

# Drawing Elements in Marushka Design



**GEOVAP**

## CONTENTS

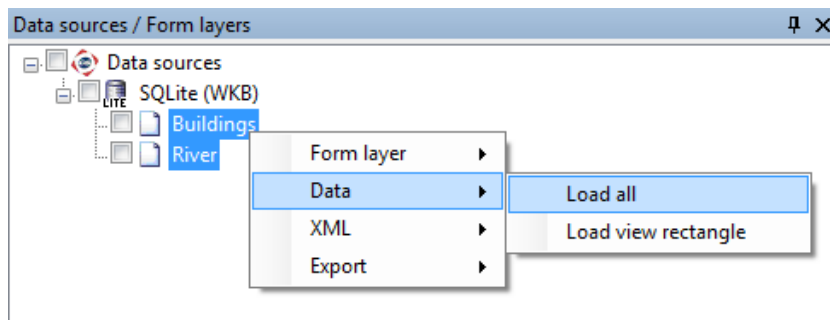
1	AIM OF THE EXAMPLE.....	2
2	WORKING WITH EXAMPLE .....	2
3	DIALOG BOX SAMPLE.....	3
4	A BRIEF DESCRIPTION OF THE EXAMPLE IN MARUSHKADESIGN .....	5

## 1 Aim of the Example

In this example we will demonstrate drawing elements and external call in MarushkaDesign. This example was created in version 4.0.1.0, so it does not have to be compatible with older versions.

## 2 Working with Example

- Unzip the contents of **DrawElement\_EN.zip** into **c:\MarushkaExamples\** folder. The target folder must be respected due to interconnection of paths with the project. In the case of placing the files in the different folder, it would not be possible to work with an example.
- Open the **DrawElement\_EN.xml** in MarushkaDesign environment.
- Select form layer **Buildings** and **River**, in context menu choose Data – Load all:



- Zoom in on the map elements, so that all the elements are approximately in the center of the map window, in some reasonable scale, such as in Fig 1.
- In map window choose: "Fit all":



- Launch the local web server:



### 3 Dialog Box Sample

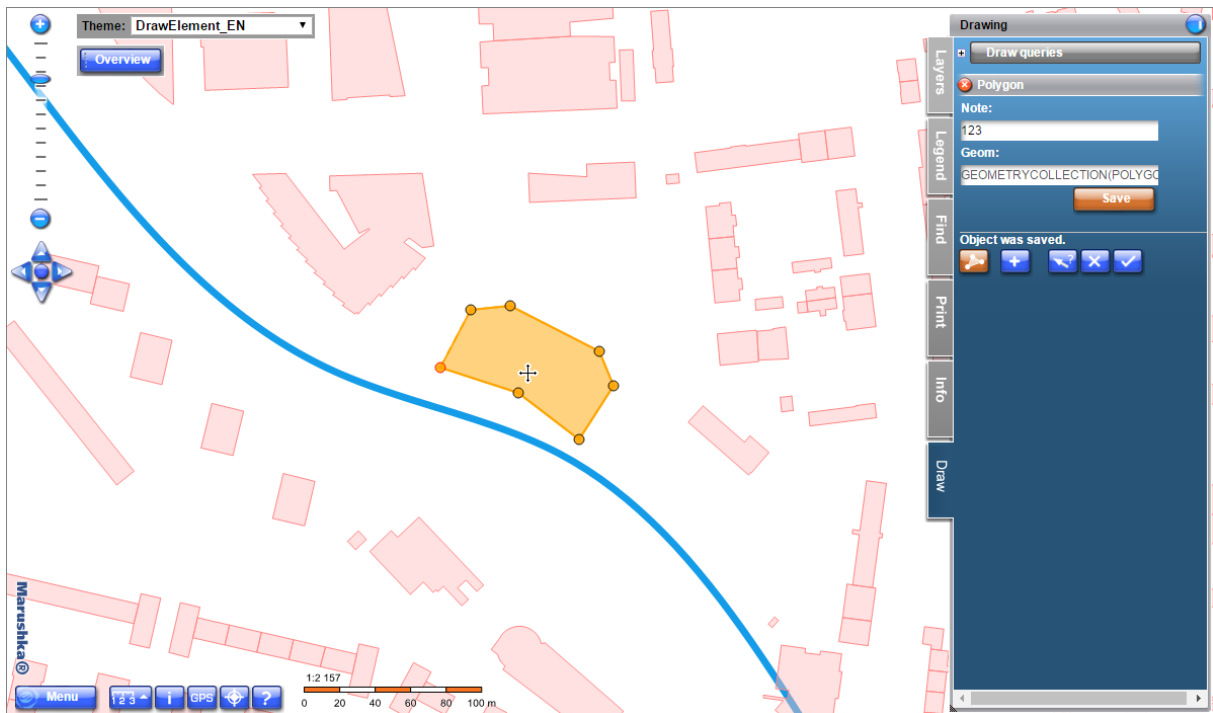


Fig 1: Example of drawing user defined polygon in panel Draw

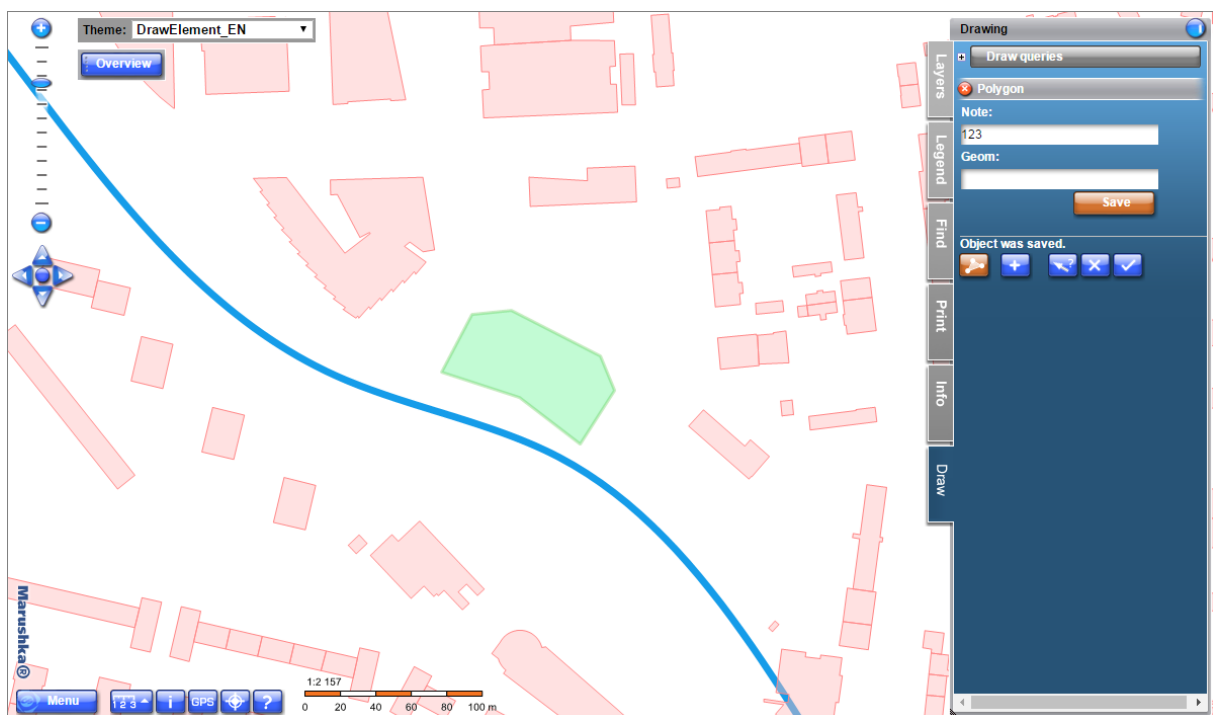


Fig 2: Polygon or any other element will be saved into the database by its definition in etalon library (here it is light green color)

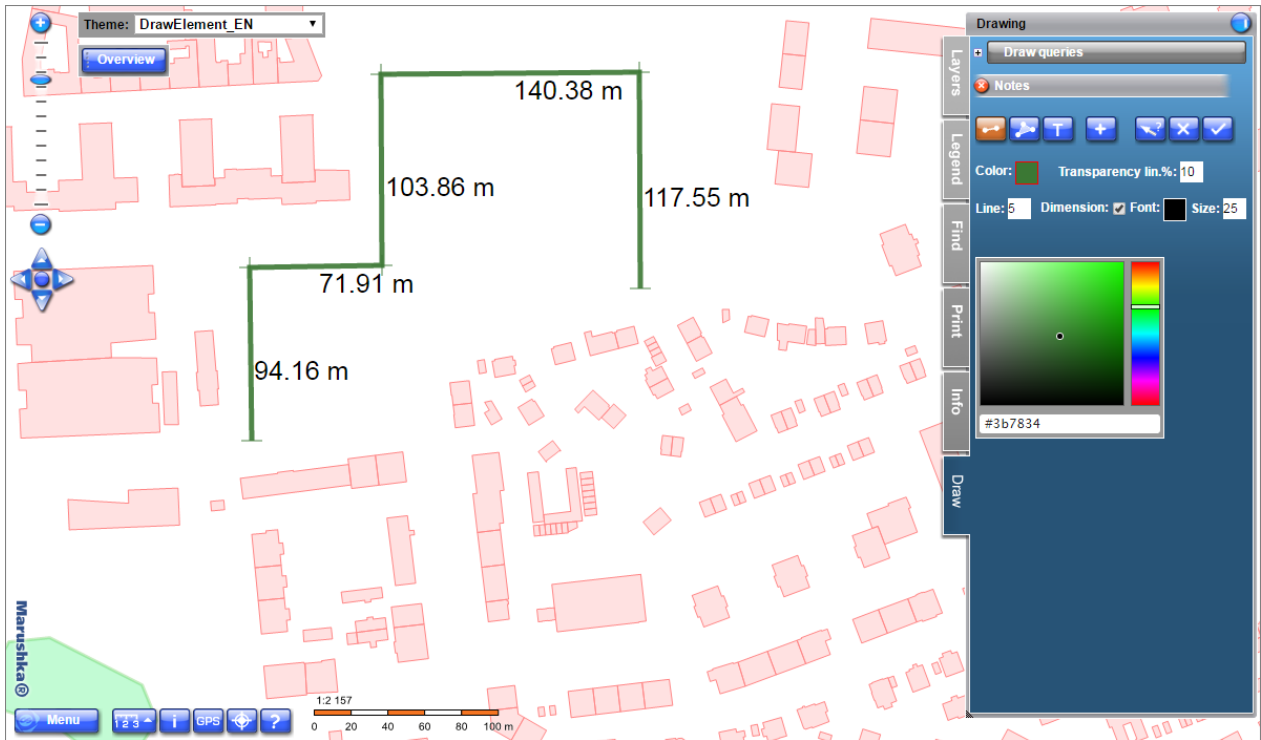


Fig 3: Example of drawing notes – line with dimensions

## 4 A Brief Description of the Example in Marushka Design

The test example contains a SQLite database with one publish layer. There is one form layer in the data store, related to the physical layer (database table). Data bearer is the physical layer (database table) "GS\_TABLE".

For proper functionality we need to have created items in the Etalon library. In this library are 4 records. All these items have in property *ElmAttribs* set attribute NOTE, making it possible to add to a drawn element some text note. *Geomtype* property has completed values *WKBPoint*, *WKBLineString* or *WKBPolygon*, by type of drawn items. Property *TableName* refers to the database table name, in this case *GS\_TABLE*.


The project includes several queries in the query library.

Query *Draw Cell* draws a cell *PointGray* from the cell library, query *Draw Cell with static code list* includes static code list (auxiliary query in the query library) *List of static values (Cell code list)*. This auxiliary query has filled in the following properties: *ListOfValues*, which lists the names of the cells from which you can choose. Furthermore the *QueryLV* property, which refers to the ID of the parent query *Draw Cell with static codebook* and is connected with it. Ultimately, it is *QueryLVNUM* property that defines the order of the parameter in the parent query.

The actual parent query *Draw cell with a static code list* contains mainly SQL phrase in the property *SlqStmtTemplate*: `SELECT ~(string)1~ "Notice", ~(string)2~ CELLNAME`. Both parameters must be again defined also in the property *QueryParameters*.

There is also the trio of queries *Draw Point*, *Line* or *Polygon*, allowing drawing these graphic elements and adding the text note to them. These queries have in form layer properties *SlqStmtTemplate* set: `SELECT '~ 1 ~' "NOTICE"`, which allows them to add text notes. This parameter must be defined again in *QueryParameters* properties. Another choice is called *Draw Text*, using which you can add a text to a reference point. This query is defined similarly as *Draw Point* query, but in Symbology it has checked option Font size with value 5 000 in target units of the data store (here mm).

There is also included a query *Delete element* through which it is possible to delete any element in the map from the database. The last query in the query library is information query called *Information query*, returning information about the selected element.

In the local web server environment is possible to select elements by clicking on button *Preselect*  in the functional keys menu, and then delete it by clicking on option Delete.

In the local web server environment, in panel *Draw* is item *Notes*, allowing drawing temporary notes into the map window. These notes, however really works as notes – they are not saved into the database and after working with the project are deleted.

There are three types of notes. The first one is the query *Create line*. You can set the line color, opacity, width and there is also a possibility to display distance between individual fracture vertices (dimensions) in terms of numerical values. It is also possible to set height and color of the displayed text.

Another note type is *Create polygon*. It is possible to set color and transparency of both line and fill.

The third note type is *Create text*, allowing creation of text notes. It is possible to set color, transparency, font height and of course the text of this note.

As mentioned above, these notes servers only to note something and they are not saved into the database. These notes can be printed or saved into PDF or PNG.