

Cascading List of Values in MarushkaDesign



GEOVAP

CONTENTS

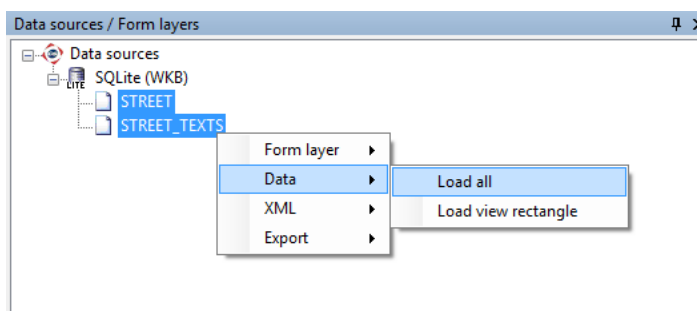
1	AIM OF THE EXAMPLE.....	2
2	WORKING WITH EXAMPLE	2
3	DIALOG BOX SAMPLE.....	3
4	A BRIEF DESCRIPTION OF THE EXAMPLE IN MARUSHKADESIGN	4

1 Aim of the Example

In this example we will demonstrate work with cascading list of values 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 **CascadingListOfValues.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 **CascadingListOfValuest_EN.xml** in MarushkaDesign environment.
- Select form layer STREET and STREET TEXTS in SQLite (WKB) data store, in the context menu choose Data – Load all:





- In map window choose „Fit all“:



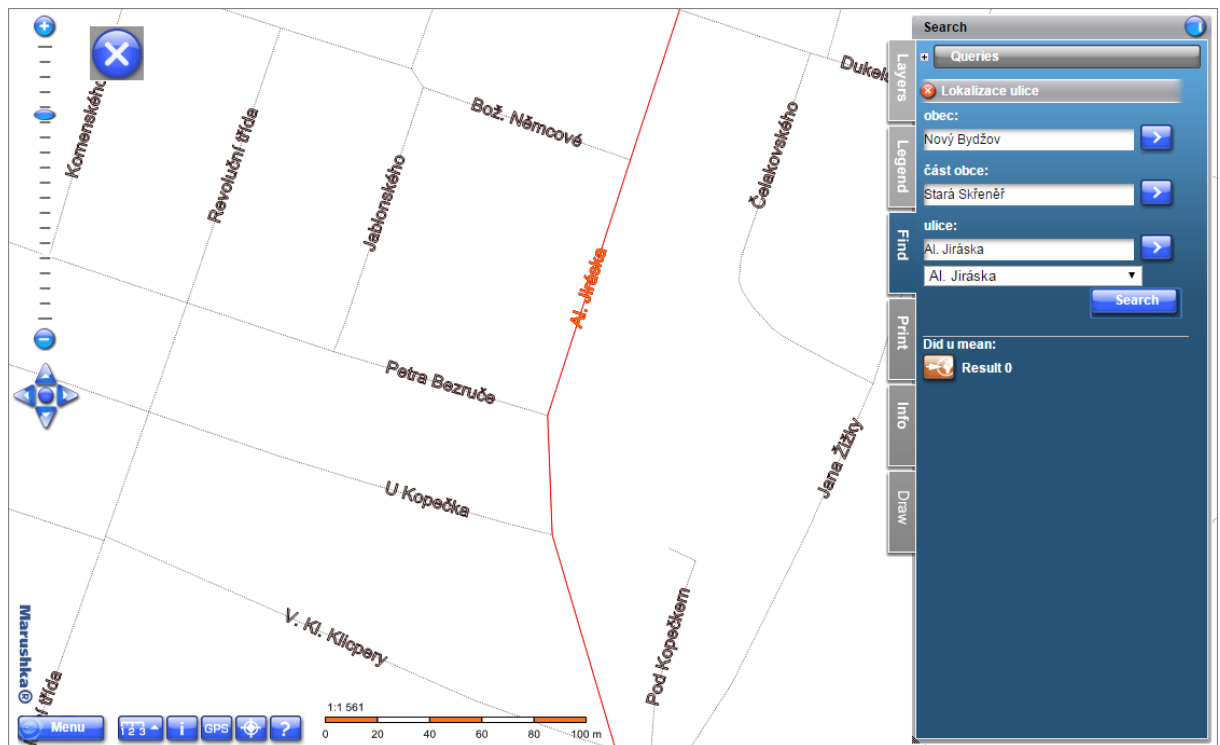
- Launch the local web server:



- In panel tabs click on the “Search” tab and select the Query *Localize street* by clicking on it. Three fields for entering municipality, municipality part and street appears. First, you need to enter the value for field municipality, where you enter the full name of one of the municipalities (Nový Bydžov, Skřivany, Smidary) and click on the button  next to the search box. It will find the municipality name that you specified. Instead of the full municipality name it is possible to enter only a fragment of the name and then search it by the same button. It is also possible to directly click on the button (without entering any text) and the menu of all available possibilities will display.
- The same procedure is applied on the selection of municipality part and street. Due to the relatively small number of records in the database, it is recommended to use  button to search available records.

3 Dialog Box Sample

Fig 1: Result of query *Localize street*



4 A Brief Description of the Example in MarushkaDesign

The aim of this example is to demonstrate the dependency of cascading lists of values in MarushkaDesign. Using the localization query with cascading lists of values can be cascaded searched municipality, municipality part, street, address and localize the query result.

This example due to simplicity and clarity involves only 3 municipalities from POU (administrative district) Nový Bydžov, and their municipality parts and streets. Specifically, the municipalities are Nový Bydžov, Smirady and Skřivany,

In the real project would be of course possible to search for addresses e.g. in the whole Czech Republic, in the whole world or in any other spatially defined area. The database, which would be used for searching, however, would have a considerable amount of data which is not suitable for this example, and therefore was chosen only the simplified version.

An example contains a form layer STREET, which display the street lines of the mentioned municipalities and form layer STREET_TEXTS displaying texts. These form layers are related to the database table ULICE_CARA.


In the query library is a localization query *Localize street*, for which were created three lists of values (code lists) – for each of the parameters just one: for selection of municipality, municipality part and street. All these lists of values (except the first one) has set property *Dependency* in the list of values properties to “True” which means that they are dependent upon the result of the previously substituted variables.

The list of values 1 selects all the municipalities and its value determines the selection of the municipality part in the given municipality. As first is evaluated the list of values 1, its current value is then substituted into the list of values 2. List of values 2 is then dependent on list of values 1. Similarly, list of values 3 depends on the current value of the list of values 2, but also on the current value of the list of values 1. Before evaluating the localization query are substituted values of all the parameters that were completed.

The most important functionality that relates to this example is the dependency of the list of values. This means that after writing the value “Nový Bydžov” in the search box it is possible to choose the parts of the municipality Nový Bydžov. This dependence is reflected so if you delete any value in the search box, all the fields below this field will be deleted (all the fields depended on this list of values).

In the event that for any specified initial character in the search exist a greater number of returned records, it would be practical to use search suggestion. It is activated by setting the property *DynamicCodeList* to “True”. The search suggestion then according to the typed characters offers an interactive menu of existing records defined in the dynamic list of values.

In this example, however, would not be practical to use search suggestion, because of a small number of options for each parameter. It would not be clear what initial character should the user enter and the query would not return any result. Therefore, this option is disabled for this query.

By clicking on the button  in the local web server environment are searched all the records corresponding to the value of this parameter (it is not necessary to enter any character into the search box), or you can click on the button to search occurrence of the specified characters.

The localization query property *SqlStmtTemplate* has the following form:

```
SELECT
D.XMIN XMIN, D.YMIN YMIN, D.XMAX XMAX, D.YMAX YMAX, D.KODUL ID, ''
NAZEV, 'KODUL' KEY_COLUMN
FROM
UIR_OBCE A, UIR_COBCI B, UIR_ULICE C, ULICE_CARA D
WHERE
A.NAZEV LIKE ~(string)1~ AND B.NAZEV LIKE ~(string)2~ AND C.NAZEV LIKE
```

```
~(string)3~ AND C.ULICE_ID=D.KODUL AND C.OBCE_ID=A.OBCE_ID
```

The query contains parameters ~ (string) 1 ~ ... ~ (string) 3 ~, where 1 ... 3 are indexes of order of the parameters listed in column *QueryParameters* of the associated localization query.

It is also possible to set symbology of the localization query result in the property *Symbology* of localization query. In this example, the query result is highlighted in red lines with orange fill.