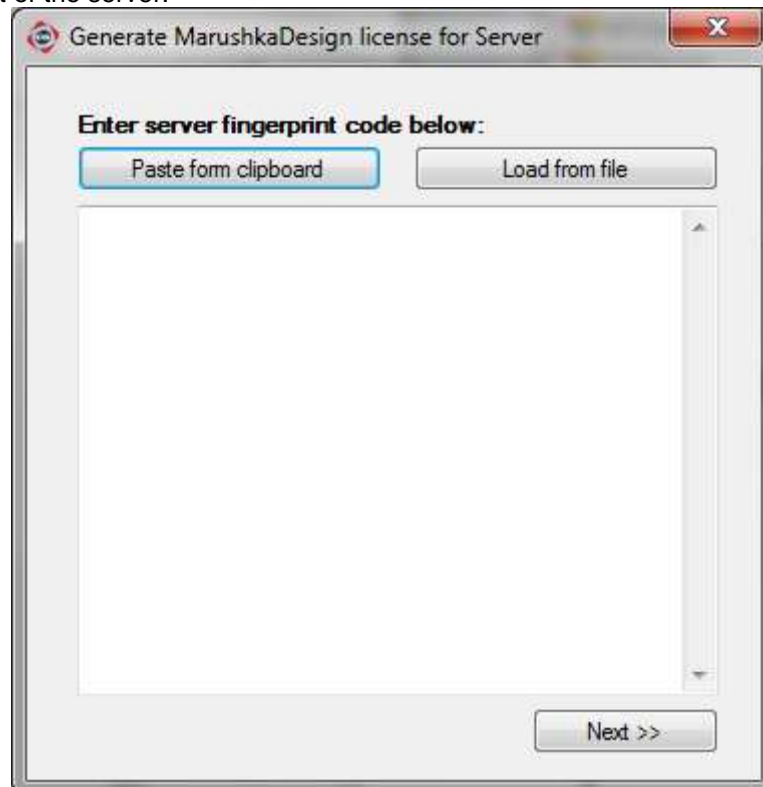
### 3.2.9 Licensing of Off-line Server

It is a server that is e.g. behind a firewall and does not have access to the license service.

#### 1) Fingerprint of the server

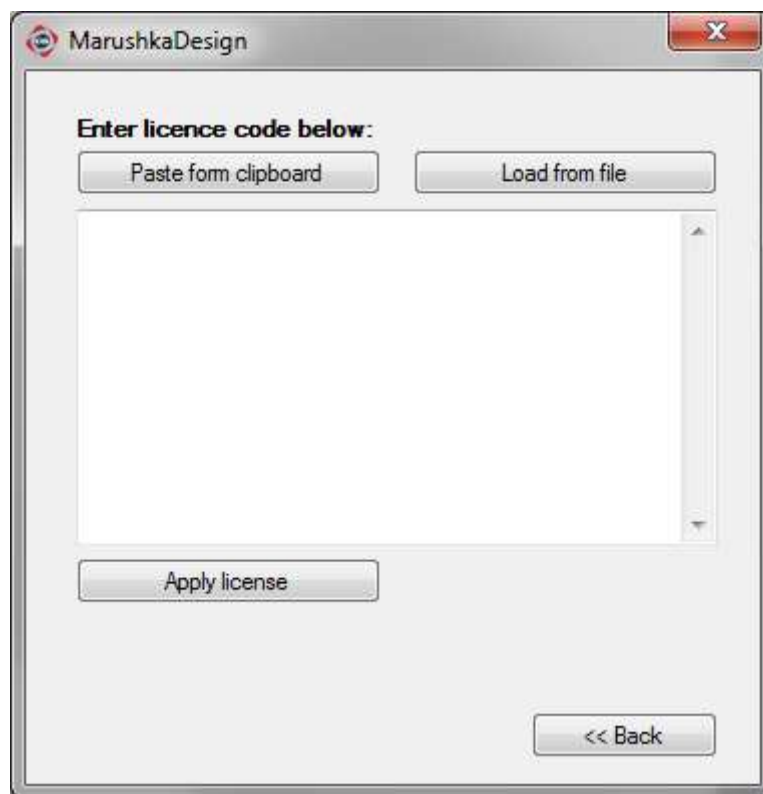
In the case you need to license a server, which is for some reason not available to license service (cannot activate your license by MarushkaDesign directly on the server), it is necessary to generate a *FingerPrint* of the server to be licensed, using application **FingerPrint.exe**. This application is included in the installation of MarushkaDesign and is located in the subdirectory "*FingerPrint*" in the installation directory of MarushkaDesign.

Copy this application on the server, for which you want to create a license, and generate a fingerprint of the server.



#### 2) Generating of licenses

If you have a fingerprint of the server, it is possible on the machine with available licensing service to generate valid licenses using the "*License manager*" and function „*Generate license for a server*“. First load the fingerprint of the server, then insert the license code and copy the created license files on the target server. In the **web.config** ASP.Net application adjust the key **LicenseDir**.



3) MarushkaDesign

If you need to run MarushkaDesign on the same machine at a same time, copy the licenses from the previous step into the MarushkaDesign installation directory on the target server.

4) Each time you change a license it is required to export licenses again.

## 4 How to Do It? Configuring Project in Marushka Design

### 4.1 Menu of MarushkaDesign

One of the basic components of the MarushkaDesign application environment is the menu, which consists of items: File, Edit, Draw, Modify, Tools, Settings, About, Modules and Main Menu Settings.

File Edit Draw Modify Tools Settings About Modules Main Menu Settings

#### 4.1.1 File

New	Ctrl+N
Open Project	Ctrl+O
Save Project	Ctrl+S
Save Project As	Ctrl+Shift+S
Exit	Ctrl+Q

Menu item **File** offers essential tools for working with the whole project. The main basis of each project is the xml format file. This file stores all the information related to each single project or to one of the themes of the project. Individual functions of the File menu are standard and quite intuitive.

#### 4.1.2 Edit

Undo	Ctrl+Z
Redo	Ctrl+Y
Delete elements (Loaded)	
Select all	Ctrl+A

**Edit** menu item include basic editing tools for working in the map window. *Undo* function returns the last executed step in the map window. The function *Redo* executes again the last returned step. *Select All* function selects all the elements in the active map window. Delete elements (Loaded) function allow bulk deletion of the elements, that were prior placed to the selection set, from the physical layer.

#### 4.1.3 Draw

Line	▶
Point	▶
Arc	▶
Curve	▶
Area	▶
Point 3D	

A set of functions of the item menu **Draw** include different options for drawing simple or more complex objects into the map window. For placing any object into the map window must be active the physical layer, into which is the new element placed. Activation of the physical layer is performed by double click of the left mouse button on the physical layer name in the physical layer dialog box. In the case that there is no physical layer, it can be created by the right mouse click in physical layer dialog box. The active layer is then highlighted

The first item of this menu is draw **Line**, containing a total 5 actions, *Line String*, *Line String by Points*, *Perp from point*, *Perp to point*, *Line String 3D*.

The second menu item is **Point**, allowing drawing point elements – *Text and Cell*.

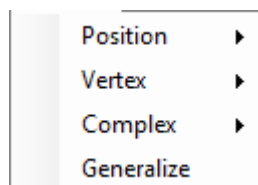
The third menu item is **Arc**, including actions *Arc by 3 points* and *Arc by center and 2 points*.

Item **Curve** contain just one function *B-spline curve*.

Next item of menu is draw **Area**, allowing to draw geometrical shapes: *Box, Rectangle, Polygon, Circle, Ellipse*

The last item of the menu is **Point 3D**, allowing to draw a point in 3D.

#### 4.1.4 Modify



Menu item **Modify** include set of tools for the modification of individual elements or the entire image displayed in the active graphical window of the application.

- **Position** - this set of three functions allows performing operations: *Move to, Copy to, Rotate* with individual elements in the map window. These map operations are executed just in the map window, they do not modify the data in the database. If the user needs to save the modified data, he has to save the data in the database.

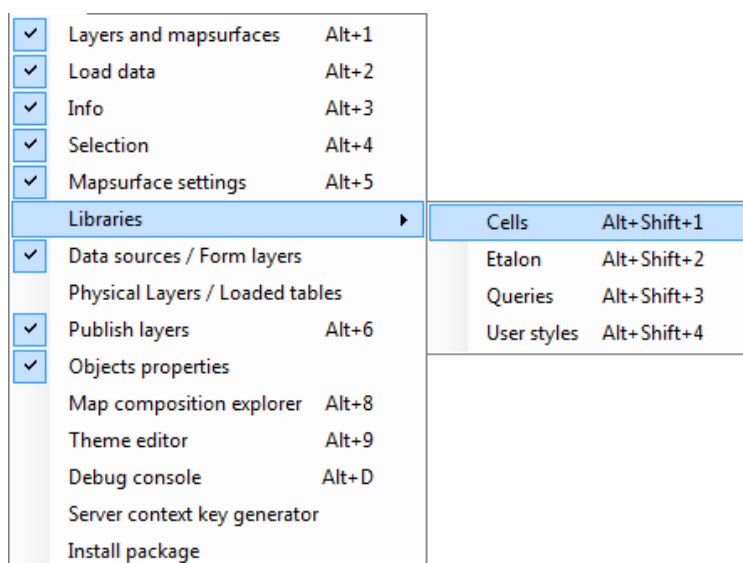
- **Vertex** – this set of features allows operations with individual vertexes – *Move, Insert, Extend*. It also allows change of diameter of selected circle or ellipse, using operation *Modify radius*.

- **Complex** – allows more complex functions, such as *Chain to line string, Chain to ring, Group rings, Decompose collection, Create collection, Create org collection, Create buffer zone, Union polygons, Intersect of two areas, Difference of two areas*.

- **Generalize** - allows generalizing elements in the selection of the certain scale.

#### 4.1.5 Tools







Using menu Tools (and also using the appropriate shortcuts) can be switched on and off toolbars and libraries, perform data selections and adjust displaying of graphical windows.



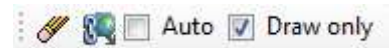




## Layers and mapsurfaces



-  - Opens window Properties of single elements
-  - Opens window Data sources / Form layers
-  - Opens window Publish layers
-  - Opens window Physical Layers / Loaded tables
-  - Opens New mapSurface
-  - Runs local web server to review a project (edited project will save automatically during launch)





## Load data



-  - Clears all elements in the map window
-  - Clears all the data and loads them again by the strategy (checked publish layers)
- ☐ Auto - Automatic load on scale change
- ☒ Draw only - Only the drawing is loaded into map window










## Info



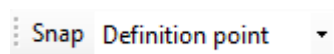
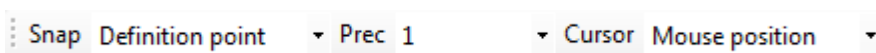
-  - Gets information about element geometry
-  - Gets information about element queries (like GetFeatureInfo service)
-  - Displays information icons
-  - Measurement of length, area and element

## Selection



-  - Active symbology
-  - Symbology of selection
-  - Selection by rectangle
-  - Selection by polygon
-  - Selection by line
-  - Selection wizard
-  - Invert selection
-  - Cancel selection
-  - Number of currently selected elements

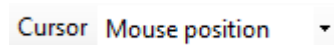
## MapSurface settings



- Snap point setting



- Number of decimal points of coordinates (displayed on bottom toolbar)



- Cursor setting

## Map composition explorer

Map composition explorer is a powerful tool for management of order and map scale ranges of single publish layers. On the left side of the map window is the dialog box with a list of Publish layers in the standard structure of MarushkaDesign environment.

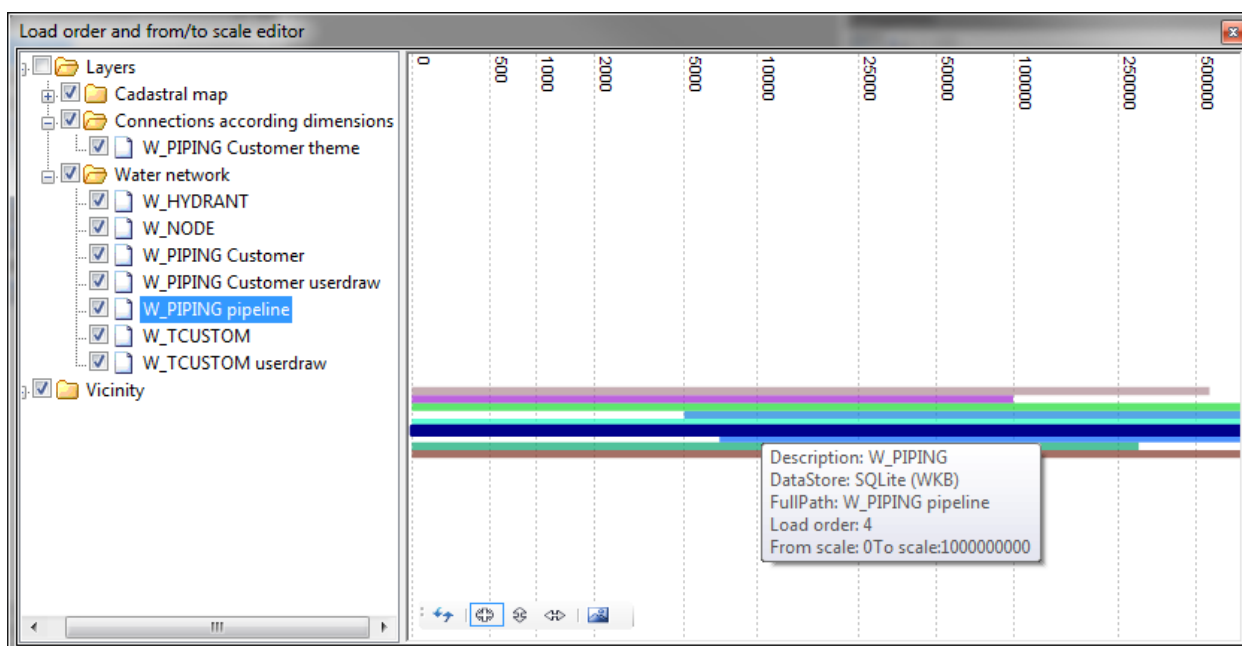
Context menu allows multiple selection of the child records of the specific publish layers, or deployment of the node. On the right side of the dialog box, you can see the color spectrum of differentiated individual layers in the order of publication and map scale ranges, in which they are displayed. At the moment when the mouse cursor moves on the line on the right side, the floating help with description of the publish layer will display, as shown in the picture.

Clicking on the line performs selection of the particular publish layer. The selected publish layer will highlight in both left and right sides of the dialog box. Selection of the publish layers can be realized also in the left side either individually or as a group of layers – either by using context menu (right mouse button) or multiselect using CTRL, resp. SHIFT and the left mouse button. With such a selected publish layer (or group of layers) you can further work in the right side of the dialog box.

You can move and zoom in / out in the right side of the dialog box using the mouse wheel. If you hold the mouse wheel, you can move in the cutout of the window. While holding and rotating the mouse wheel you modify the map composition cutout and perform zoom in or zoom out. In case, when the publish layer or the group of layers is selected in the right part of the dialog box you can change order of rendering and scale range.

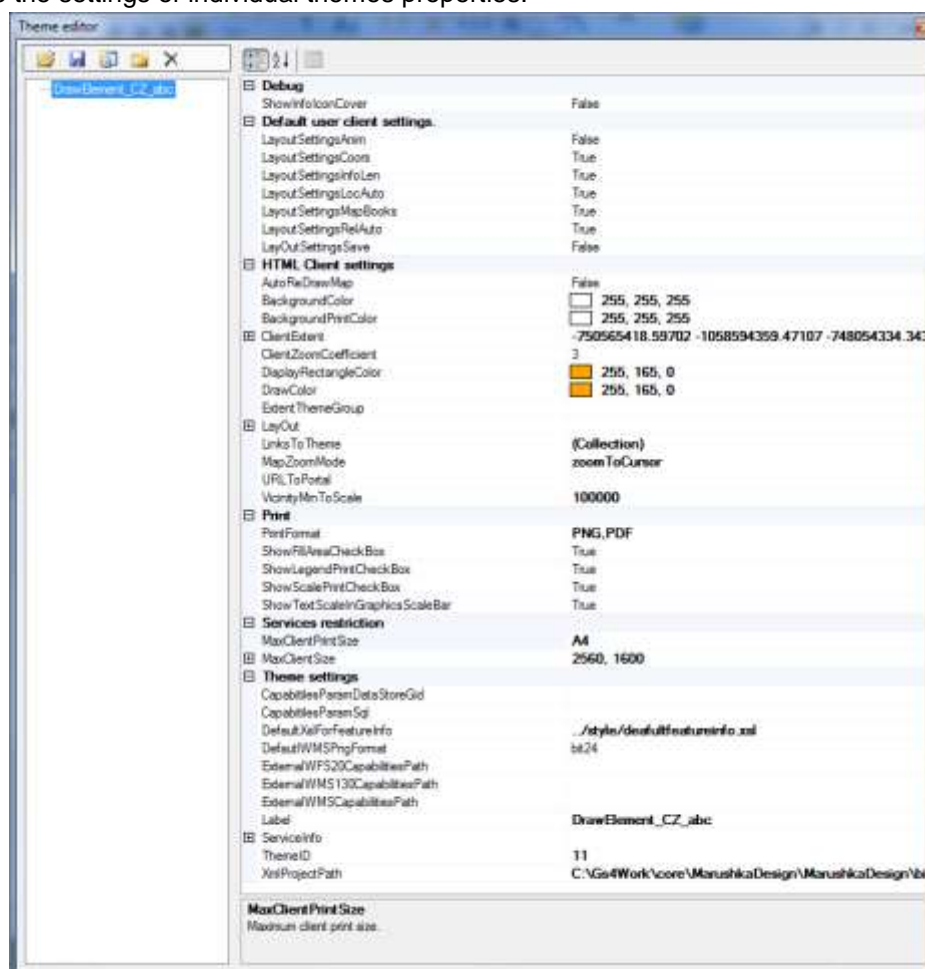
Switching between functions (change of order and scale range) is possible by either right mouse click or using the icons at the left bottom corner of the window. While changing order of layers, will be during vertical layer movement always displayed two publish layers, between which will be placed the newly created layer. When changing the scale (in the first step is set an initial scale) will be the new scale graphically highlighted, which is then confirmed by the left mouse button. In case when you add any other






publish layer, rename, modify or remove the publish layer, then it is necessary to display changes perform refresh by clicking the button in the lower left panel.



## Theme Editor

It is used for preparation of the web publication of the project / projects. In the following dialog box you can see the settings of individual themes properties.



On the left side of the dialog box is a set of icons for the configuration of file management control issues. From the left are the icons open , save  and new theme project , add new theme item  and delete selected item  Under this set of icons is a list of themes.

An important feature in this part of the window is a pop-up menu for each theme, which is activated by the right mouse button over the selected theme. In the context menu is an important feature *Set client extent from window*. After activating this function, select the limiting rectangle that will become the default setting of the HTML client (the value is copied to the row ClientExtent in the right side of the dialog box, so there is no need to enter coordinates manually). The other two pairs of functions are used to copy and the subsequent insert of two rather broad categories. Theme settings of HTML client - *LayOut settings* and settings for layout info - *Serviceinfo*.

The entire side of the dialog box is reserved for a setup of the theme items of the particular theme. Individual settings are divided into diverse categories; each item has at the bottom of the dialog box help, specifying further properties of the particular function.

*Default user setting* – a category of user settings that can be changed also in the web presentation of the given topic in Menu – Settings. It is therefore only the initialization setting that is also user-modifiable. In this setting category is possible to switch on / off **settings**: map animation, coordinates animation, number of info elements, automatic localization, bookmark overlap map (if you disable this function the window of map publication is all visible and the dynamic legend or other functions depending on the displayed cutout are not distorted), reload map automatically on window resize and get last map position on the load (uses cookies).

*HTML Client settings* – initialization settings that are modifiable only by the project manager in the configuration xml file. The settings overview - automatically redraw the map while changing the displayed layers, background color, starting limiting rectangle in the target units of the target cartographic projection, zoom factor (the number of times it zooms in / out when zooming the map), layout (category allowing on / off of different user control elements, auxiliary components, overview display, headers, language mutations, etc.), links to available themes, URL of the main portal in menu and minimal scale of dynamic overview.

For parameter *LayOutScalebarMinMaxValue* can be set the minimum and maximum values of scales of HTML client. This is activated by setting the parameter *AutoFromLayers* to 'True', otherwise the minimum and maximum values of HTML client taken from scales of publish layers of the project. Therefore, if the value of this parameter is set to 'False', it is possible to set minimum and maximum values manually. If in the theme editor is set integration of Google layers, the minimum and maximum scales, as well as individual zoom layers are derived from underlying Google map layer, which have individual zoom levels firmly defined.

In the chapter [8.7 External call of the map client Marushka](#) are listed parameters of call, which are complemented by their variables in the theme. There are descriptions of the individual parameters of individual parameters and their allowed values. These values can be set either in the theme, or they can be called via the HTML interface. For more information, see chapter 8.7.

For parameter *LinksToTheme* can be set a list of available themes for the current theme, for parameter *URLToPortal* can be set URL to the main portal in menu. Using parameter *VicinityMinToScale* allows setting the minimum scale of dynamic overview map. If the value is lower than the minimum value *ToScale* of all the form layers labeled as overview layers, the minimal value of this layer will be used.

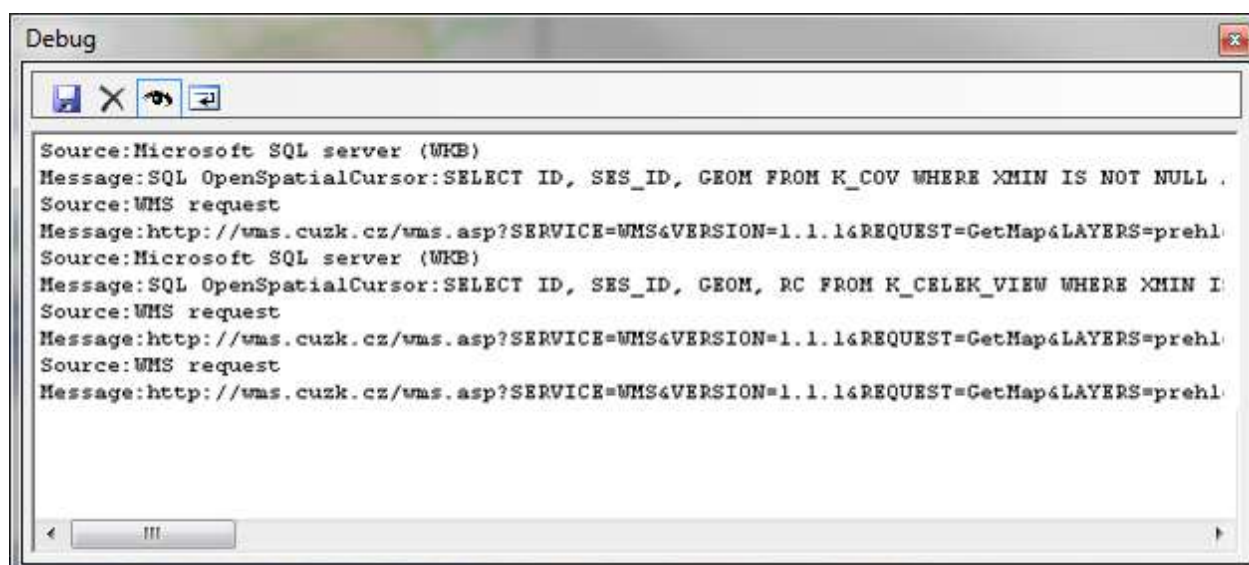
*Theme settings* – category of other settings of the theme. The default type of PNG images provided by WMS service, paths to the external capabilities of WMS and WFS files, the label of the theme, its ID (a unique identifier for the topic), the path to the project specific issue.

*Services restrictions* – this contains setting restrictions for print jobs, where you can define the largest paper size that can be printed on and the maximum size of the map client window in pixels.

*Print* – includes various settings for printing from the client.

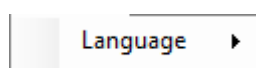
## Debug Console

It is a tool that allows you to create advanced testing of specific project. The basis of the debug console is a dialog box, whose image is shown in the following picture:

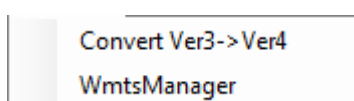


In the left upper part you can see the set of four icons. Using the first one you can save the result in the console window into a text file, the second icon clears all the contents from the debug console, the third icon is used to trace all – it is good to have this feature turned on only when testing specific problems, otherwise it makes work with the project considerably slower. The fourth icon is used to turn on line wrapping to improve the legibility of the contents of the console. The contents of the debug console can be SQL scripts of individual spatial queries, which you can then try to run in SQL editors and thereby detect faults. It can also display queries to WMS services that can also tell you the direction to solving the problem.

#### 4.1.6 Settings

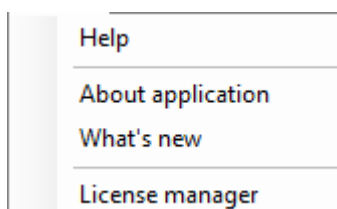


#### 4.1.7 Modules



In menu item modules are registered plugins, it is possible to further work with them.

#### 4.1.8 About



In this menu is after clicking on Manual item displayed this manual in PDF format. Under item About application is the dialog box, containing information about the currently installed version, physical path to the executive modules and registered modules. Under What's new is the PDF file, containing information about the improvements in comparison from previous versions and extensions of functionality. Under the item License manager is a license manager of main product and modules, containing information about the module name, whether the modules are currently loaded, type and license status.

## 4.2 Tools Overview in MarushkaDesign

In this chapter, we will describe work with GSFrameWork objects in application MarushkaDesign. An example of layout of one of the projects is showed in the following picture. The base is the **Data sources** window (1).

Individual data sources are used as a source of specific publish data in a web presentation. Configuration, of which entities from these data sources are published, is performed by **Publish layers** (2).

Properties of the selected object (an object may be in this case, the entire data source, each data source entity, publish layer, possibly also element of any additional library) are defined in the window **Object Properties** (3).

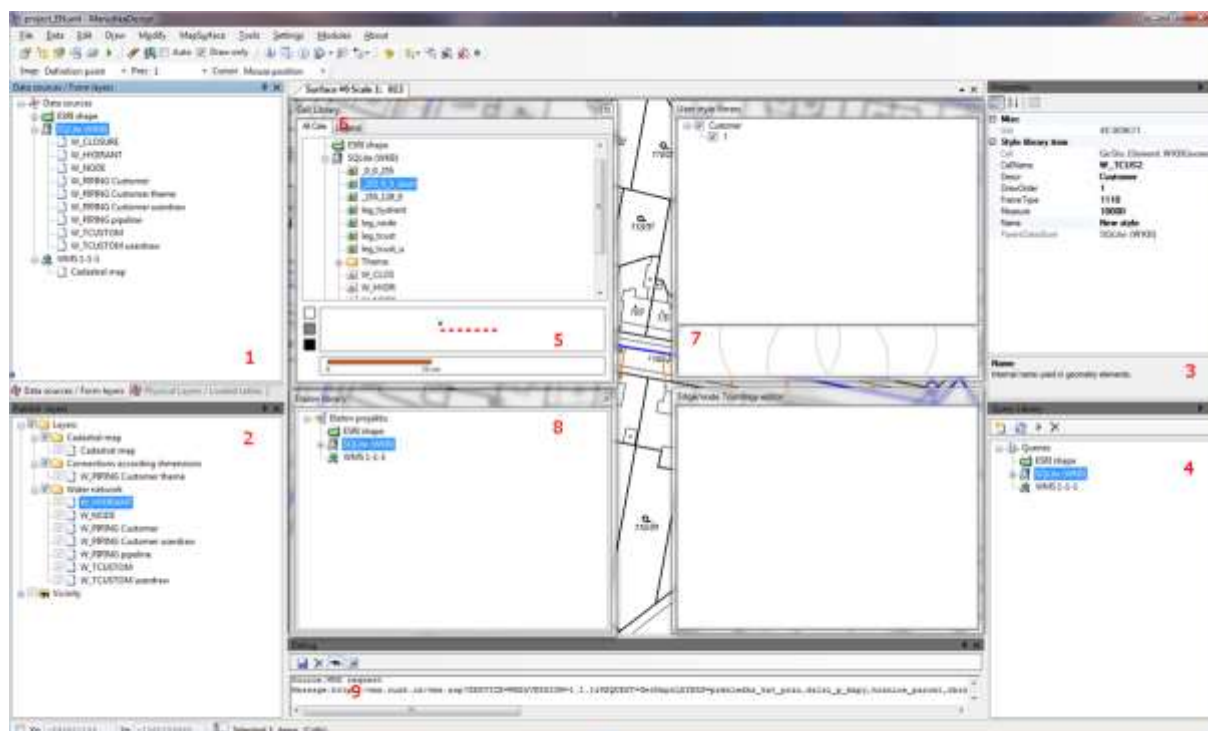
Another, already optional accessory of the web presentation can be various kinds of queries – **Query library** (4) window at the right bottom of the picture – in this window are defined various information queries over the graphic elements, localization queries, or statistical (general queries) without having to choose graphical element.

In case when the cells are included in the project, then their displaying is defined in the **Cells library** (4) - a Cell library is defined for each data source separately. In a dialog box Cell library is also a tab **Legend** (6), which allows displaying static or dynamic legend to the individual categories of the displayed elements.

Another distinguishing feature of the project may be user styles. They can be found in **User style library** (7). User style library is used to display linear objects with user style. Displaying the data in a user style library is based on the selected project – displaying only its user styles.

Another possibility to use data from the different data sources provides **Etalon library** (8). Etalon library is used for validation of data from data stores, resp. for drawing data in a web environment as defined in etalon tables. In the background is hidden MapSurface, displaying the data.

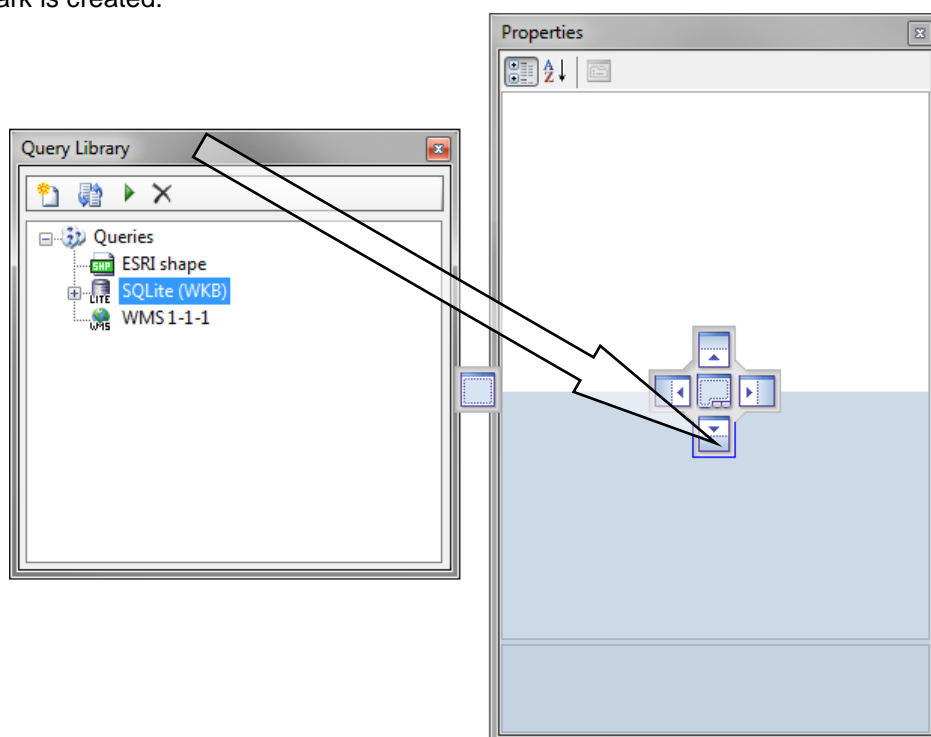
The last important part of the displayed image is a **Debug console** (9). This tool allows debugging individual queries during development and testing of individual projects. The individual components of this environment will be described in the following chapters in more detail.





## 4.3 Manipulation with Dialog Boxes (Windows)

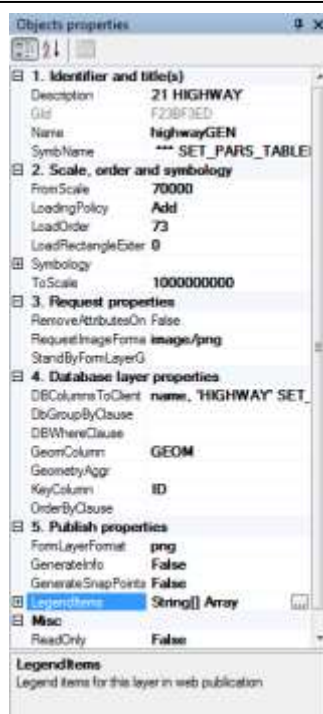
In addition to normal handling with individual windows MarushkaDesign environment offers also advanced window docking. Every opened window can be docked to the edge of the main window or to another opened dialog boxes. In the following picture is an example of an option, how to dock a window to the window of another tool box. By dragging the title box of the dialog box inside another dialog box will display a "cross" in the center of the window. Moving the cursor to the appropriate position of the cross will perform docking of the first dialog window and the second dialog window will be cut off as in a picture below. If you move the cursor into the center of the cross, the size of the dialog box retains and a new bookmark is created.




### 4.3.1 Dialog Box Object Properties

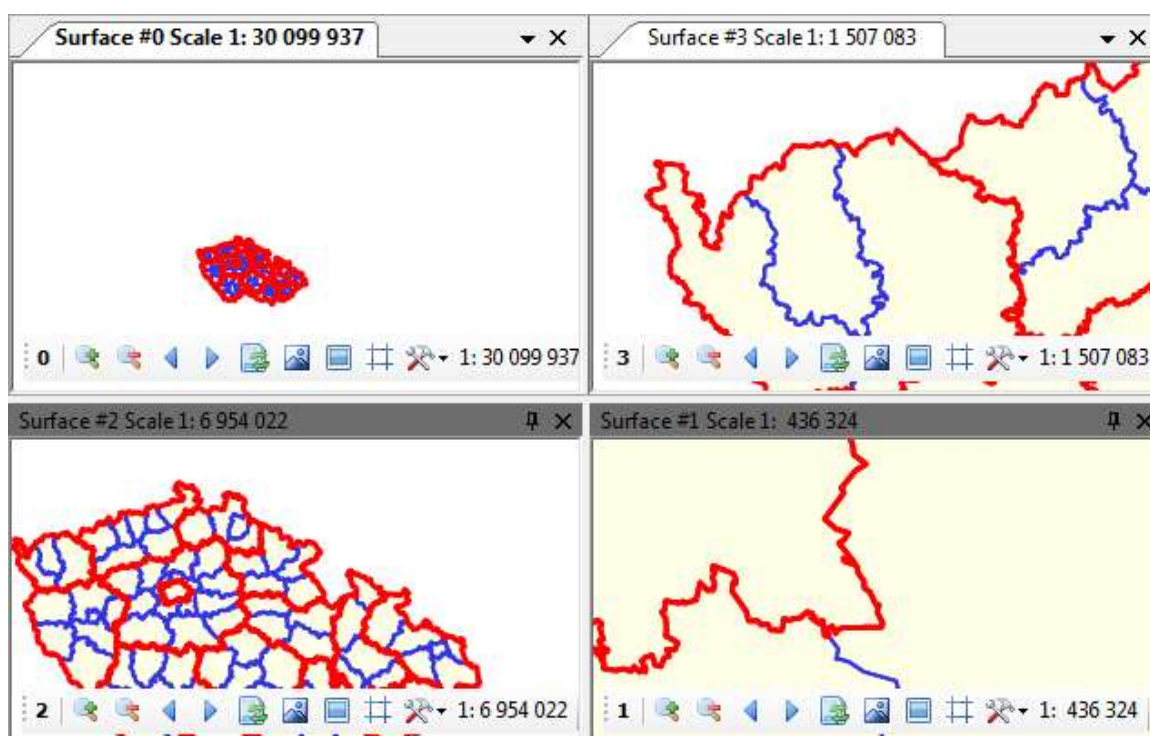
One of the basic and most important components of the MarushkaDesign environment is the dialog box properties of an object. Individual properties of objects are dynamically generated depending on the group to which object belongs. Item in the properties window can be sorted by property group or alphabetically. Sorting by the category feature is sensible and transparent and in most cases of configuration of the object properties also most efficient.

An important feature of this dialog box is to display help for the specific properties of the objects at the bottom. Help is always displayed for the currently selected property.



### 4.3.2 Map Surface

The main map window includes a set of tools for arrangement of graphical windows. It allow previewing the form layers, resp. publish layers by their current configuration. After the first launch of MarushkaDesign, you will find the mapSurface window in the middle of the overall composition user interface. The new mapSurface can be opened using the button *New mapSurface*  on toolbar *Layers and mapsurfaces*. You can therefore open more mapsurfaces and you can switch between them using the tabs in the header, the maximum number is limited to 8. The scale and current scope of the individual windows can be changed independently. Loading the new form layer affects all opened mapsurfaces, the current scale and a displayed cutout remains unchanged.






Retrieving of data is performed by the right mouse click on the relevant form layer. In the form layer dialog box choose the corresponding layer and by the right mouse click choose the option *Data – Load all* to retrieve all the elements of the layer. To retrieve the elements within the current mapSurface cutout choose the option *Data – Load view rectangle*.


The mapsurface window contains the toolbar in the left lower corner. It is used to navigate in the map window or to change the displayed parameters:




 - Index of the active map Surface (0-7)


 - Zoom in


 - Zoom out

 - Goes to previous view

 - Goes to next view

 - Refreshes the view

 - Displays view with all the loaded elements

 - Defines a cutout for displaying

 - Other MapSurface properties

The last button in dropdown toolbar settings allows choosing one of the predefined decimal scales, you can change the predefined color of the MapSurface, turn on / off displaying of auxiliary grid (grid is displayed in scales 1:800 and above), print the window contents (export MapSurface window contents to a bitmap file), and invert the background color of the element.

### Navigation in the Map Surface window

In addition to the controls in the MapSurface window toolbar you can move it using Ctrl key and movement of the mouse. You can achieve the same functionality by holding the middle mouse wheel. Simultaneously by pressing the Ctrl key (or middle mouse wheel) and scrolling the mouse wheel you can zoom in / out the view.

## 5 Basis Objects in MarushkaDesign Environment

### 5.1 Basic Terminology

**Tilde notation** - all the objects that occur in the projects, use for their description and classification the tilde notation that provides a tree object segmentation into individual items. Each tilde symbol (~) indicates one branch in the tree. Branch of the tree can only contain either branch or node (object).

**Project** – is represented by the configuration XML file that store data for the subsequent web data presentation. The project consists of a complete definition of the map composition including settings of rendering of individual layers, definition of individual queries, elements from cell libraries (including their geometry), custom styles, etc.

**Theme** – is defined by the unique identifier ID, under which are stored individual settings. The themes are then displayed in the subsequent web publication. Each theme has its own settings of the displayed publish layers, and its own setting of the initial displayed area. In the whole map publication always appears at least one theme. The maximum number is not limited. One theme matches exactly one xml configuration file of the project. Above each project file is the xml file configuration of the themes, which contain mainly the paths to the individual projects and their initial settings.

**Data Store** – is a sub part of each project. Within each project, you can define any number of data sources that are natural carriers of data. It may be a different type of database resources, vector drawings in wkb, dgn formats etc., georeferenced raster documents, various web services, etc. In the data source root are configured client target units, cartographic projection, resp. whether to sort the legend, if it occurs.

**Physical Layer** – represents the physical storage of geographical data such as the name of a table in a relational schema.

**Form layer** – objects of the form layers represent requirements for the delivery of spatial data from the data store. Requirement may be restricted by the spatial query (window query). It can be, for example SQL query, which retrieves the spatial data in case of relational databases (which may consist of several different physical layers – e.g. view (database view), file / folder with vector data, the list of layers for WMS service. After connecting the data source, will be all the form layers in this data source available. Form layers can be then removed from the data source. It is also possible to add new form layers, clone them for different graphic displaying, set scale range and order of their rendering.

**Annotation layer** - is similar to the form layer; it is used among other things, to display the accompanying map elements, such as watermarks, additional text label, logo, still displayed elements and others.

**Publish layer** – each publish layer contain at least one form layer. Publish layer can contain several form layers that can appear individually (if properly adjusted displaying) or can be displayed several form layers at once. Publish layer is the layer, which is then visible in the web publication.

**Libraries** - libraries are the supportive objects of the project. They are formed by the **Query library**, in which are stored queries on graphic elements, localization queries, viewing queries, or for more complex document management and other types of queries cooperating with the data of various data stores. Then there is a **Cells library** and **Legend library**. The **User Styles library**, which keeps the definition of user styles, **Etalon library** for e.g. acquisition of new data in a web environment, defined by the symbology in this library. Each library is always related to a specific data source; just a legend is common to all data sources – to the entire project.

**Web publication** - is the outcome of configuring the project in MarushkaDesign environment. The basic tool to view the resulting configuration is Marushka (server), which shows the resulting map composition in a web browser.

**GID** - the unique identifier of the object within the whole project. It is usually automatically generated while creating the project, but it is also user modifiable. After modification uniqueness of this attribute must be ensured. Its value doesn't have to be a number – it is a text string. This attribute is then used e.g. for external calling of correspondent object.

**ID** – is a unique identifier for the object within the data store. In some cases is used for "linking" of individual objects (e.g. for queries, when the list of values refers to its master query).

**Information icon** – object displayed in a web publication above the graphical element, which is defined by the information query. Information icon takes the background color from the symbology of the graphic element and contains default text "i", which can be changed by the parameter: SET\_INFO\_ICON\_TEXT, all the information icons can be defined by a raster cell using parameter

SET\_INFO\_ICON\_IMAGE. Information icon can also have its floating help, which is defined by parameter SET\_INFO\_ICON\_LABEL.

**Vicinity (Overview)** – Is used to display the overview maps, usually is displayed using a smaller scale than the current project in the main map window. It therefore shows a larger area than that shown in the main map window and helps to identify the location of the content of the main map window within a higher territorial units or a larger area.

## 5.2 The Project

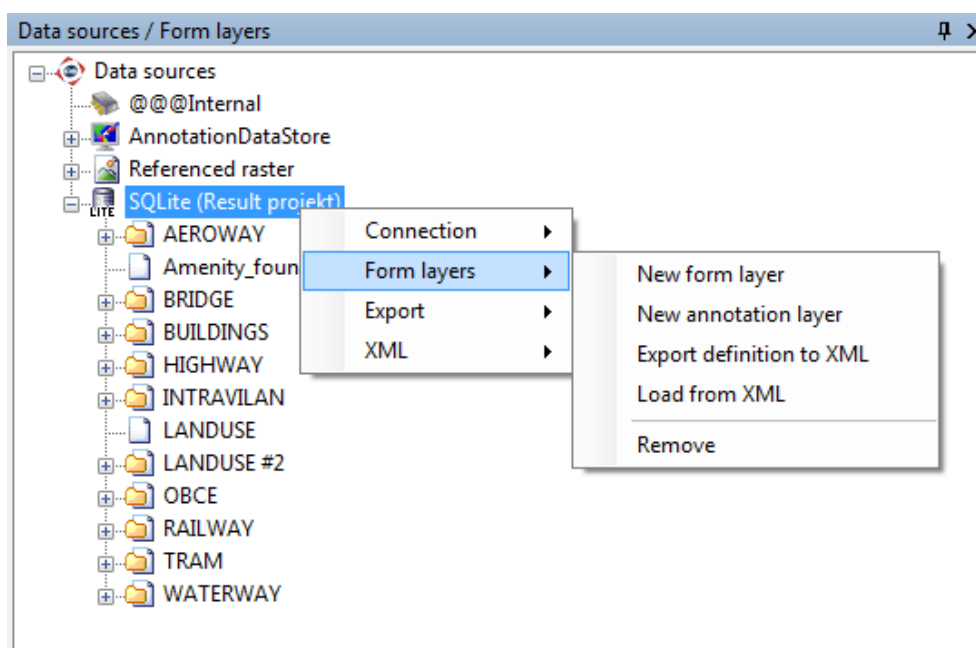
It is a list of data sources (complete structure of map composition), definition of requirements for spatial data (form layer), setting the target projection of the project, cells libraries, queries, user styles libraries and map composition explorer, from which are generated map compositions for various scales. In addition, it is possible to set in the project parameters PassedParams. Project is saved in the configuration XML file, which is created by selecting "Save Project" in MarushkaDesign application.

The configuration file is uniform for all applications based on the GSFrameWork core, e.g. map compositions provided by a server Marushka® are identical to those which are drawn by a strong client MarushkaDesign.

It is possible to edit an XML file outside MarushkaDesign either manually, in any XML editor or Notepad. For this kind of editing is recommended rather advanced knowledge of XML, because improper interventions can cause damage to the project, which could not be working afterwards.

## 5.3 DataStore – Date Source

The data store window is opened from the Tools panel **Layers and mapsurfaces**.



Object DataStore represents an abstract data source that provides access to geographically referenced data in vector or raster form. The data is then inquired through form layers. The basic properties of specific data stores are inherited from this object. Currently supported data stores are:

- Relational databases (database data stores):
  - Open GIS Consortium WKB format ORACLE
  - Open GIS Consortium WKB format Microsoft SQL Server
  - ORACLE Spatial Data Option
  - MySQL WKB format
  - PostgreSQL

- SQLite WKB format
- Vector files (file data sources):
  - Open GIS Consortium GML format
  - Intergraph / Bentley DGN format
  - ESRI Shape File format
  - OGC - WKB file including support of spatial selections
  - Topographic GPX (saved tracks from GPS devices)
- Geographically referenced raster files:
  - TIFF, GeoTiff, ECW, JPEG, PNG, GIF...
- Web Map Services and Web Feature Service by the OGC specifications

DataStore object contains mainly:

- Data store type (this chapter)
- Data required for the connection
- Library of symbols, cells (see [Cell Library](#))
- User Style library (see [User Style library](#))
- Etalon for drawing and data validation (see [Etalon library](#))
- Saved queries (localization, information) (see [Query library](#))
- Source cartographic projection (see below)
- Units in which the source provides data (see below)
- Queries on the geometrical data, layers (form layers, see [Form Layer – requirement on spatial data](#))
- Physical layer data (Layer, see [Physical layer – physical data storage](#))
- Publish layer (see [Publish layer](#))

### 5.3.1 Cartographic Coordinate Systems

#### 2. Projection a units

Current Transform	JTSK->JTSK
SourceProjection	JTSK
SourceUnits	millimetre
TargetProjection	JTSK
TargetUnits	millimetre

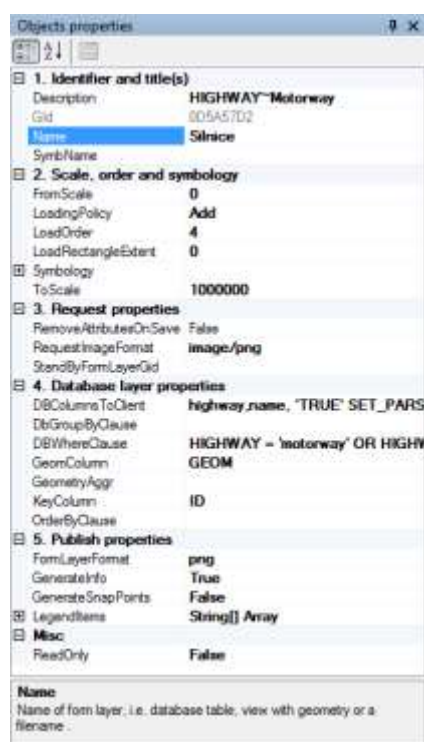
For

the data source is

assigned a code of cartographic projection, with which this source provides geographical data. Using cartographic transformations that are a part of core of the system can be transformed into the target projection from two different sources. In the data store properties Projection and units can be set the source and target projection. **Changing the target projection will change the projection of all connected data sources.** The target projection can be just one. Currently, the core supports following cartographic projections:

- S-JTSK
- S-JTSK/05
- Mercator
- UTM
- S-42
- WGS84
- ETRS
- ETRS-LAEA
- NAD83/California
- SWEREF99
- KKJ/Finland
- ED50/TM33

### 5.3.2 Form Layer – The Requirement for Spatial Data



In the sample dialog box properties are shown different adjustable parameters of a single form layer (example of form layer from database data store).

#### 1. Identifier and Title(s)

- a. **Description** – verbal description of form layer, it is possible to use tilde notation for tree displaying of individual layers
- b. **Gid** – the unique object identifier, object must be unique within the project
- c. **Name** – The name of the layer corresponding to the form layer name of the data source (in parentheses can be given complete SQL query (select must contain system columns

of database data source), which will further specify the entity, e.g. with suitable where clause).

- d. **SymbName** – additional description to distinguish different symbologies in one form layer

## 2. Scale, Order and Symbology

- a. **FromScale - ToScale** – scale range of the specific entity
- b. **LoadingPolicy** – only for database data sources – *Add* – layer will be loaded as a separate image, *Merge* – for all layers of the same name and the same columns loaded to the client station (layers can vary in where clause) is created just one query. In the case of using the Merge policy should all the layers have the same LoadOrder otherwise it will be used practically at random only one of them.
- c. **LoadOrder** – order of rendering form layers
- d. **LoadRectangleExtent** – extension of the spatial query in source units.
- e. **Symbology** – definition of user symbology for displaying form layers – setting colors, filling, transparency, text size, cells (see picture below)

## 3. Request Properties

- a. **RemoveAttributesOnSave** – when exporting to WKB file it removes the element attributes
- b. **RequestImageFormat** – Required graphical format (just for web services WMS) *png* - efficient, but with a narrow range of colors (256 colors) – suitable for vector data, *jpg* – suitable especially for ortophoto, less economical, in less color layers is too compressed and unaesthetic
- c. **StandByFormLayerGid** – backup layer (GID), which is used in case of unavailability of the layer

## 4. Database Layer Properties

- a. **DBCColumnsToClient** – This item is a list of values (including their aliases) that are tightened into geometric elements on the client workstation or publish server. Some of them affect the graphical presentation of geometry (`SET_PARS...`, `SET_ENV_...` – see below). This item can be edited directly or with a special editor that is activated by clicking on the edit button at the end of the line. The whole structure of the query, which is then sent to the server, is following: `SELECT SYSCOLUMNS, DBCColumnsToClient from LayerName where WindowQuery and DBWhereClause group by DbGroupByClause order by OrderByClause`. When using parameters it is necessary for safety to provide their types. Parameter names are prefixed with tilde and their syntax should look like this: `~ (type) name ~`, where type can have the following values: **string** (text string), **int** (sign 32-bit integer), **long** (sign 64-bit integer), **decimal** (128-bit non-negative real number with a precision of 28 digits with a lower range), **double** (8-bit nonnegative accurate to 15 decimal places with high range), **datetime** (date data type), **idlist** (list of primary keys). Among other parameters, that you can use are Passed params defined in the data stores that can be called directly from URL (e.g. [http://\\*url\\_serveru\\*/default.aspx?Themeld=11&BARVA=255%200%20255%200](http://*url_serveru*/default.aspx?Themeld=11&BARVA=255%200%20255%200).)
- b. **DbGroupByClause** – group by clause for geometric queries
- c. **DBWhereClause** – where clause restricting loading of this layer
- d. **GEOM** – name of the column with geometry of the object
- e. **GeometryAggr** – geometric aggregation function
- f. **KeyColumn** – column name with the primary or unique key
- g. **OrderByClause** – clause "Order by"

## 5. Publish Properties

- a. **CoversAll** – indication, that this layer will cover all, loading layers with lower priority order will be suppressed in a web publication
- b. **FormLayerFormat** – image format for web publishing. Valid values are: JPEG=0, PNG=1 (24 bit png), PNGQANT=2 (8 bit png), PNGRAWQANT=3 (8 bit png with possibility do define transparency). This setting is important especially in combination of vector layers with e.g. ortophoto, when you need to quickly and effectively display combination of ortophoto map and vector drawing. For ortophoto set the priority JPEG, for vector e.g. priority PNG. If you activate only the vector, as a result you will get a PNG



image (in a very good quality of vector drawing). If you activate only ortophoto map image in JPEG format. In the combination of both layers you will get the precedence JPEG format and vector drawing might be slightly distorted. But the resulting image will be readable without much vector distortion.

- c. **GenerateInfo** – permission for generating info icons for the web publication
- d. **GenerateSnapPoints** – Generate snap points
- e. **LegendItems** – Legend items for this layer in a web publication (detailed description of work with the legend will be described below – see Legend)

In the lower part of the window is a brief description of active properties.

### 5.3.3 Symbology

To set the form layer symbology is used item Symbology in category 2. *Scale, order and symbology* in the form layer properties. Item *Symbology* is divided into five sub-categories, depending on what type of element the user wants to edit – *Fill* (area), *Line*, *Raster*, *Symbol* and *Text*. In the case of using specific attribute, this attribute is added to the box next to the group name, the names of individual attributes are separated by semicolons. Overall view of all the attributes of the category Symbology is available in the next page figure. Nearly all the symbology items of form layers can be now modified in bulk, which greatly saves time when working with the project.

**DebugDraw** – displays the system information for the element, it is possible to choose one of five options (*MBR* – limiting rectangle, *SnapPoints* – snap points, *ReferencePoint* – reference point, *Org* - geometry, *RangePolygon* – polygon range)

#### a. **Fill** – area fill symbology settings

*HStyle* – fill symbology style  
*RgbFColor* – fill color  
*RgbHColor* – fill symbology style color  
*Texture* – fill texture (raster)

**LayerOpacity** – setting overall form layer transparency (in %); 100 – absolutely opaque layer, 0% - absolutely transparent layer; this transparency setting takes precedence over all parameters *SET\_PARS\_* or before the properties set through symbology

#### b. **Line** – line elements symbology settings

*LineRealWeight* – line width (in source units)  
*LineWeight* – line width (in pixels)  
*RgbColor* – line color  
*Style* – line style  
*Ustyle* – line user style

#### c. **Raster** – raster symbology settings

*Brightness* – brightness settings  
*Colorize* – colorize raster  
*Contrast* – contrast settings  
*DrawingInterpolationMode* – drawing interpolation mode  
*DrawingPixelOffsetMode* – mode of drawing images – the user sets quality, high quality images take longer to load and logically, low quality images take shorter to load  
*InvertColor* – invert raster color  
*RasterOpacity* – raster opacity (in %)  
*TransparencyType* – type of raster opacity – it is possible to choose *OneColor* – when only the one color gets transparent or *RangeTwoColor*, when the whole transition from one color to another gets transparent.  
*TransparentFrom* – transparency from color – it is possible to use color picker, when it is possible to select any object on the screen or select color from the

*TransparentTo* – transparency to color – the color can be selected in the same manner as in preceding item

**d. Symbol** – symbol symbology settings

*DrawUnits* – units used for displaying symbols

*SymbolName* – cell name from the cell library

*SymbolScale* – symbol scale factor

**e. Text** – text symbology settings

*DrawUnits* – units selected for drawing

*FontFamilyName* – text font name

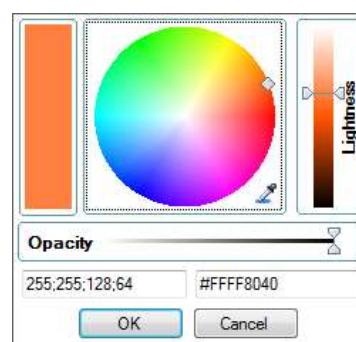
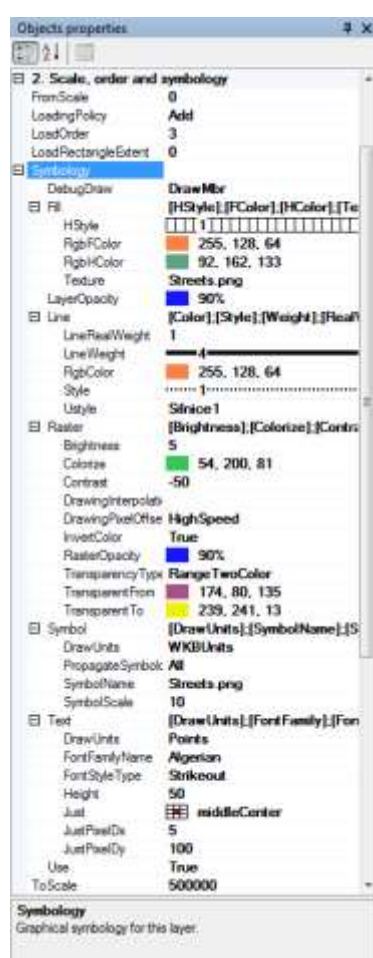
*FontStyleType* – font style type

*Height* – text height

*Just* – text justification type, there are 9 possibilities

*JustPixelDx* – shift of the text reference point on x axis (in pixels)

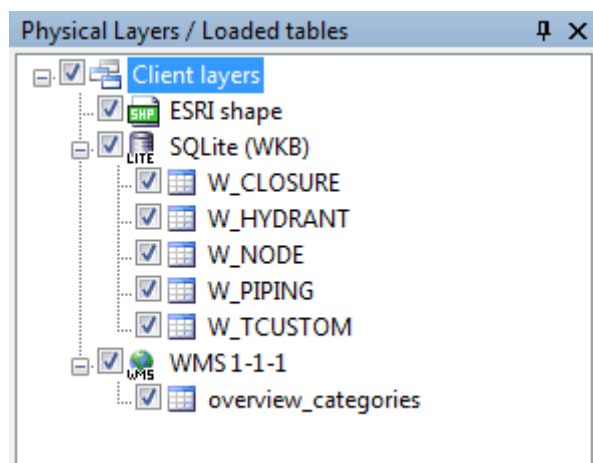
*JustPixelDy* – shift of the text reference point on y axis (in pixels)



### 5.3.4 Physical Layer – Physical Data Storage

**Physical layer** represents physical storage of geographical data, such as a name of the table in a relational schema. List of physical layers is updated automatically upon formal request to download data into the graphical window. Physical layer is enabled by mouse double click on the item and active layer is then highlighted. In active physical layer can be then drawn other elements (see *Draw* menu), then place them into individual data sources or the data can be moved between different data sources. The objects in the physical layers are only temporary objects specifically designed to work in MarushkaDesign environment and they are not a direct part of the project.





The physical layer includes especially the following:

- A list of loaded geometric elements
- Structures for fast spatial access
- Methods for spatial selections combined with any predicate, which relates to the geometric attributes of the element
- Up to 8 symbology options for graphical presentation (e.g. more graphical windows)
- The display order in the client application

### 5.3.5 Publish Layer

It is a named list of form layers for publication purposes of the HTML client and also other supported services (WMS, WFS, TILE). The end user has just an offer of these named lists – publish layers. Each publish layer can then include a group of form layers displayed in various scales. In the publish layers is again used tilde notation for tree view of individual layers. It is possible to add comments in the form `/*comment*/`. These comments can be displayed or hidden, when these comments are hidden, they can be used to sort publish layers without visible interference in their name for the end user. The user can hide or display these comments using checkbox in header of dialog box publish layers.

#### a) HTML Publish Properties

Each publish layer contains the layer properties for its web publication:

2. Marushka HTML publish properties	
CheckedGroup	
DefaultChecked	True
DefaultCheckedInfo	True
DefaultLayer	False
GenerateInfo	True
InfoQueries	(Collection)
Popular	False
Visible	True

**DefaultChecked** – publish layer will be displayed after the initialization of web publication (true).

**DefaultCheckedInfo** – Information icons will be displayed in web publication (true).

**DefaultLayer** – this layer will be always rendered (true). This setting is applicable for example for various watermarks and annotations in case where the value of the item is set to False. Such settings are reflected so that the layer will always be rendered, but will not appear between the layers in web publication.

**GenerateInfo** – permission to generate info icons for web – this setting is done for every form layer separately.

**CheckedGroup** – Layers in this group can be exclusively checked. In a defined group may be checked just one layer (in web publication on these layers are not displayed checkboxes, but radiobuttons). The title of this group is then displayed in a web publication as a floating help.

**InfoQueries** – Allows you to display additional information for the corresponding publish layer. In case that you have created a defined information query here (GID parameter of relevant query), then in a web publication at the publish layer appears an icon allowing to display information about this layer. It may be a redirection to a web page, it can display another important information about this layer, e.g. in html template defined by called query.

**Popular** – For this layer was created a special button in a web publication placed in the top part of client. Text displayed in this button, is a string behind the last tilde. You can also assign another label to the popular layer. It will appear as a floating help for this button. This text can be entered in field AbstractTitle in category 1 – Identifier and titles.

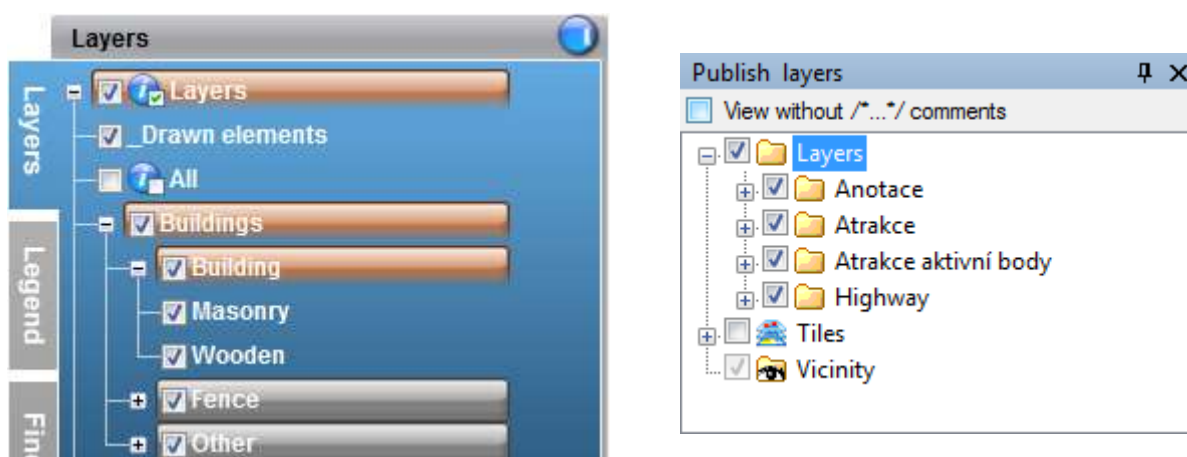
**Visible** – Publish layer is visible (true) in the tree structure of publish layers in the web publication.

## b) Web Publish Services

4. OGC Web Map Service	
AllowWMS	True
GetFeatureQueries	(Collection)
5. OGC Web Feature Service	
AcceptQueries	BBox
AllowWFS	False
MaxRequestArea	
WfsFeatureSchemaLocation	
WfsFeatureTypeNameSpace	
WfsFeatureTypeType	BasicFeatureType
WfsMaxFeatures	-1
6. OGC Web Map Tile Service	
AllowWMTS	False
BackgroundColor	0, 255, 255, 2
TileCleaner	GeSto.Engine.WM
WmtsRootFolder	
WmtsTileExtent	0
WmtsTilesCashed	False
WmtsTilesCashedMinLevel	
WmtsTilesFolder	
WmtsTilesIndex	Not set

In the web publication is then recognizable scale range thanks to the graying of publish layers in the case of its scale unavailability.

In the following image on the left is not available (because of the scale) information layer for the publish layer Cadastral map. When creating publish layers, in the default publish layer window are predefined folders Tiles and Vicinity, other publish layers, however, the user must the user must define himself. The user may also make work with publish layers easier, by right click on a single layer and chooses *Layer – Form layers – Restore pub layers from form layers*. This function copies the structure of form layers and creates publish layers from them. If the publish layers has in the dialog box *Publish layers* checked the checkbox to the left to the folder picture, so these layers will be loaded during action Clear and load by policy from toolbar *Load data*.



### 5.3.6 Annotation Layer

Annotation layer is similar to the form layer; it is used among other things, to display the accompanying map elements, such as watermarks, additional text label, logo, still displayed elements and others. It is mostly used for elements, which the user want to display all the time and he does not want to link them with displayed form layers or it would be difficult to display them in other way.

In category 2. Scale, order and symbology it is possible to define symbology for annotation layer. In the case, that it is not necessary to display annotation layer in all the scales, it is possible to set scale range using definition of parameters *FromScale* and *ToScale*.

If it is desired to display using the annotation layer a particular element, but the user do not want to display the element in the list of annotation layers, so in the publish layer properties 2. Marushka HTML publish properties set item *DefaultLayer* to value 'True' and other items to 'False' and items with other possible options left blank.

In the same category it is possible to define following items:

*CellName* – name of cell displayed using annotation layer

*Just* – justification of annotation

*StartOffsetX* – the starting offset in the X axis (in pixels)

*StartOffsetY* – the starting offset in the Y axis (in pixels)

*StepX* – step for repeating the annotation in the X axis (in pixels)

*StepY* – step for repeating the annotation in the Y axis (in pixels)

*Text* –text displayed through annotation layer, it is possible to enter here a combination of multiple parameters SET\_ENV\_, separated by commas

*Type* – type of annotation

## 6 File Data Sources Properties

### 6.1 WKB

WKB files can be loaded into MarushkaDesign environment either individually or as a whole folder. In case, that is connected component or entire file folder structure, then the components will form the branches and individual files tree nodes of form layers. WKB files are used only as a vector source; it is possible to use attributes, which they carry along with themselves. WKB files must be indexed for faster access. In the case of missing indexes can be used feature in the properties of the relevant data store in parameter **Actions** function **CreateMissingIndexes**.

Connected WKB files can be then exported as a subproject and then used in other projects.

### 6.2 SHP

Using SHP is not the most appropriate. Due to the amount of data stored within them is however necessary that MarushkaDesign environment allow loading this source. After loading SHP it is advisable to export them to the data store with faster access, such as a SQLite database.

### 6.3 DGN

Sources of DGN drawings are again used mostly for import into the MarushkaDesign project and then converting them to more suitable data source. The disadvantage of DGN drawings as a map surfaces is primarily inability of spatial indexing.

### 6.4 Raster Files

Raster files can be connected individually or as a group of files or an entire file folder. In case, this group of files will be connected, a pair of auxiliary files index.wkb and index.frx will be created, which will greatly accelerate subsequent reading of grids. Because first will be loaded the index files and subsequently will be loaded the raster files.

For loading raster drawings (especially TIFF format) it is advisable to deal with their optimization. TIFF does not have to include thumbnails for faster access and therefore is required to create them by an external application. First find statistics about the file (<http://www.gdal.org/gdalinfo.html>), if you find out that TIFF thumbnails are missing, then you can create them using gdaladdo (<http://www.gdal.org/gdaladdo.html>).

### 6.5 WMS

Of course there is a possibility of WMS connection (web map service) as a further mapping mapSurface source. Individual WMS layers can be merged into one layer – for example create a new form layer of data source and into field Name list the different wms layers separated by commas. During such a publication, then there is only one requirement sent to the appropriate wms server. Similar functionality can be achieved by using policy *Merge* (category 2. *Scale, order and symbology*) for each form layer. In this case however, it is not possible to change the symbology of downloaded data. You will then get a wms result in a format that you can choose in the parameter **RequestImageFormat**.

### 6.6 Remote Databases

The advantage of this approach is that it does not require database client and it is possible to use external data stores, but they must support GSFWWS service. This service allows you to connect to the database through http protocol. GSFWWS is a separate commercial product of GEOVAP Company.

## 7 Properties and Possibilities of Database Data Stores

### 7.1 Sequences and System Objects for OGC WKB Format

For WKB format support by the Open GIS Consortium is needed to execute SQL scripts `gs_sys.sql`, as well as in products GS V5 and GS V6. Graphical tables whose structure is based on the database, in which they were created, slightly differ. They all have common bases. Each graphical table must have a primary key **ID** unique within the entire data store, and also columns for spatial object identification – limiting rectangle of element (**xmin**, **xmax**, **ymin**, **ymax**) and spatial object key for faster identification and indexing of particular database objects (**SPAT KEY**, **SPAT INDEX**).

### 7.2 Dynamic Columns and Columns SET\_PARS\_

Just like previous versions of GeoStore do products GSFrameWork support dynamic generation of texts and symbols based on the attributes of geometric objects. For improving performance, it is recommended to insert list of dynamic columns `GS_DYN_COLUMNS` into table `(TABLE_NAME, COLUMN_NAME)`.

GSFrameWork also support other dynamic properties of geometric objects in the following manner. This dynamic symbology changes are defined in the properties of form layer in category 4 – *Database layer properties – DBColumnsToClient*, so whenever you in the query to the geometric object experience any of these columns, according to the content the corresponding property of the object is changed.

Column	Geometric Element Attribute	Format
SET_PARS_TEXT	Content of text in point element	String
SET_PARS_PIXELSIZE	Absolute size in pixels	String [True, False]
SET_PARS_HEIGHT	Point element text height	number
SET_PARS_ROTANGLE	Rotation angle of text / symbol in radians. For a definition in degrees must concatenate the column with prefix '(DEG)'. Ex.: '(DEG)'    angle, where angle is column containing the rotation angle in degrees	number
SET_PARS_SCALE	Scale factor for symbols	number
SET_PARS_ROTMATRIX	Transformation matrix for symbols	4 x number
SET_PARS_WEIGHT	Line thickness	number
SET_PARS_STYLE	Line style	number
SET_PARS_POINT_FROM_CORG	Creating a point element (text) from non-point element is used for the generated text	dx [dy [just [tol[isText]]]]  dx, dy shift in axis X and in axis Y  Meaning of the parameter just:  leftTop - 0 leftCenter - 1 leftBottom - 2 middleTop - 6 middleCenter - 7 middleBottom - 8 rightTop - 12 rightCenter - 13 rightBottom - 14

		<p>tol – tolerance of generalization for linear elements</p> <p>false - is ignored</p> <p>isText – true/false optional parameter, strictly suppress the text element</p>
SET_PARS_MASK_FROM_AREAL	<p>Creating a mask from a single ring polygon ("Polygon with a hole") Ex.: '-917786000 -1241062000 -415828000 -889938000'</p> <p>SET_PARS_MASK_FROM_AREAL</p>	'xmin ymin xmax ymax' – limiting mask rectangle
SET_PARS_RGBCOLOR	Line color	4 x number (ARGB)
SET_PARS_RGBFCOLOR	Fill color	4 x number (ARGB)
SET_PARS_RGBHCOLOR	Color pattern color	4 x number (ARGB)
SET_PARS_USTYLE	User style	string
SET_PARS_HATCHSTYLE	Fill style	number
SET_INFO_ICON_IMAGE	Name of the raster cell that is used for HTML clickable icon in publication	String
SET_LEG_ITEM	Name of the raster cell, which is used as a legend item in a HTML publication	String
SET_INFO_ICON_LABEL	Text that automatically appears after pointing the mouse cursor on the symbol in HTML publication	String
SET_INFO_ICON_TEXT	Text, which is inserted into info icon instead of "i"	String
SET_INFO_ENABLE	If FALSE, the generating bubbles for the element are canceled	String [True,False]
SET_INFO_ICON_COVER	Point element is covered by a transparent info icon	String [True,False]
SET_INFO_ICON_LEGEND_ENABLED	Excludes adding icons to legend info for the element with the value FALSE	String [True,False]
SET_PARS_COLLECTION_BREAK	Breaks cells into individual elements, each of them is cropped	String [True, False]
SET_PARS_CROP_OFF	Prohibition of cropping geometry to fit in window	String [True, False]
SET_PARS_WIDTH	Line width in units of data store	number
SET_PARS_REPLACE_COLOR	<p>Replaces the color of the element. Parameters: CFH gap original color 4 x number (ARGB) gap new color 4 x number (ARGB) C: Color, F: Fill color, H: Hatch color</p>	String
SET_INFO_ICON_NOQUERY	<p>Displays info icon, even in the case, that there is no query for the given element. Parameters: String [True,False] Example: 'TRUE'</p> <p>SET_INFO_ICON_NOQUERY</p>	String [True, False]
SER_PARS_CELLNAME	Sets CELLNAME from the Cell library for loaded element	String
SET_PARS_DRAWSMALLPARTS	When object is too small draw engine draw just small line. Default value is FALSE, When TRUE draw	String [True, False]

	engine draw small object properly (Slower)	
SET_PARS_TABLENAME	Sets (forces) TABLE name for loaded element	String
SET_PARS_IMAGE_FROM_CORG		String
SET_PARS_JUST_PIXEL_DX	Move of text origin in pixels in x axis.	number
SET_PARS_JUST_PIXEL_DY	Move of text origin in pixels in y axis.	number

### 7.3 Dynamic Parameters of SET\_ENV\_ environment

Individual parameters SET\_ENV\_ are used for substitution of individual queries. When using any of these parameters, you must also define such a parameter in Passed Params of the project and enter the default value. Without this definition could happen that server won't know the value of parameter SET\_ENV and the outcome could end by an error condition.

The values of these parameters cannot be changed by user from the outside, these are dynamically replaced by the server (their value changes over time), if they occur in the database query, e. g. invoking the print job at SET\_ENV\_PRINT.

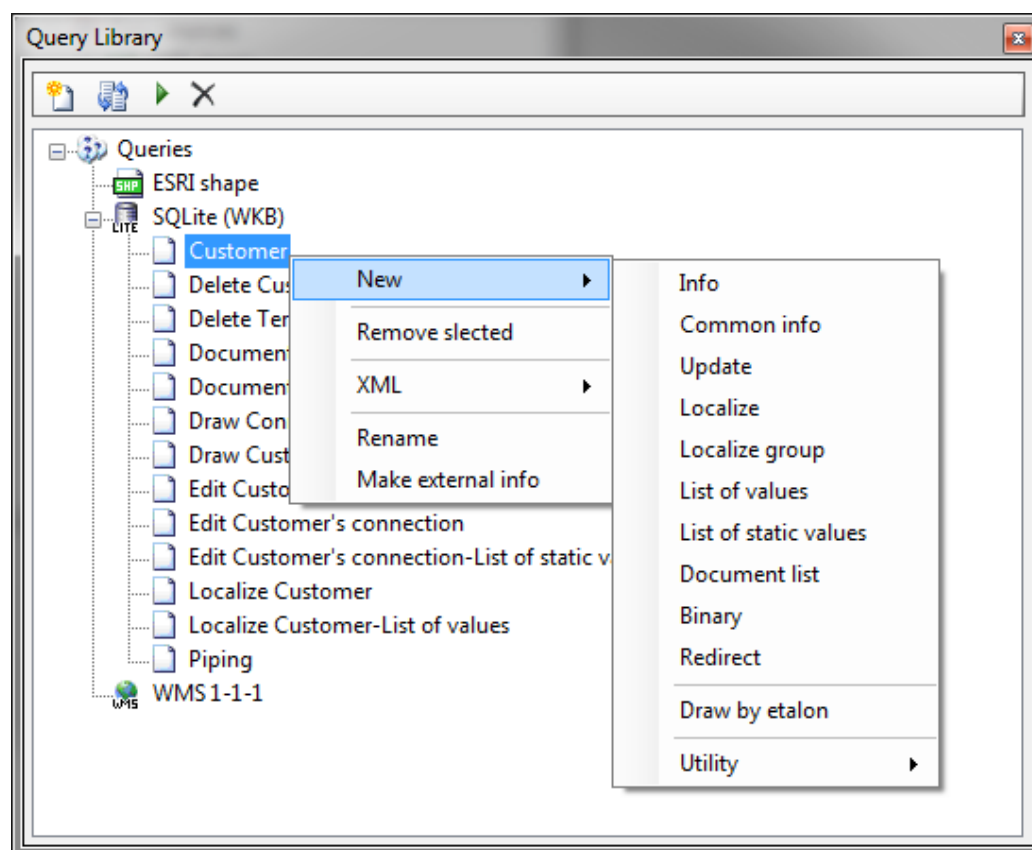
It is possible to combine individual parameters in property text of annotation layer, so that the annotation can display more parameters.

Column	Geometric Element Attribute	Format
SET_ENV_USER	Parameter that can work with a username of authenticated user (in case of authorized access to the data store)	String
SET_ENV_SCALE	Parameter used to define the scale range of displayed layer	Double
SET_ENV_ISPRINT	Parameter, that allows different behavior of publication when the print job is invoked	Boolean
SET_ENV_MACHINENAME	Parameter containing the name of user computer	String
SET_ENV_THEMEID	Parameter containing the ID of the current theme	Int
SET_ENV_DATETIME_DATE_DD_MM	Parameter containing current date in format: <i>Day-Day Month-Month Year</i>	String
SET_ENV_DATETIME_DATE_MM_DD	Parameter containing current date in format: <i>Month-Month Day-Day Year</i>	String
SET_ENV_DATETIME_TIME	Parameter containing current time in format: <i>Hour-Hour Minute-Minute Second-Second</i>	String



## 7.4 Query Library

The Query library is opened in menu *Tools – Libraries – Queries*.



Context menu, which is activated by the right mouse click in a list of queries, enables the insertion of new query, query removal, loading queries from a database table, moving queries between data stores and export / import of selected queries into xml file.

Individual queries are stored in the configuration file of the project, eventually in the WWWQUERY table or other table with the same structure. Table WWWQUERY can store only certain types of queries. Existing library of queries can be further exported from the project and imported back into the project.

The bases of all are practically only two families of queries whose properties are inherited by their further descendants. These two categories are the most important part of the use of non-graphical information on individual elements. These are the **information** and **localization** queries.

**Information** queries are the supporting base for obtaining information about single objects, statistics of objects that share common characteristics (e.g. statistic of lengths of engineering networks in a defined area) and their capabilities are very extensive. In the web publication is possible to open the result of an information query either in a bubble, in the panel on the right side of the web publication, possibly in a new window of the web browser, where you can generate the entire HTML page by defined templates.

SQL query defined in the property `SqlStmtTemplate` contains parameter `~(long)ID~`, which refers to the primary key of the table specified in the property `LayerName` to which the key is related. The value in table `LayerName` provides a formal link with the database layer of the data store, to which the query is linked. This layer will then have the relevant query available. In the properties of the form layer, you can define if for this layer will be enabled information icons, which can be then displayed in the web publication (their displaying can be switched on / off). The appearance of information icons may be standard (defined by application), or user defined in form of any icon of any color. In case of a large number of graphic elements the icons can unite into one, and the resulting icon will then be distinguished by a "+". In the web publication can be the query triggered also in information mode when it is not necessary to display informational icons. In both cases, the following elements from around the cursor can be "traveled" through by clicking the right mouse button.

**Localization** queries are the second important category in publishing data. The result of the localization queries is a territory (limiting rectangle) and the layer highlighting the localized object, resp. group of objects. Each localization query may include several parameters. Parameter of the query can be



a simple text object, or it may be dependent on dial entries. In the second case, it is possible to use search suggestion function, which is offered on the basis of written text. In the web publication is displayed in the card "Search".

**Update** query type belongs to the family of information queries and it inherits most of the properties from them. This type of query is used for database updates of specific dates. Another type of query for direct editing of an existing element in the base is query Delete element.

A pair of queries **List of values** and **List of static values** serves as auxiliary queries to localization queries and queries type Update. These queries define the permitted values that are on an input as user-defined parameters.

Another group of queries are **documents queries**. This group consists of three types of queries. **Binary** – used to select the document associated with geometric table element. **List of documents** – is an auxiliary query defining a list of documents and options of its displaying. **File browser** – advanced database query allowing a document management related to the geometric table element. Query allows you to view, insert and remove documents.

The separate type of query is **Redirect** enabling the move to parameterized website.

Another specific type of query is **Draw by etalon** allowing drawing new elements in the web client. These elements are drawn by the graphic symbology, which is defined in the etalon of the project. Newly drawn elements may have filled any element attributes of the resulting select. In the same category could be classified also another type of query **Draw and locate WKT string**. This type of query is used to localize and render temporarily the graphic element that is defined by the input parameter – text WKT string.

## 7.4.1 Interconnection with Database Table WWWQUERY

The compatibility of the projects GeoStore V5 and definition of database queries from table wwwquery was partly maintained. Due to the extensiveness of the changes it was not possible to keep one hundred percent compatibility of the data and the initial load of the older project will not be preserved in full data consistency. In the event that there is a database table wwwquery, it must contain the following columns and their desired values. Need to enter the values in each column depend on its type.

QUERY\_TYPE – Definition of query type (0 – Information query on graphical element, 20001 – Query to documents, 22001 – Auxiliary query to select the document from DB, 1 – Localization query, 1003 – Bulk localization, 3 – List of offered values).

ID – unique id in the database query within the database data store

QUERY\_NAME – query name, a tilde notation is used

QUERY – SQL query

TABLE\_NAME – identification of GS table, to which is the query associated

QUERY\_LV – id of associated localization query

QUERY\_LV\_NUM – index of parameter of localization query

QUERY\_BUF – maximum number of results per query, the default value is 1

QUERY\_PARAMS – a list of parameters (code lists, input fields), parameters are separated by a ";"

REPORT – template for a result (optional)

## 7.4.2 Information Query on Graphic Element

This type of query is the basic query type allowing displaying any database information, or it's editing.

### Query properties

AttributeName – query is assigned to an element of the layer only if it contains the attribute and its value is equal to AttributeValue (optional)

LayerName – form layer name, to which is the given information query assigned

QueryBuf – maximal number of the results of a single query, default value is 1

ResultTemplate – HTML template of the result (optional)

SlqStmtTemplate – SQL query template. Query must include the parameter ~(long)ID~, this parameter is before evaluating substituted by the ID value of the graphic element. Each selected column must be named; this name is then displayed as the name of the result table.

## Example

```
SELECT
  A.ID "ID",
  A.TYPE "Type",
  A.LOCATION_TYPE "Type of placement",
  A.TEXT "Note"
FROM
  CP_CHECK_POINT A
WHERE
  A.ID=~(long) ID~
```

## Displaying of the Result

ResultHeight – height of information bubbles, resp. window (in pixels)

ResultWidth – width of information bubble, resp. window (in pixels)

TableStyle – type of displaying query in client

TargetName – name of the html window. This is a standard parameter Target of the opened link in HTML code, which is used for integration between HTML applications

ViewStyle – type of displaying query in a client

- InPanel – information tab on the right side of the web publication
- InPopUpBubble – in a "bubble" near the element
- InNewWindow – opens a new window and the result can be displayed as any website

WindowLeft – left indent while opening the new window (in pixels)

WindowOrigin – the starting point of the window for calculation of the position

WindowTop – above indentation while opening a new window (in pixels)

## 7.4.3 Common Info

This query allows displaying common information without selecting a graphic element. In the web publication is this type of element displayed by default in an Information tab, it is also possible to set the query result to display in new window or in the information bubble, this type of query is not associated with the element LayerName, therefore it is not necessary to enter it.

A query can contain parameters ~i~ ... ~j~, where i ... j are indexes of the order of parameters of the associated query.

## Query Properties

QueryBuf – maximal number of the results per query, default value is 1

QueryParameters – definition of the parameters, which are used as parameters for searching in SQL query.

ResultTemplate – HTML template of the result (optional)

SlqStmtTemplate – SQL query template

## Displaying of the Result

ResultHeight – height of information bubbles, resp. window (in pixels)

ResultWidth – width of information bubble, resp. window (in pixels)

TableStyle – type of displaying query in client

TargetName – name of the html window

ViewStyle – type of displaying query in client

- InPanel – information tab on the right side of the web publication
- InPopUpBubble – in a "bubble" near the element

- `InNewWindow` – opens a new window and the result can be displayed as any website

`WindowLeft` – left indent while opening the new window (in pixels)

`WindowOrigin` – the beginning of the window for calculation of the position

`WindowTop` – above Indentation while opening a new window (in pixels)

## 7.4.4 Query Type Update

This is a specific type of information query, which is also bonded to a specific element from a defined form layer, which can be edited in the database (updated).

### Query Properties

`AttributeName` – query is assigned to an element of the layer only if it contains the attribute and its value is equal to the `AttributeValue` (optional)

`LayerName` – form layer name, to which is the given information query assigned

`QueryBuf` – maximal number of the results per a single query, default value is 1

`QueryParameters` – parameter definition, correct type definition is important

From properties of the localization queries do localization query type update takes over possibility to turn on the search suggestion (property *DynamicCodeLists*), that will be used with modifiable parameters.

The specific setting of this query are two blocks of SQL commands, the first of which (`InitSlqStmtTemplate`) defines editable element with a list of columns that will be displayed and the second (`UpdSlqStmtTemplate`), which perform the database update of editable element. Another property of this query is `RefreshMapAfterUpdateDone`, which, if set to “True” will after finish of the query type update refresh the map widow content.

`InitSlqStmtTemplate` – query parameters can be of different types. It can be a non-editable text parameter, editable text parameter, or items of dynamic or static code lists, which are defined as code lists of localization queries. All the editable items must be named as numeric indexes in ascending order and the parameters must be defined in the property query parameters. The column named differently than the number will not be edited. Property of this item only for reading is given by the order of columns in the select definition.

`UpdSlqStmtTemplate` – SQL command for the database update of a particular graphic element displayed using select from the previous definition. It is very important to use type casting of parameters, if the parameters are not type casted; there is a risk of SQL injection.

## 7.4.5 Document Query

This query is used to select the documents associated with a geometric element table. The query consists of two queries.

- Query, which select overview of the documents.
- Query that select binary data that represents "document".

### Query Properties

`AttributeName` – Query is assigned to an element of the layer only if it contains the attribute and its value is equal to `AttributeValue` (optional)

`LayerName` – form layer name, to which is the given information query assigned

`QueryBuf` – maximal number of the results per a single query, default value is 1

`ResultTemplate` – HTML template of the result (optional)

`SlqStmtTemplate` – SQL query template. Query must contain the parameter `~(long)ID~`, this parameter is before evaluating substituted by the ID value of the graphic element. Each selected column must be named; this name is then displayed as the name of the result table. The query must contain ("return") following columns:

`ID` – a unique document id

`EXTENSION` – extension of the document (e.g. jpg, doc, xls)

`LABEL` – document name

BIRTH\_DATE – time of creation or last modification of the document

BINARYQUERY – id of binary query

### Example

```
SELECT id ID, blob_type EXTENSION, NOTE LABEL,
DATE_OF_INSERTION BIRTH_DATE, 10004 BINARYQUERY from testdocumenttable
```

## 7.4.6 Auxiliary Query for Selection of Document from DB

### Query Properties

LayerName – form layer name, to which is the given information query assigned

QueryBuf – maximal number of the results per a query, default value is 1

ResultTemplate – HTML template of the result (optional)

SlqStmtTemplate – SQL query template. Query must contain the parameter ~(long)ID~, this parameter is before evaluating substituted by the ID value of the graphic element. Each selected column must be named; this name is then displayed as the name of the result table. The query must include ("return") following columns:

DOCUMENT – binary data

EXTENSION – extension of document (e.g. jpg, doc, xls)

BIRTH\_DATE – time of creation or time of last modification of the document

### Example

```
SELECT
DOCUMENT DOCUMENT, BLOB_TYPE EXTENSION, DATE_OF_INSERTION BIRTH_DATE
FROM
testdocumenttable where id=~(long)ID~
```

## 7.4.7 Localization Query

It is a query for localization of an element or territory. Localization query result is an area (limiting rectangle), in which will after evaluation of localization query move map window and form layer displaying localized object or group of objects, which are displayed as a localization result.

A query can include parameters ~i~ ... ~j~, where i ... j are indexes of the order of parameters listed in column QUERY\_PARAMS.

### Query Properties

LayerName – form layer name, to which is the given information query assigned.

QueryBuf – maximal number of the results per a query, default value is 1

QueryParameters – parameter definition

SlqStmtTemplate – SQL query template. The query must contain ("return") following columns:

XMIN, XMAX, YMIN, YMAX – limiting rectangle (RANGE)

ID – of the element which should be highlighted after localization (IF ID is NULL or it is not a number, is located only RANGE)

LABEL – description of the localized element

Symbology – definition of graphical symbology of localized element

### Example

```
SELECT * FROM (
select a.xmin-60000 XMIN, a.ymin-40000 YMIN , a.xmax+60000
XMAX, a.ymax+40000 YMAX, id ID, a.text LABEL
from CM_OB_PARC a, o_cad_area b
where
```

```
upper(b.name_cad_area)=upper('~1~') and
(a.object=b.codca||'1'||cm.get_parcc('~2~') OR
a.object=b.codca||'2'||cm.get_parcc('~2~'))
WHERE 1=1
```

Symbology – definition of graphic symbology of localized element

## HTML publication

AllowLocalizeService – publish a query using localization server

DynamicCodeLists – all lists of values of localization query are "Search suggestions"

GenerateInfo – permission to generate Info icons in the web publication, in the case that for this form layer will be defined at least one information query

HTMLVisible – visibility of information query in a web publication

CheckPubLayers – a list of publish layers, which will become active after processing a localization query

CheckPubLayersStyle – AddToCurrent – original active layers will continue to be displayed, the list of publish layers cited in the preceding paragraph will be added to active layers

ReplaceCurrent – original active layers won't be displayed, only the list of publish layers defined in the previous paragraph will be displayed

## 7.4.8 Localize Group

The functionality of this query is practically the same as in previous case with the only difference that it locates and displays the entire group of elements. Another difference in the query definition is in the mandatory columns of the query, which must contain all the system table columns - ID, XMIN, XMAX, YMIN, YMAX, SES\_ID, TYP\_GEOM, GEOM, SPAT\_KEY, in case of WKB database data store. Everything else related was described in the chapter: [Localization query](#).

## 7.4.9 A List of Offered Values (Code List)

An auxiliary query for evaluation of offered values in the localization query parameter.

Evaluation of code list of localization query may depend on the evaluation of the preceding associated parameters of the localization query.

E.g. localization query that locate streets will have two parameters, municipality and street. For each parameter, there will be a code list. The first one is for selection of municipality A, the second one is for selection of street B. A selects all municipalities and its value will determine the selection of streets in this municipality. First is evaluated the codebook A, its current value is then substituted into the code list B. B therefore depends on A. Before evaluating the query, all the parameters that were already filled are substituted.

A query can contain parameters ~i~ ... ~j~, where i ... j are indexes of order of parameters listed in column QUERY\_PARAMS of associated localization query.

A query returns a single column, which is a list of values.

## Query properties

Dependency – code lists dependency

LayerName – form layer name, to which is the given information query assigned

QueryBuf – maximal number of the results per query, default value is 1

QueryLV – id of associated localization query

QueryLVNUM – index of parameter of localization query

SqlStmtTemplate – SQL query template

## 7.4.10 A List of Static Values

Auxiliary query for evaluation of offered values in the parameter of localization query without database query.

## Query Properties

LayerName – form layer name, to which is the given information query assigned  
 QueryBuf – maximal number of the results per query, the default value is 1  
 ListOfValues – list of static values  
 QueryLV – id of associated localization query  
 QueryLVNUM – index of parameter of localization query

### 7.4.11 Redirection

It is a specific type of information query that allow redirecting to a web page with an arbitrary parameter.

SQL query template (`SlqStmtTemplate`) must contain the column URLADDRESS and parameters, that will be added according to the selected element. Query must include parameter `~(long)ID~`, this parameter is substituted before the evaluation by the value ID of graphical element.

#### Query properties

AttributeName – query is assigned to an element of the layer only if it contains the attribute and its value is equal to AttributeValue (optional)  
 LayerName – form layer name, to which is the given information query assigned  
 QueryBuf – maximal number of the results per query, default value is 1  
 QueryParameters – definition of the parameters, which are used as parameters for searching in SQL query  
 ResultTemplate – HTML template of the result (optional)  
 SlqStmtTemplate – SQL query template

#### Displaying of the Result

ResultHeight – height of information bubbles, resp. window (in pixels)  
 ResultWidth – width of information bubble, resp. window (in pixels)  
 TableStyle – type of displaying query in client  
 TargetName – name of the html window  
 ViewStyle – type of displaying query in client
 

- InPanel – information tab on the right side of the web publication
- InPopUpBubble – in a "bubble" near the element
- InNewWindow – opens a new window and the result can be displayed as any website

 WindowLeft – left indent while opening the new window (in pixels)  
 WindowOrigin – the beginning of the window for calculation of the position  
 WindowTop – above Indentation while opening a new window (in pixels)

#### Example

```

select
  'http://cs.wikipedia.org/w/index.php' URLADDRESS,
  CITY_NAME "search"
FROM
  U_ALL_CITIES
WHERE id=~(long) ID~
  
```

This SQL template opens the website <http://cs.wikipedia.org> with parameter "CITY\_NAME". The string will be searched on this server, usually about the selected municipality, for example: <http://cs.wikipedia.org/w/index.php?search=PRAHA>.

### 7.4.12 Draw by Etalon

This is a type of query, allowing drawing elements from graphic symbology, which is defined for a given element in the table of etalon (default table GSFW\_ETALON) in the relevant structure, which is described in the chapter [Etalon library](#). In the query properties is possible to define parameters, which will be saved into the data store along with graphical information of the drawn element.

#### Query properties

`QueryParameters` – definition of parameters

`SlqStmtTemplate` – SQL query template

### 7.4.13 Delete Element

It is a type of query that allows physical removal of the element from the data store. The query structure is identical with standard information query, except that the template of SQL query (*SlqStmtTemplate*) contains the keyword DELETE, which takes care of deleting the selected element. The query must include parameter ~(long)ID~, this parameter is substituted before evaluation by the value ID of the graphical element.

### 7.4.14 File Browser

This is a type of query that allows management of documents in data stores. It allows viewing, inserting and deleting. This query has a specific category of settings.

File browser setting	
CanDelete	True
CanInsertNew	True
SlqDeleteDocumentTemplate	DELETE FROM <>TABLE </i>W
SlqDocListTemplate	SELECT <>d</i> ID, <>blob_ty
SlqInsertDocumentTemplate	INSERT INTO <>TABLE </i>(BL

`CanDelete` – parameter that defines whether the document from this query may be deleted

`CanInsertNew` – parameter that defines whether a document from this query may be inserted

`SlqDeleteDocumentTemplate` – delete database parameter allowing deleting the document from the database

`SlqDocListTemplate` – initialization select returning a list of documents

`SlqInsertDocumentTemplate` – database construction insert allowing insertion of a new document into the database.

### 7.4.15 Draw and Locate WKT String

In the web publication is this query displayed in a tab *Search*. This query parameter is a string defining the vector geometry of element. Definition of displayed and localized element is in the following form:

TypeOfElement(x y,x y...) where TypeOfElement may be: POINT, LINESTRING, POLYGON, and GEOMETRYCOLLECTION. Entry of coordinates must be in the format of the source cartographic projection of data store, in which is the query defined. X-coordinate is from coordinate Y separated by a space. All coordinates (other than point) are separated by commas. Element type *GeometryCollection* allows rendering and localization of several different objects.

### 7.4.16 Multi Select

This type of query is used for multi selection and subsequent recall of individual query above selected elements. This query must be an application (in a web presentation will be displayed only as a button).

The setting options of multi select query:



<input checked="" type="checkbox"/> <b>5 Application</b>	
ApplicationButton	
IsAppliacion	<b>True</b>
<input checked="" type="checkbox"/> <b>Selection settings</b>	
HighlightFormLayer	
LabelAttributeName	
MaxSelectedElements	5000
Query	
SelectedSaveFileName	<b>SelectedElements.txt</b>
SelectFormLayer	
SelectionString	<b>element/element0.</b>
SelectTolerance	<b>1</b>
ShowSelectedLabels	<b>False</b>
ShowSelectedLabelsSaveButton	<b>False</b>

## Selection settings

The first step is to set the query as an application (*IsApplication = True*). To that application button can be assigned its own icon (item *ApplicationButton*).

*HighlightFormLayer* – the form layer name is used to highlight the elements.

*LabelAttributeName* – Name of the attributes of the selected elements, which will be used as a text description of the selected elements in visual component of html client. This attribute must be defined in **SelectFormLayer** and *DBCColumnsToClient*. If not specified, the ID of the element will be used.

*MaxSelectedElements* – the maximum number of selected elements

*Query* – Information query that is used for evaluation of the selected ID. The resulting collection of ID is in the form ID1, ID2, ID3 and information query should be in the form "... where id in (~ (idlist) ID ~)".

*SelectedSaveFileName* – In case, that the button for saving the description of the selected elements is displayed, this text will be used as a default file name.

*SelectFormLayer* – form layer that will be used for the definition of selected elements

*SelectionString* – text in status bar of the component

*SelectTolerance* – tolerance in the source units is used to enlarge the restrictive selection rectangle (necessary for the selection of elements of type TEXT).

*ShowSelectedLabels* – displays components for a list of descriptions of selected elements

*ShowSelectedLabelsSaveButton* – displays buttons to save descriptions of selected elements

## 7.4.17 Localize Target Coordinates

This is a simple localization query, which allows localization of the target coordinates. In the specific properties of this query is just following

<input checked="" type="checkbox"/> <b>Misc</b>	
LabelXcoordinate	<b>X</b>
LabelYcoordinate	<b>Y</b>
Precision	<b>0</b>
SwapCoordinate	<b>False</b>

*LabelXcoordinate* – description for the X-coordinate displaying in the web publication

*LabelYcoordinate* – description for the Y-coordinate displaying in the web publication

*Precision* – accuracy of coordinates

*SwapCoordinate* – allows swapping of coordinates X a Y

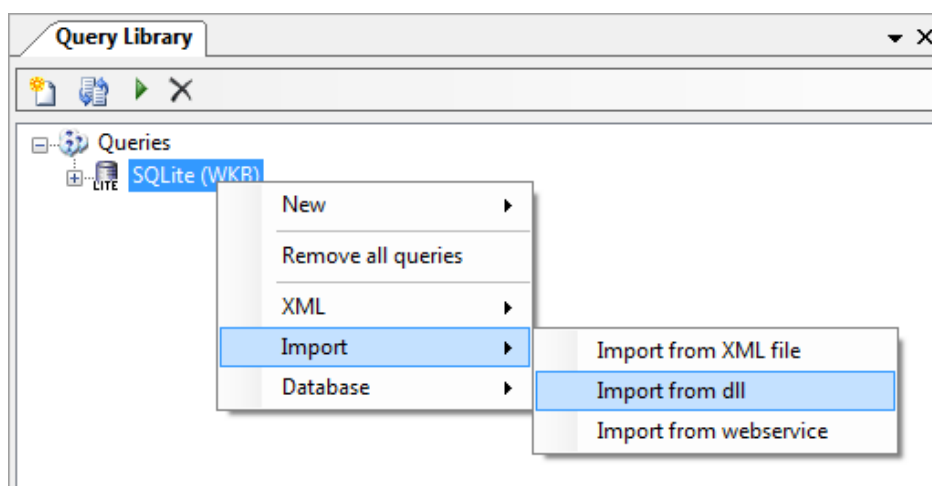
## Query properties

*Range* – sets tolerance of limiting rectangle in source units.

The query result will be then displayed in the web publication in the tab "Search".External Queries

## 7.4.18 External Queries

The core of the system GSFrameWork is fully opened for development of other query types. It is possible to use either database connections and also a wide set of geometric/topologic functions, for example graph tracing, spatial analysis type buffer zone, intersections and generally any set operations. The only condition during such a development is implementation of classes that are inherited from the base objects of the system core. The result of this development is ad hoc a DLL library, containing a set of query tools, which will be imported into the specific project. Any query can be published through the web service and allow its standard protocol SOAP calling.



## 7.5 Cell Library

It includes several groups of graphical elements that have different uses in the project. The bases of this library are vector cells *WKBGeometryCollection* whose definition can be stored in the database table (for database sources is implicitly set to GS\_CELL). Cell library can be further edited and then saved primarily into the configuration xml file of the given project, in which is stored the definition of all cells – including those that were not imported into the project from the database tables. The second basic type of cell is a raster cell that can be thought of as the image of a small format.

In case of using the database tables for the cell library, it is necessary that the table contains, in addition to the standard columns for graphical tables, also the following columns:

CELLNAME	The internal name used in the geometric elements
DESCRIPTION	Description of cell using tilde notation
CELLTYPE	Symbol type
CAPTION	Text in the legend in the web publication

The figure below shows the sample preview of cell library dialog box, where are displayed individual cells. Again, there was used a tilde notation and a tree structure for better clarity and orientation in this library. In the lower part of a dialog box is displayed preview of a selected cell with highlighted cell origin, under the preview is displayed graphical scalebar, symbolizing the cell size. Preview background color can be changed using the buttons on the left side of the dialog box.

By right mouse click in dialog box of Cell library you invoke the context menu of Cell library, in which you can perform editing of elements. The first line of the context menu is used to create a new cell. Vector cell can either be created from selected elements in the map window or from the elements of selected wkb file. Raster cell can be created either from a prepared file that contain raster information or from the selected vector cell. Selected cells can be removed, can be exported into xml file, or stored in the database. But in the databases are applied various restrictions, it e.g. cannot store raster cell. In the edit item of this context menu, it is possible to change the selected cell. You can change the origin of cells on one of nine points, you can replace the color of one cell with another (only for vector cells), you can

increase (or decrease, if you select magnification parameter higher than zero and less than one), you can set the default MBR, you can change the image used for raster cell.

In the right side of the picture shown below are displayed properties of an item from a cell library:

**Caption** – If it is a cell designed for a legend, the text in this field will be displayed as a description of this legend

**CellName** – Cell name that must be unique within this data store

**CellType** – Symbol type (Generic – symbol used for drawing, Infolcon – symbol used for the active information icon of the information queries in the web publication, CaptionItem – legend item in a web publication)

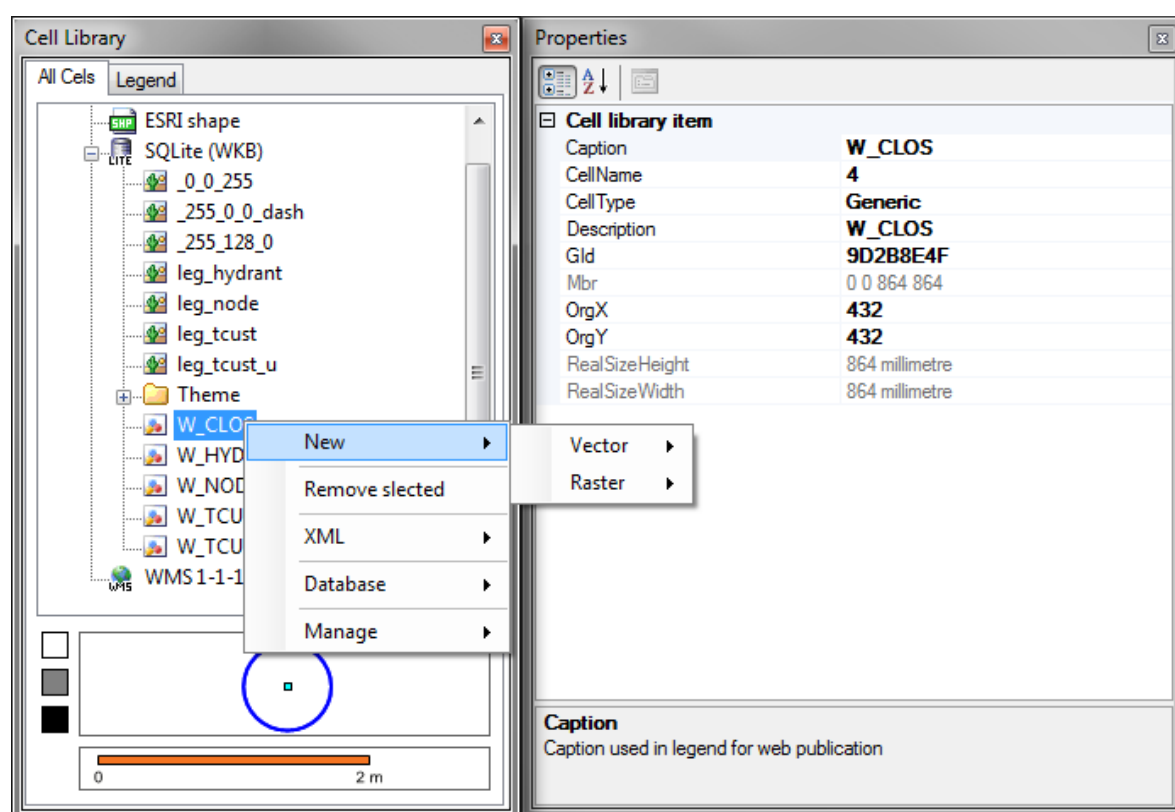
**Description** – description of the cell using tilde notation, thus the cell will be displayed in the tree window in Cells library

**GId** – the unique identification number of an object

**Mbr** – limiting rectangle of cell

**OrgX** – X – coordinates of reference point

**OrgY** – Y – coordinates of reference point



### 7.5.1 Legend

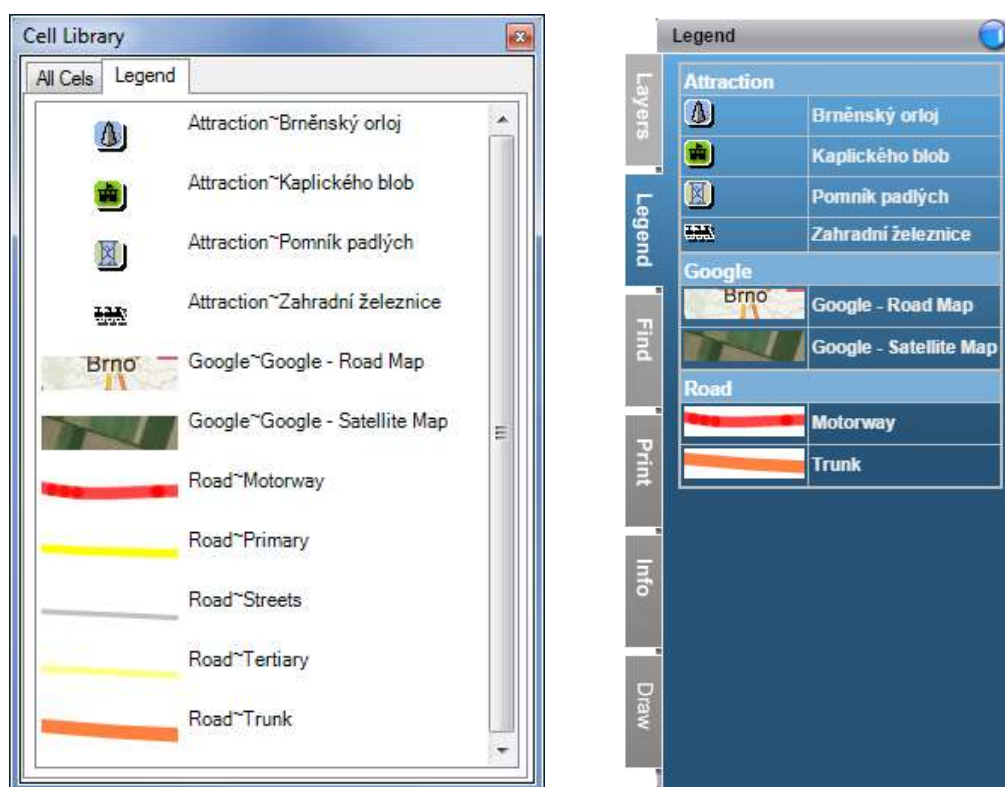
This part of a dialog box serves to preview the individual cells of the legend and for user classification. By default, the legend is sorted by the value in the parameter *Caption*. The sorting method of the legend can be changed in the properties of the data source. The legend can be also sorted by parameter *CellName* or by parameter *GId*. Option *Custom*, which is also available in this menu, will be selected automatically after the user intervention into legend sorting in the cell library. Sorting is performed by holding the left mouse button and dragging the element into the desired position. Legend items can be also selected by multi selection of cells (by selecting the group of cells using CTRL and left mouse button or SHIFT and left mouse button), then by holding the mouse button and dragging the items into desired position.

## Use of the Legend in Project

The legend in the project has two options of generating. The first one is a static, which will be displayed within form layer. Individual elements available for particular form layer are defined in properties of form layer in category 5. *Publish properties – Legend items*. When selecting this type of displaying, the legend will be displayed with all its items when this layer is activated.

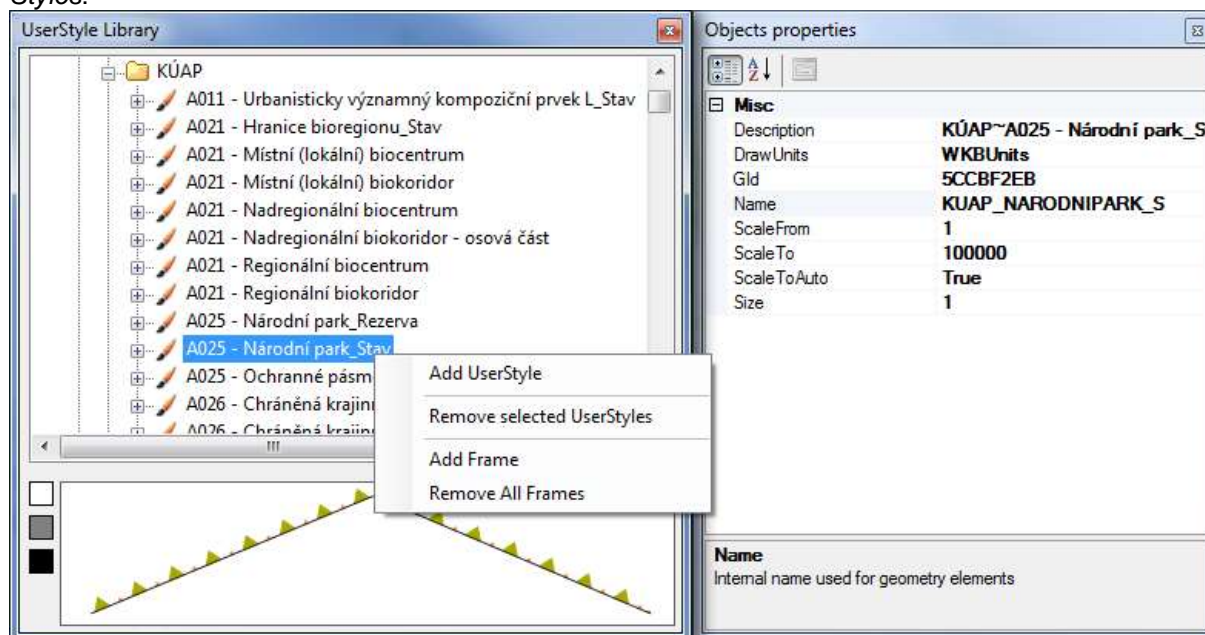
The second way to generate the legend is by using pseudo column SET\_LEG\_ITEM, which will be defined in the properties of form layer in category 4. *Database layer properties – DBColumnsToClient*. More on definition of dynamic columns and pseudo columns is mentioned in chapter [Dynamic columns and columns SET\\_PARS](#). So defined legend is displayed only in the case when at least one rendered element is meeting the criteria of legend item and it will have a penetration of its limiting rectangle with the currently displayed map window.

Suggestion: Legend can be categorized by inserting string into field *Caption* before the actual description of the legend item. It is then necessary to separate this string by the character ~ (tilde), as in the left figure below. It is also necessary to have a legend sort set by the *Caption* parameter. Attention, this procedure cannot be used, when there are already used tilde notations in the form layers. Legend will after this procedure sort similarly as in the right figure below.



## 7.6 User Style Library

The User style library is controlled by editor, which is activated in the menu *Tools – Libraries – User Styles*:



The User style library is a part of the data source configuration (is saved into XML project file and is reloaded each time the project is opened).

Each uses style in a list of styles can contain 1 – n user style frames, which is a custom cell style.

User style library includes these features: Add new User style, Remove selected User style(s), Add new user frame and Remove all frames.

After selecting specific items from a User style library are available its properties, as shown in the right part of the picture:

Cell – geometry of the cell itself, the cell name of the library is defined in the following line

CellName – name of cell

Descr – description using tilde notation

DrawOrder – order of rendering element in the user style, within which it should be unique

FrameType – type of a pattern that is used for distinguishing whether it is a cell or a guiding line

Name – unique name of user style

ParentDataStore – name of the parent data source

As in product GS V6 are the User styles primarily stored in tables or in configuration file of the project.

### GS\_STYLE

ID	primary key of style
NAME	name
DESCR	description

### GS\_STYLE\_DEF

CELLNAME	cell name
FRAMETYPE	pattern type
TACT	retry interval
STYLE_ID	foreign key to GS_STYLE(ID)
DRAW_ORDER	draw order

In MarushkaDesign environment it is not necessary to save styles into the database. Styles and cell libraries can be stored in the project (XML). In GSFramework is a Cell library and User style library of properties of data store (DB, connection, DGN file, etc.).

### 7.6.1 Convention in Takeover Symbology

In the following text, if the number of style starts with symbol **sy**, then this symbol can be of values:

- 0 – resulting element takes over the color from the original element
- 1 – resulting element preserves symbology of sample cell
- 2 – resulting element takes over the color and thickness of the main element

for example **sy10**:

- 10 - takes over the color from the original element
- 110 - preserves symbology of a sample cell
- 210 - takes over the color and thickness of the original element

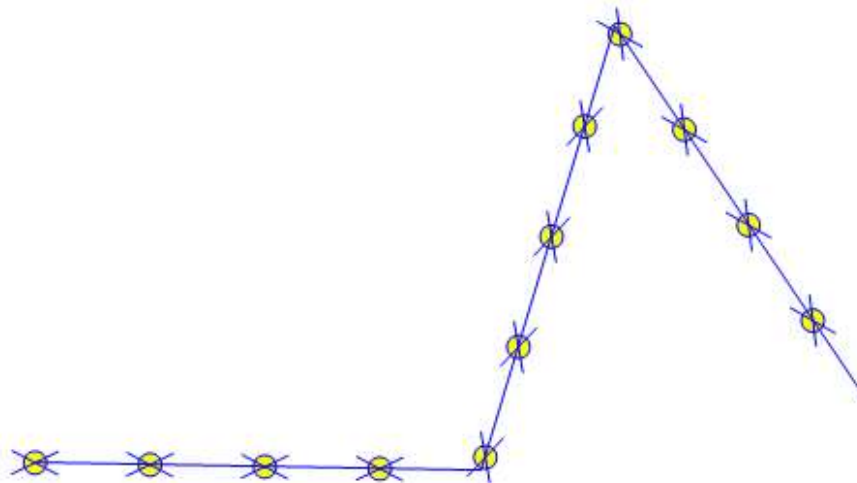
### 7.6.2 Styles **sy10** and **sy17**

Patterning line / border of polygon in a distance specified by a column TACT. Styles **sy10** starts with cell, **sy17** starts with a space.

Mandatory columns in the table **GS\_STYLE\_DEF**:

CELLNAME, FRAMETYPE, TACT, STYLE\_ID, DRAW\_ORDER

Example:



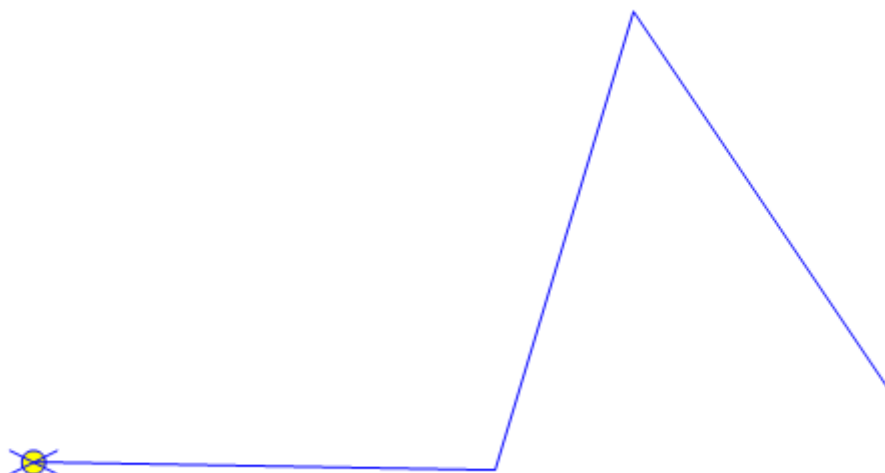
### 7.6.3 Styles **sy11** and **sy12**

Patterning of start (**sy11**) and end (**sy12**) of line / border of polygon by a cell.

Mandatory columns in a table **GS\_STYLE\_DEF**:

CELLNAME, FRAMETYPE, STYLE\_ID, DRAW\_ORDER

Example:



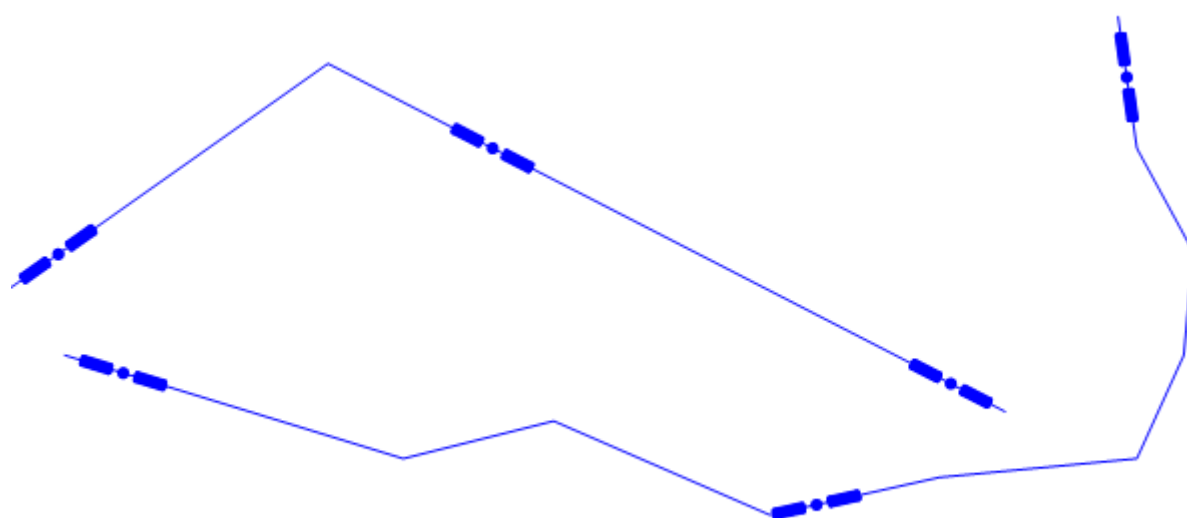
#### 7.6.4 Styles **sy21**, **sy22** and **sy23**

Patterning of start (**sy21**), end (**sy22**) and a middle part (**sy23**) of line / border of polygon by a cell. Unlike styles 11 and 12 is a cell touching by a left / right edge the start / end of the line. The initial space is defined by property **TACT**. The cell is placed only if there is at least one segment of a polyline of a size larger than a size of cells that are placed in the X-axis.

Mandatory columns in a table **GS\_STYLE\_DEF**:

**CELLNAME**, **FRAMETYPE**, **TACT**, **STYLE\_ID**, **DRAW\_ORDER**

Example (combination of styles 21, 22, 23 and 1000):



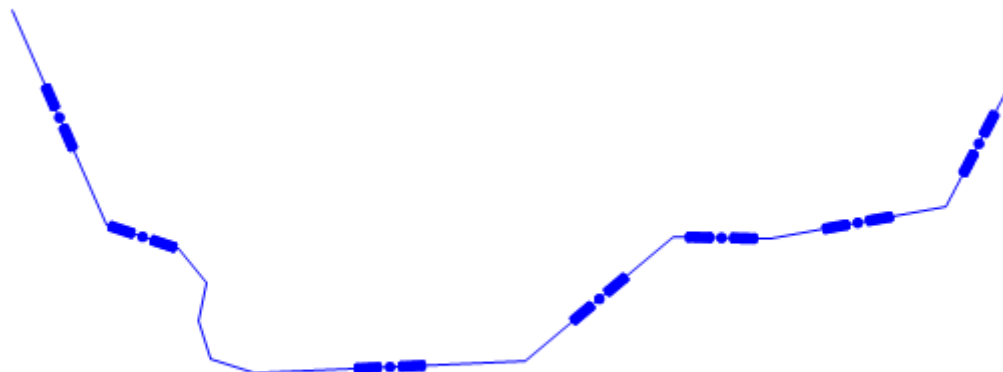


### 7.6.5 Style sy24

Patterning of origins and centers of all the abscissas of line / boundary of polygon by a cell. The cell is placed only if there is at least one segment of a polyline of a size larger than a size of cells that are placed in the X-axis.

Mandatory columns in a table **GS\_STYLE\_DEF**:

CELLNAME, FRAMETYPE, STYLE\_ID, DRAW\_ORDER



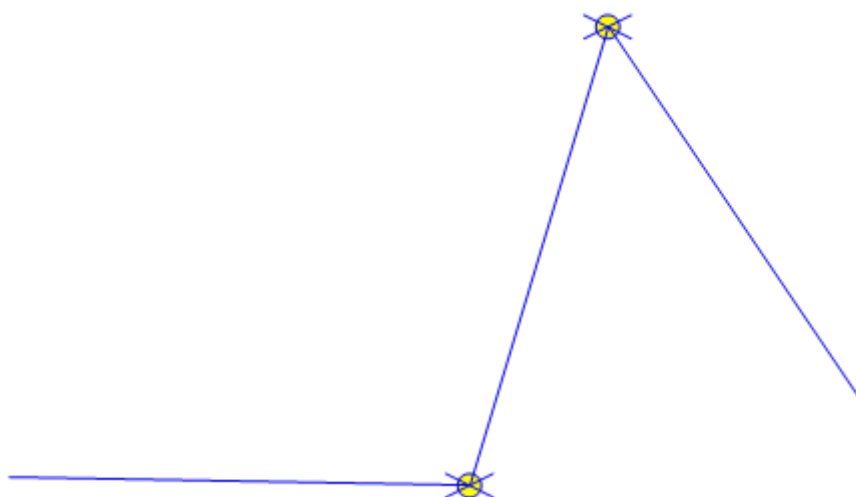
### 7.6.6 Style sy13

Patterning of internal vertices of line / boundary of polygon by a cell..

Mandatory columns in a table **GS\_STYLE\_DEF**:

CELLNAME, FRAMETYPE, STYLE\_ID, DRAW\_ORDER

Example:



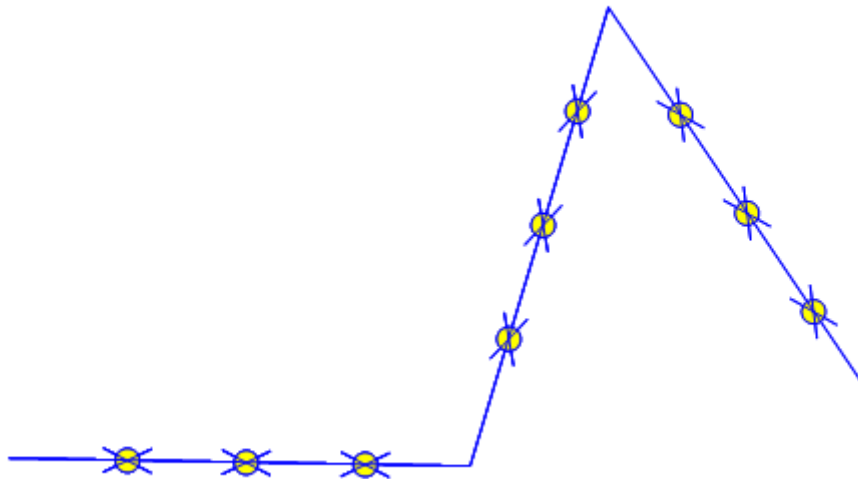
### 7.6.7 Styles sy14 and sy15

Patterning line / boundary of polygon by a cell in a distance specified by a column TACT, omitting the space around vertices. Styles **sy14** start with a cell and **sy15** start with a space.

Mandatory columns in a table **GS\_STYLE\_DEF**:

CELLNAME, FRAMETYPE, TACT, STYLE\_ID, DRAW\_ORDER

Example:



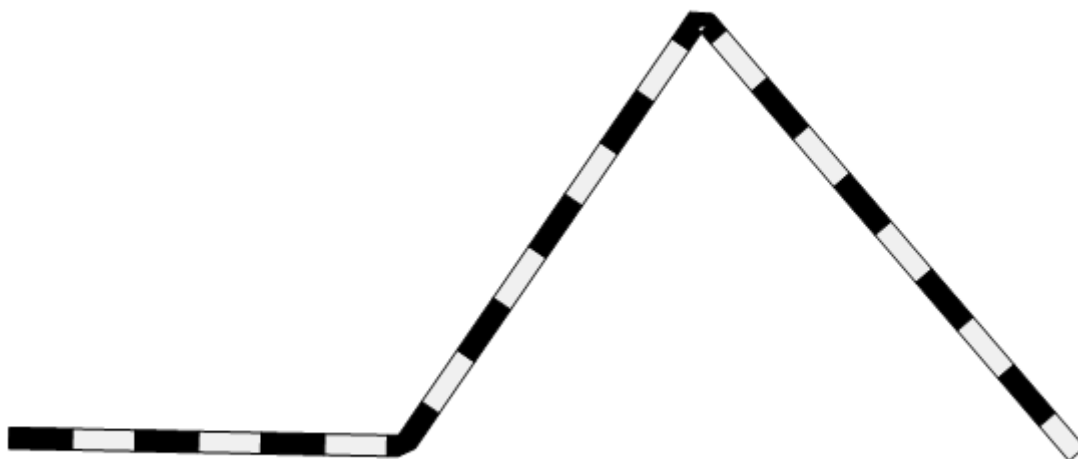
### 7.6.8 Style sy16

Continuous patterning of line/boundary of a polygon by a cell, cells are mutually bonded and on vertices they "bend". The bends occur just in a fracture points of a cell; therefore it is advisable to draw the sample cell so that it can bend in the X-axis. In lines parallel to the X-axis, insert a point in a row although it visually looks like a line.

Mandatory lines in a table **GS\_STYLE\_DEF**:

CELLNAME, FRAMETYPE, TACT, STYLE\_ID, DRAW\_ORDER

Example:

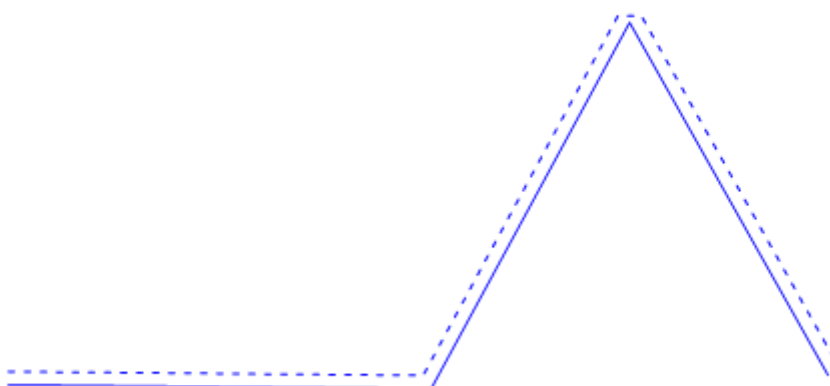


### 7.6.9 Style 10ws

Drawing of a parallel with the original line in a distance TACT with a thickness **w** and basic style **s**. The color was taken from the original line, in case **w=9** and **s=9** is taken the whole symbology.

Mandatory columns in the table **GS\_STYLE\_DEF**:  
FRAMETYPE, TACT, STYLE\_ID, DRAW\_ORDER

Example:

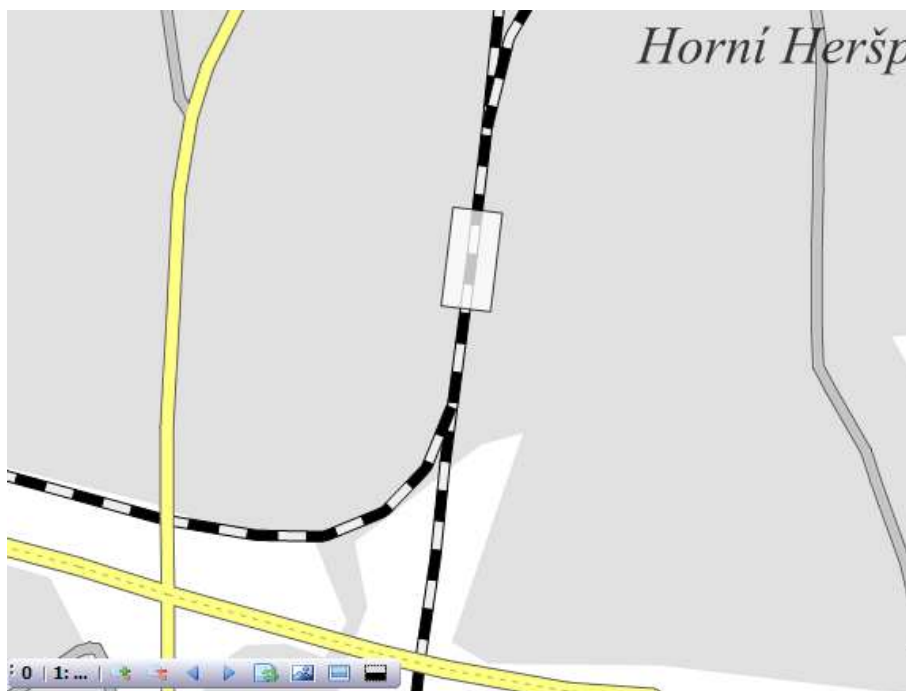
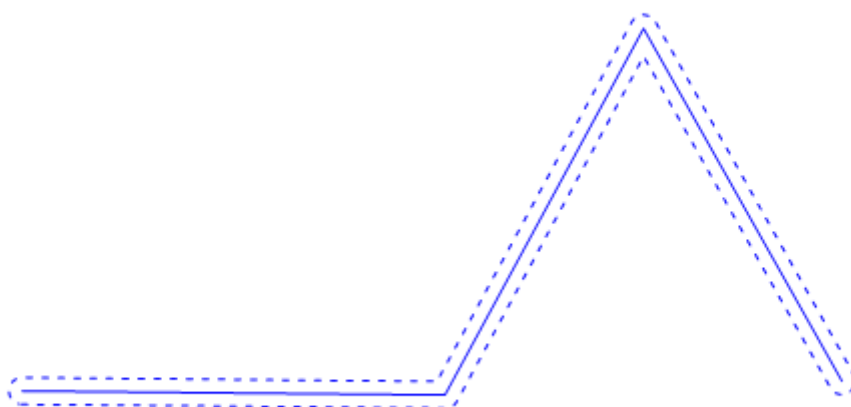


### 7.6.10 Style 11ws

Buffer zone created around the original line in a distance TACT with a thickness **w** and basic style **s**. The color is taken from the original line, in case **w=9** and **s=9** is taken the whole symbology.

Mandatory columns in a table **GS\_STYLE\_DEF**:  
FRAMETYPE, TACT, STYLE\_ID, DRAW\_ORDER

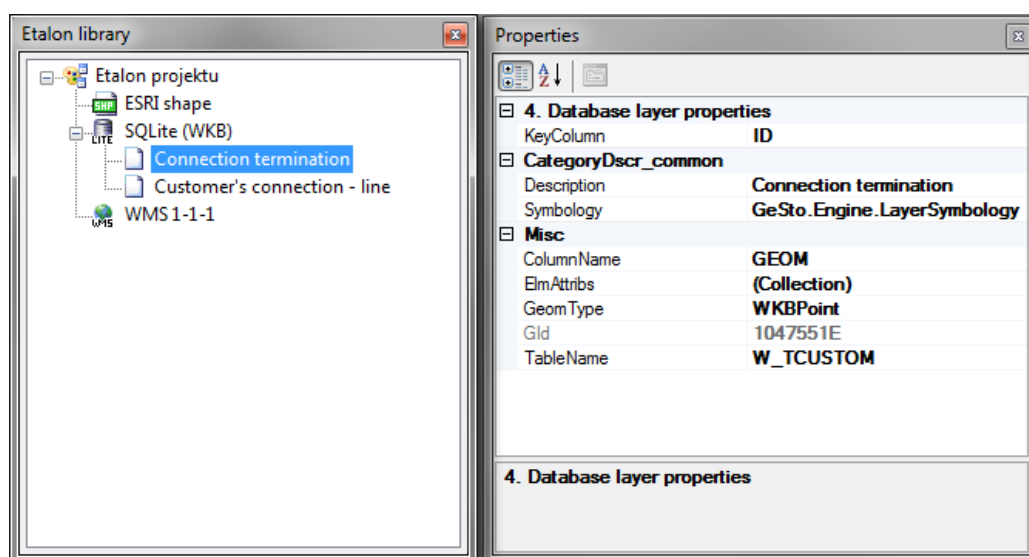
Example:



## 7.7 Etalon Library

Etalon library is a table that provides instructions for drawing for each physical layer. It can also be used to validate the acquired / imported data before storing them in data store. In MarushkaDesign it is an object that belongs to the data store. Etalon may be a part of a project (XML configuration) in case of database data store it may be stored in the relational table.

In MarushkaDesign environment is the following dialog box a foundation for editing this library.



In this sample dialog box, you can see that the etalon library is always assigned to a specific data source. In the context menu will after using the right mouse button appear options to add a new record to the library and for removing the selected record(s) from the library. After selecting the specific item from etalon in the right side will appear a dialog box with its properties, as shown in the right side of the figure:

KeyColumn – a column with a primary or unique key

Description – a description using tilde notation

Symbology – graphical symbology of a layer

ColumnName – a column with geometry

ElmAttribs – mandatory columns of acquired geometrical objects

GeomType – type of drawn WKB object may have one of the following values: Unknown (unknown), WKBPoint (point object, text), WKBLineString (line, polyline), WKBPolygon (polygon), WKBGeometryCollection (cell)

GId – the unique object identifier

TableName – name of the physical layer, in which the object is located

In the event that GSFV\_ETALON will use a database table and it won't be defined in the project, then the founding script of this table should look like this (in SQL syntax for Oracle):

```
CREATE TABLE GSFV_ETALON
(
  ID                NUMBER,
  DESCR             VARCHAR2(80 BYTE) NOT NULL,
  TABLE_NAME       VARCHAR2(64 BYTE),
  ATTRIBS           VARCHAR2(512 BYTE),
  GEOM_TYPES        VARCHAR2(32 BYTE),
  COLOR             VARCHAR2(16 BYTE),
  FCOLOR            VARCHAR2(16 BYTE),
  HCOLOR            VARCHAR2(16 BYTE),
  WEIGHTS           VARCHAR2(16 BYTE),
  STYLES            VARCHAR2(64 BYTE),
  HSTYLES           VARCHAR2(64 BYTE),
  CELL_NAMES        VARCHAR2(64 BYTE),
  DYN_COLUMN        VARCHAR2(64 BYTE),
```

```

TEXT_FONT      VARCHAR2(64 BYTE),
DRAW_PARAMS    VARCHAR2(2000 BYTE)
);

COMMENT ON COLUMN GSFW_ETALON.ID IS
  'Primary key of etalon table';
COMMENT ON COLUMN GSFW_ETALON.DESCR IS
  'Description';
COMMENT ON COLUMN GSFW_ETALON.TABLE_NAME IS
  'Physical table name';
COMMENT ON COLUMN GSFW_ETALON.ATTRIBS IS
  'Mandatory attributes and their defaults (att1=value1,...)';
COMMENT ON COLUMN GSFW_ETALON.GEOM_TYPES IS
  'Comma delimited el-types 1-point, 2-linestring, 3-polygon, 7-collection';
COMMENT ON COLUMN GSFW_ETALON.COLOR IS
  'Comma delimited outline color - A,R,G,B';
COMMENT ON COLUMN GSFW_ETALON.FCOLOR IS
  'Comma delimited fill color - A,R,G,B';
COMMENT ON COLUMN GSFW_ETALON.HCOLOR IS
  'Comma delimited hatch color - A,R,G,B';
COMMENT ON COLUMN GSFW_ETALON.WEIGHTS IS
  'Correct weights, 1st is default for drawing';
COMMENT ON COLUMN GSFW_ETALON.STYLES IS
  'Correct styles, 1st is default for drawing';
COMMENT ON COLUMN GSFW_ETALON.HSTYLES IS
  'Correct hatch styles, 1st is default for drawing';
COMMENT ON COLUMN GSFW_ETALON.CELL_NAMES IS
  'Correct cells, 1st is default for drawing';
COMMENT ON COLUMN GSFW_ETALON.DYN_COLUMN IS
  'Dynamic column name';
COMMENT ON COLUMN GSFW_ETALON.TEXT_FONT IS
  'Correct font (Name=Times New
Roman, Style=Bold|Italic|Regular|Strikeout|Underline)';

CREATE UNIQUE INDEX GSFW_ETALON_UK ON GSFW_ETALON;

ALTER TABLE GSFW_ETALON ADD (CONSTRAINT GSFW_ETALON_PK PRIMARY KEY(ID);

```

Name of column	Meaning
ID	Primary key of the table
DESCR	Tree description (~ convention)
TABLE_NAME	Physical layer (table)
ATTRIBS	Mandatory acquired attributes in the form column = value,...
GEOM_TYPES	Types of allowed geometries (WKB standard)
COLOR	Allowed line color (empty means allowing all colors)
FCOLOR	Allowed fill color (empty means allowing all colors)
HCOLOR	Allowed sample color (empty means allowing all colors)
WEIGHTS	Allowed line thickness
STYLES	Allowed line styles (non-numeric mean user styles)
HSTYLES	Allowed patterns of filling
CELL_NAMES	Allowed cells
DYN_COLUMN	Dynamic column
TEXT_FONT	Allowed text font, in form e.g.: <b>Name=Times New Roman, Style=Bold Italic Regular Strikeout Underline</b>
DRAW_PARAMS	Parameters of drawing, CAD layer, text height..., are offered in MarushkaDesign application, properties of etalon items

## 8 Publish Server Marushka®

Web publication – is the result of configuration of the project in MarushkaDesign interface. It is a basic tool the resulting configuration in Marushka (server) that displays the resulting map composition in the web browser. A web publication consists of the main map client dialog box, among other things it allows to create a variety of print outputs, supports descriptive information and queries, drawing queries and communication with external applications.

Local web server makes the project available to all the Internet users, who know the address. This project can be then through a browser interface and the following sending of http request launched by any user, who knows the exact IP address and password to the project (if it is protected by password). This project can be then using web browser and by subsequent sending http request to run any project connected to internet.

Local web server returns images based on submitted queries. It is possible to change the returned images format in category 5. Publish properties, item *FormLayerFormat* of the given publish layer. Setting the format of the returned image will be reflected on the image quality, range of displayed colors and also the file size. The PNG format is especially suitable for vector drawings because it supports only a narrow range of colors. Its advantage is a small resulting file size. While JPEG is especially suitable for photos, because it supports only a narrow range of colors; its disadvantage is, however, that the resulting file sizes are very large.


In the local web server, the user will only see only the layers that are included in publish layers. In the map publication can be also displayed layers that will be displayed in the map window, but they won't be visible in a list of layers in publication, so the user cannot switch off these layers in the browser. This functionality can be achieved by setting in Category 2. *Marushka HTML publish properties* item *DefaultLayer* to value "True" and item *visible* to "False".

Start limiting rectangle is taken primarily from the definition stored using theme editor for the given theme. If in the theme editor is this value yet not defined, the current window viewport is used.

For proper functionality of the Web server is necessary to set the correct path in the Web.config file, which is discussed in more detail in chapter [8.6 Modifying Web.config configuration file](#).

On one computer can run more MarushkaDesign applications (instances) at once, which is useful for example to compare two projects in different versions. Since version 4.0.0.0 Marushka can also run on the same computer in more than one window of the local web server simultaneously. This is possible, because if one webserver is already running in one webserver window, so the other window changes the address of connecting port. Therefore, they can run simultaneously.

### Window Local Webserver

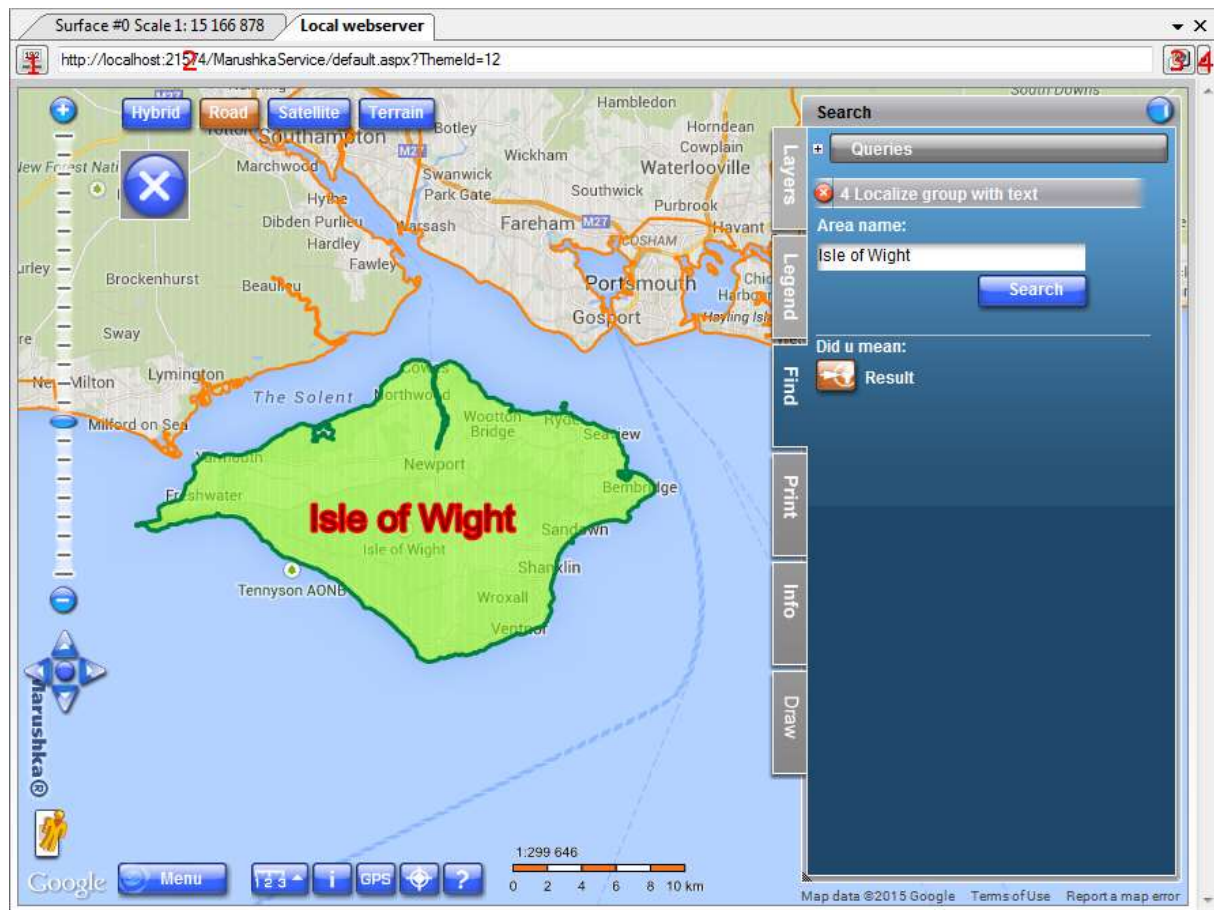
A local webserver is started using the button  in toolbar *Layers and mapsurfaces*. This starts the new dialog box named *Local webserver*. This dialogue box cannot be moved by dragging by the left mouse button or otherwise as the other dialog boxes, because it is firmly docked.

Above the Local webserver dialog box is the **Status bar, showing the server address (2)**. This address can be copied, pasted in the browser window and subsequently sent to any other computer connected to the Internet. This link can be sent to anyone, for example by e-mail, through Skype or through any other communication channel, the user can then display the given project on his computer.

Using button **Select machine address (1)** is possible change the server address, when instead of the name of the local server localhost is displayed specific IP address of the currently used server. If the user has a choice of multiple IP addresses, so after click on this button will automatically offer you all available IP addresses.

Using the button **Launch in default browser (3)** is possible to start the project in the default web browser. If the user want to select different than default browser, click on button **Select a Browser (4)**, to the right to previously mentioned button. After clicking on it, the list of all currently installed web browsers appear, from which the user can choose.





## 8.1 Requirements

- IIS 6 and higher
- .NET 2.0
- Configured alias to the SQL database, in case of using a database data store

## 8.2 Application Pool

It is appropriate for the application Marushka® to have its own application pool. In ISS7 Manager select a node **Application pool** and using the right mouse button select a choice **Add application pool**. In a dialog box fill the name, e.g. "MarushkaServer". Next select the version of .Net Framework to "Net Framework v2.XXX" and as a **Managed pipeline** select **Classic** and confirm the selection.

If you run the server on 64bit machine, select the created application pool and in the right side of the dialog box in section **Actions** select **Adjust the application pool – Advance settings**. In the next dialogue box in section **General** set Allow 32bit application set "True".

## 8.3 Creating an application on IIS 7

First, create a folder for Marushka® application, mostly subfolder of the folder WWWROOT (e.g. "Marushka") and copy an application into that folder.

In ISS manager choose node „**Default Web Site**“, by right mouse button choose an option "Add Application". In the following dialog fill alias such as "Marushka", choose created application pool and set the physical path to the previously created directory.

## 8.4 Setting up Application Directories of Marushka

**Marushka application** directory contains the **bin** directory. It is important that this directory is visible for the user, or for the group, under which run IIS (default is ISS\_IUSER or group IIS\_WPG).

It is also important to set for directory **tmp** directory permissions for entry and deleting for users or group, under which runs IIS (default is IIS\_IUSER or group IIS\_WPG).

## 8.5 Copying the Configuration XML

Configuration XML may contain passwords so it is important not to display this file on the web (preferably stored outside the web folder and keep the rights for reading IIS or copy the file to the **bin** folder and disable browsing.)

## 8.6 Editing the Configuration File Web.config

Fix items of the file **Web.Config**, particularly items:

Path to root directory of installed Marushka server instance.

```
<!-- Path to root dir-->

<add key="PathToRoot" value="c:\inetpub\wwwroot\Marushka\"/>
```

The path to the file used for logging errors. It is recommended to place this file outside of the web directory ISS. It may contain error messages of the database. This directory must have set rights to modify / read / write / display contents of the user under which IIS runs (default is IIS\_USER or group IIS\_WPG).

```
<!-- Path to log file default it is LogFile.txt in temp directory. When
it is set, then directory and file has to have write permission for IIS, when
not set LogoFile in temp root will be use. -->
<add key="PathToLogFile" value="c:\MarushkaTemp\logfile.txt"/>
```

Path to temp directory is used by server for storing the generated files. This directory must have set rights to modify / read / write / display contents of the user under which IIS runs (default is IIS\_USER or group IIS\_WPG).

```
<!-- Path to temp directory to store temporary files. Directory has to
have write, modify, delete permission for IIS when not set tmp dir in root
will be set -->
<add key="PathToTemp" value="c:\MarushkaTemp\"/>
```

Server periodically (every two hours) deletes all generated files in the directory, which is defined by the key **PathToTemp**.

```
<!--Automatic cleaning temp directory -->
<add key="ClearTemp" value="true"/>
```

If tracing of the server is set to TRUE, server writes each event to the file defined by the key **PathToLogFile**. Caution on production servers may TRUE setting generate extremely large files and slow running of the entire server.

```
<!--Trace All -->
<add key="Trace" value="false"/>
```

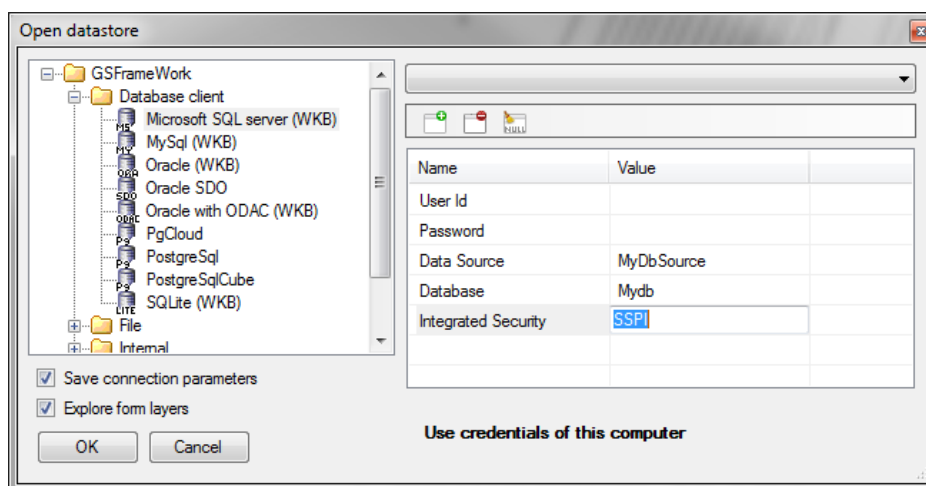
Setting of path to theme collection settings and to the project. This file and the projects to which this file references, is recommended placing outside of the web directory IIS. It may contain definitions of SQL, passwords etc.

```
<!-- Path to theme collection settings -->
<add key="PathToThemeSettings" value=" c:\MarushkaTemp\themes.xml"/>
```

## 8.6.1 Authorization Settings

Map publication server Marushka allows several types of authorization of access to the published project. Authorization type is defined in the configuration server file.

Without



### Authorization

Define public publication (i.e. approaches are not verified) in the settings of the application so that in the Web.config file in the section <appSettings> set the following entry:

```
<!--Flag when authorization is in use-->
<add key="Authorization" value="false"/>
```

Then in the section </system.web> set:

```
<authorization>
  <allow users="*" />
</authorization>
```

### Verifying Access to the Initial Form

A form authentication is based on that user and his password are verified against a list of users defined in Web.config. Set the items in a following way:

Allow the authorization:

```
<!--Flag when authorization is in use-->
<add key="Authorization" value="true"/>
```

Set the authentication type:

```
<!--Type of Authorization component-->
<add key="AuthorizationComponent"
value="FrameWorkServer.Component.BasicFormAuthorization, FrameWorkServer"/>
<!--End Type of Authorization component-->
```

Set the list of users in the section `<authentication mode="Forms">`, e.g.:

```
<authentication mode="Forms">
  <forms loginUrl="login.aspx" name=".ASPXFORMSAUTH" timeout="10">
    <credentials passwordFormat="Clear">
      <user name="geovap" password="test"/>
      <user name="test" password="1234"/>
      <user name="mike" password="test"/>
    </credentials>
  </forms>
</authentication>
```

In section `</system.web>` deny access to unauthorized users:

```
<authorization>
  <deny users="?" />
</authorization>
```

## Windows Verification

Marushka application is verified by the domain identity of Windows user under the same identity, the user connects to database data stores.

## MSSQL

If you require that the application should be verified by NT authentication, you need to connect all the database data stores under the certified identity of Windows system. In the data store connection parameters leave empty parameters "User Id" and "Password" and fill the parameter "Integrated Security = SSPI". User logged into Windows must have set correctly identity on the database server side.

Further, work with the project is exactly the same as work with the standard authentication.

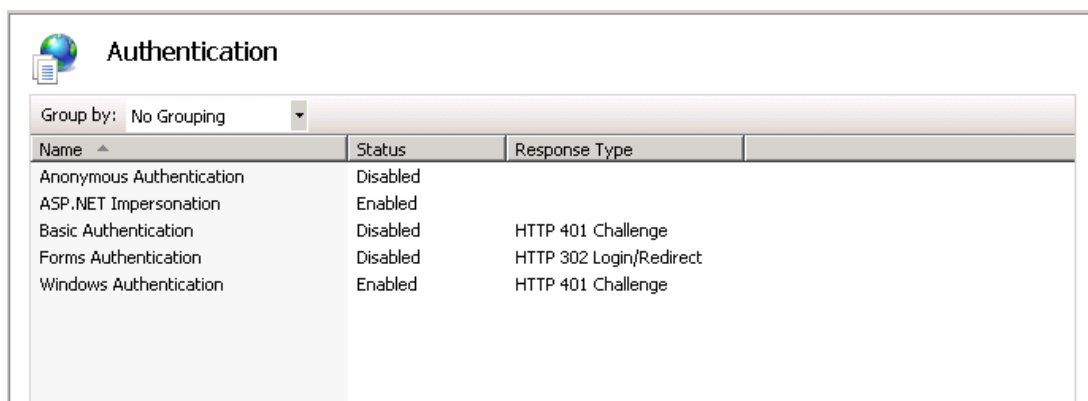
## Setting ASP .NET Application

In IIS Manager select Marushka application instance and move onto the Authentication settings.



In the following authentication card set values:

*Anonymous Authentication=Disabled*  
*ASP.NET Impersonation=Enabled*  
*Basic Authentication=Disabled*  
*Forms Authentication=Disabled*  
*Windows Authentication =Enabled*



### Setting the Web.config file

Setting the primary key to value "true":

```
<!--Web page is (un)authorized-->
<add key="Authorization" value="true" />
```

Setting the authentication component to type BasicFormAuthorization:

```
<!--Basic authorization component (user/password list in this file see <forms
loginUrl="login.aspx...-->
<add key="AuthorizationComponent"
value="FrameWorkServer.Component.BasicFormAuthorization, FrameWorkServer" />
```

In section <system.web> remove authorization module BasicAuth:

```
<httpModules>
    <add name="BasicAuth" type="FrameWorkServer.BasicAuthorizationModule"/>
</httpModules>
```

In the same section replace item:

```
<authentication mode="Forms" ... >
```

with:

```
<authentication mode="Windows" />
```

and complete the identity definition

```
<identity impersonate="true" />
```

delete all the definitions <location path= ...

Next is a complete sample section <system.web>

```
<system.web>
    <httpRuntime executionTimeout="300" maxRequestLength="262144" />
    <webServices>
        <protocols>
            <add name="HttpPost" />
            <add name="HttpGet" />
        </protocols>
    </webServices>

    <!--The <authentication> section enables configuration
         of the security authentication mode used by
         ASP.NET to identify an incoming user.
    -->
```

```
<authentication mode="Windows" />
<identity impersonate="true" />
</system.web>
```

### Verifying by Account on the Database Server

It is the strongest type of authorization, however presupposes the ability to connect the map server to the database server. This type of authentication allows setting the initial rectangles for each user. Items of `web.config` file are set as follows:

Enable authorization:

```
<!--Flag when authorization is in use-->
<add key="Authorization" value="false"/>
```

Select the authorization type:

```
<!--Type of Authorization component-->
<add key="AuthorizationComponent"
value="FrameWorkServer.Component.DataBaseUserFormAuthorization,
FrameWorkServer"/>
```

Define the select command for an initial limiting rectangle for the logged on user, this select command must return columns XMIN,YMIN,XMAX,YMAX and must be feasible in all accounts under which users accesses to the project:

```
<add key="ExtentSelect" value="select XMIN,YMIN,XMAX,YMAX from UAP_UZIVATEL
C, UAP_UZIVATEL_ROLE D, UAP_ROLE_FENCE E, UAP_FENCE F where C.JMENŌ = USER
AND C.UZIVATEL_ID = D.UZIVATEL_ID AND D.ROLE_ID = E.ROLE_ID AND E.FENCE_ID =
F.ID"/>
```

Define alias of the server:

```
<add key="ServerAlias" value="uap"/>
```

Define the type of server from the following options: ORACLE, ORACLEDA, and MSSQL. If you have a project that uses an interface for communication with the Oracle Data Access, it is necessary to define the server type as ORACLEDA. It is impossible to combine approaches into ORACLE databases using various interfaces!

```
<add key="ServerType" value="ORACLE"/>
<!--End Type of Authorization component-->
```

## 8.6.2 Additional Settings

The map project publication can be a part of the portal. In this case, define the address (link) of main menu item as follows.

```
<add key="URLToPortal" value="../Default.aspx"/>
```

## 8.7 External Call of Map Client Marushka

It is possible to call map client from the external web site or from an external application via the HTTP interface using POST/GET. For example using HTML form in HTML web page or in a browser command line (or using TAG `<A href="">`). If you use the GET method so you have to remember that the maximum string length is limited to about 2 000 characters.

By the parameters in the call you can suppress the use of components that are enabled in the client configuration. That is in the theme definition section *LayOut* (definition of distribution of HTML client). If for example parameter *LayOutVicinity* (view overview map) is set to false it is not possible to turn on an overview map from the outside, but if it is enabled, you can disable it by false value.

The names of the parameters applicable to the majority of calls are identical to the names of variables in *Layout* setting. But instead of prefix *LayOut* they contain a prefix *Mar*. For example the parameter for disabling the overview map is *MarVicinity*, in theme editor is named *LayOutVicinity*.

### Parameters Defined in the Theme

Parameter of the call	Variable in the Theme	Description	Values
MarApplication	LayOutApplication	Display application button	true/false
MarCrossBar	LayOutCrossBar	Display control cross for movement in the map window	true/false
MarGeometryQuery	LayOutGeometryQuery	Display geometry queries button	true/false
MarGPS	LayOutGPS	Display GPS button	true/false
MarHTMLFoot	LayOutHTMLFoot	HTML foot file name	Files located in the directory HTML, e.g. foot.dat, to not display an empty string
MarHTMLHead	LayOutHTMLHead	HTML header file name	Files saved in directory HTML, e.g. head.dat, to not display an empty string
MarMeasurement	LayOutMeasurement	Display a button measuring	true/false
MarMenu	LayOutMenu	Display menu	true/false
MarPanelDraw	LayOutPanelDraw	Display bookmark draw	true/false
MarPanelExpand	LayOutPanelExpand	Maximize bookmark panel	true/false
MarPanelFind	LayOutPanelFind	Display bookmark Search	true/false



MarPanelIndex	LayOutPanelIndex	Index of current bookmark	[0,1,2,3,4]
MarPanelInfo	LayOutPanelInfo	Display bookmark information	true/false
MarPanelLayers	LayOutPanelLayers	Display bookmark layers	true/false
MarPanelLegend	LayOutPanelLegend	Display bookmark legend	true/false
MarPanelPrint	LayOutPanelPrint	Display bookmark print	true/false
MarPolygon	LayOutPolygon	Display button selection by polygon	true/false
MarPreselect	LayOutPreselect	Display info tool (function "Click in map -> navigate through selected elements ->and display information")	true/false
MarPreselectOn	-	Activates function info	true/false
MarScalebar	LayOutScalebar	Display a scale slider	true/false
MarTheme	LayOutTheme	Display theme menu	true/false
MarTools	LayOutTools	Display tool menu	true/false
MarVicinity	LayOutVicinity	Display overview map	true/false
MarWheel	LayOutWheel	Zoom using mouse wheel	true/false
MarWindowTitle	LayOutWindowTitle	Title of web browser	string
MarKN	LayOutKN	Display button for access to Land cadaster	true/false
MarPopularQuery	LayOutPopularQuery	Display/hide popular query	true/false
MarPopularQuery	LayOutPopularQuery	Display/hide popular query	true/false

### Layers, Information Layers, Limiting Rectangle and Theme

Parameters for switching on / off layers and info layers and client shift to other areas.

Parameter of Call	Description	Values
ThemeID	Theme ID	[0 .. n]
MarExtent	<p>Parameter defines initial limiting rectangle of HTML client. The values are separated by commas (resp. "%20" because it should be URLEncoded).</p> <p>Actual values are in the coordinate system S-JTSK and mm.</p>	[xmin ymin xmax ymax]
MarMiddlePoint	<p>Defines the center of the client. Values are defined similarly as the parameter MarExtent. With this parameter, it is advisable to call another parameter MarScale.</p>	[x y]
MarScale	<p>Parameter defining the scaling parameter that will be used for displaying in the HTML client. It is recommended to use this parameter along with the parameter MarMiddlePoint.</p>	number
MarUid	<p>List of layers to be turned on after the call. Layers are identified by the global ID of publish layer. Individual values are separated by spaces.</p>	[GID1 GID2 GID3 ...]
MarUidi	<p>List of info layers to be turned on after the call. Layers are identified by the global ID of publish layers. Individual values are separated by spaces.</p>	[GID1 GID2 GID3 ...]

## Authorization

HTML client can be called if authentication is also turned on. It is necessary to really consider this action, because the username and password are the components of default plaintext query.

Parameter of Call	Description	Values
UserName	Username of the user	string
Password	Password	string

## Queries, Localization

HTML client can be also called in such a way that during the first rendering of map evaluates the localization query and shifts starting limiting rectangle to the result of localization. If the result includes controls for highlighting, so it will highlight them.

Parameter of Call	Description	Values
MarQueryID	Unique ID of the query, item GId in the query properties	GID
MarQParamCount	Number of localization query parameters	[0 .. n]
MarQParamX	Query parameter value with index X. Indexes are from 0. The value is entered URLEncoded	string

## Example of Calling the Client with Localization

In the Theme with id=1 you have defined a localization query "*Localization by house number / number of registration, by street*". This query has three parameters (property value QueryParams=Municipality;Street;house number/number of registration). This query has the property GId= 296B89E1. Client is then called with the following parameters:

```
ThemeID=1
MarQueryID = 296B89E1
MarQParamCount =3
MarQParam0= Zlín
MarQParam1= A. Randýskové
MarQParam2= 1679 P Zlín
```

Parameters include Czech characters, so encode them using URLEncoded UTF-8.

```
ThemeID=1
MarQueryID=296B89E1
MarQParamCount=3
MarQParam0=Zl%C3%ADn
MarQParam1=A.%20Rand%C3%BDskov%C3%A9
MarQParam2=1679%20P%20Zl%C3%ADn
```

The resulting query to the server using the GET method is as follows:

<http://adresaserveru/Marushka/default.aspx?ThemeID=1&MarQueryID=296B89E1&MarQParamCount=3&MarQParam0=Zl%C3%ADn&MarQParam1=A.%20Rand%C3%BDskov%C3%A9&MarQParam2=1679%20P%20Zl%C3%ADn>

If you use an HTML form and calling by POST method, encoding is done automatically.

### .Net Application

If you call a HTML client from .Net application you can send a complete form layer to user (object FormLayer from namespace GeSto.Engine) serialized into XML and encrypted by 32bit key.

To create the string you will need a reference to the project GSFrameWork.dll and ExternalServerFunction.dll, then the generated key "ContextKey", which is consistent with the key defined in the server settings of Marushka and specifically in the file "web.config".

```
<!-- Context key-->
<add key ="ContextKey" value="HewSErTSHsXs=" />
```

You get the required string from the object instance using the following method:

```
public static string CreateFormLayerString(FormLayer fLayer, string
contextKey)
{
    string result = string.Empty;
    try
    {
        result = NameConvertor.Encrypt(contextKey,
        FormLayer.XMLStringFromFormLayer(fLayer));
    }
    catch
    {
    }
    return result;
}
```

The resulting string is then sent in the parameter **LocalizeName**, and then you should send the corresponding global ID of data store, thus parameter **LocalizeId**.

A new Contextkey is generated using MarushkaKeyGenerator application.

Parameter of Call	Description	Values
MarLocalizeName	Encrypted and serialized instance of object FormLayer of global ID data store	string
MarLocalizeDsid	Global ID of data store (property GId)	string

### Special Client Modes:

**GetFence mode**

The client can be called in special mode GetFence that is used to enter the polygon (fence). After the call, the client switches to mode "entering fence" and if user successfully completes (enters) fence (polygon), the client returns the coordinates of the specified URL. The returned coordinates of fence / polygon are in coordinate system S-JTSK and in millimeters.

Parameter of Call	Description	Values
action	Action type	getfence
returnFenceURL	URL of pages for processing	string
MarFenceLimit	Maximum fence area	string

**Request:**

<http://localhost:4361/WebServer/default.aspx?action=getfence&returnFenceURL=zpracuj.aspx>

**Response:**

http://localhost:4361/WebServer/zpracuj.aspx?fenceX=-523487871 -519173780 -519576966 -522469826&fenceY=-1165647451 -1164125423 -1166605018 -522469826

## Annex – UML diagram of objects GSFrameWork

