Common Info Query Type in MarushkaDesign

GEOVAP

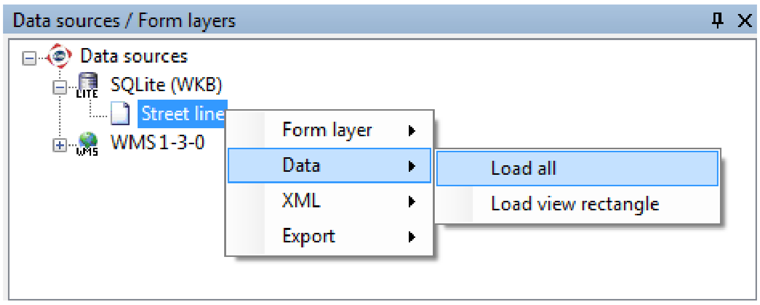
CONTENTS

1 Aim of the Tutorial

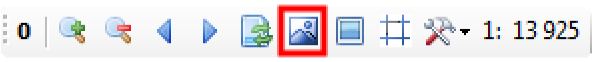
In this tutorial we will demonstrate functionality and practicability of Common info query type in MarushkaDesign. This tutorial was created in version 4.0.1.0, so it does not have to be compatible with older versions.

2 Work with the Tutorial

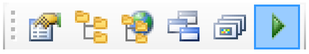
- Unzip the **CommonInfo_EN.zip** into **C:\MarushkaExamples\** folder. The target folder must be respected due to interconnection of paths with the project. In the case of placing the files in the different folder, it would not be possible to work with the tutorial.
- Open the **CommonInfo_EN.xml** in MarushkaDesign environment.
- Select form layer *Street line* in SQLite (WKB), in the context menu choose *Data – Load all*:



- In map window choose “Fit all”:

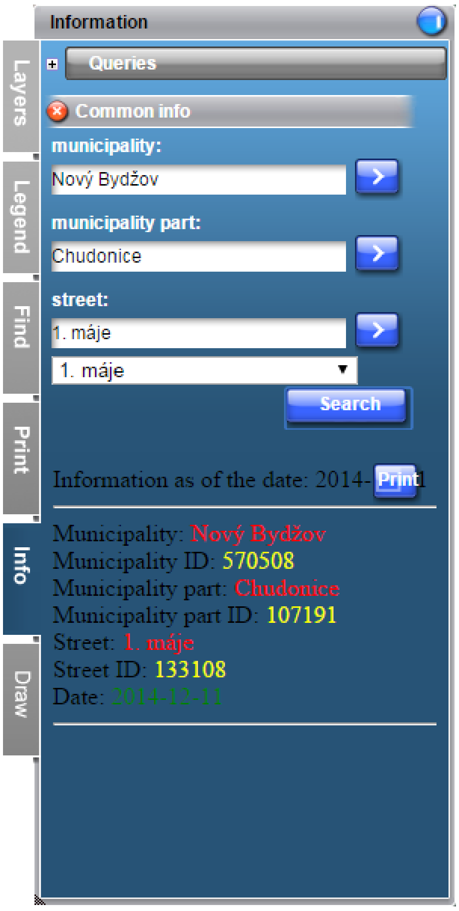


- Launch the local web server :



3 Dialog Box Sample

Fig 1: *Common Info* query result in panel



municipality: Nový Bydžov

municipality part: Chudonice

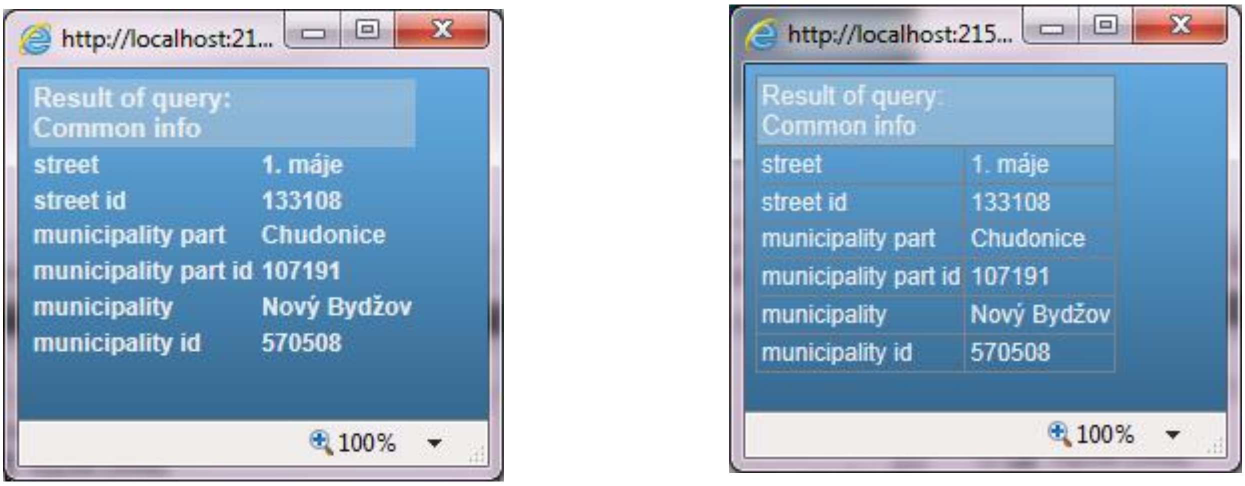
street: 1. máje

1. máje

Information as of the date: 2014-12-11

Municipality: Nový Bydžov
Municipality ID: 570508
Municipality part: Chudonice
Municipality part ID: 107191
Street: 1. máje
Street ID: 133108
Date: 2014-12-11

Fig 2: Comparison of results of the query *Common info*, displayed in a new window, without a completed HTML form, on the left was used *TableStyle = nogrid*, on the right side was used *TableStyle = grid*



Result of query: Common info

street	1. máje
street id	133108
municipality part	Chudonice
municipality part id	107191
municipality	Nový Bydžov
municipality id	570508

Result of query: Common info

street	1. máje
street id	133108
municipality part	Chudonice
municipality part id	107191
municipality	Nový Bydžov
municipality id	570508

4 A Brief Description of the Tutorial in MarushkaDesign

The aim of this tutorial is to demonstrate the functionality and usability of query type *Common info* in MarushkaDesign. Query *Common info* allows displaying common information without selecting a graphical element in the map window. In the web publication is this query displayed by default in panel *Info*. This query allows you to define multiple input parameters, which can be code lists/lists of values defined in the cascade. This type of query is not associated with the element *LayerName*, this information is mentioned in parameter *SlqStmtTemplate*.

Using this query, for which were created lists of values, can be cascaded search for municipality, municipality part and street and can be displayed any information to this result, such as HTML page.

The project due to simplicity and clarity involves only 3 municipalities from POU (the responsible designated authority) Nový Bydžov, municipality parts and streets. Project includes municipalities Nový Bydžov, Skřivany and Smidary. Form layer *Street line* displays street lines of these municipalities from SQLite (WKB) data store, for better visibility are street lines displayed by yellow and texts by red. There is also an underlying ortophoto map of Czech Republic, connected via WMS 1.3.0 (form layer *Ortofoto* in WMS 1-3-0 data store).

In the query library are created three lists of values for query *Common info*. For each of defined parameters, there is just one list of values: for selection of municipality, municipality part and street. All of these lists of values, except the first one have set item *Dependency* in the properties of list of values to *„True“*, which means that they depend upon the results of previously substituted variables.

It is necessary to define inter alia parameters *QueryParams* (number of parameters varies between 0 and N), *SlqStmtTemplate* and optionally *ResultTemplate*. These parameters are mentioned in detail in chapter 4.1 of this document.

In practice, it is possible using this query to display to the searched municipalities, municipality parts and streets e.g. their *ID*s, as shown in the attached project. Instead of *ID* it is possible to display any common information from the database, e.g. total population, electoral participation, or any other information related to a given spatial element.

4.1 Query Common info properties:

In the following inventory is a list of properties, which can affect behavior of query type *Common info*. Individual properties are classified into categories for clarity, as it is in form layer properties in MarushkaDesign.

4.1.1 Category 2. Query properties

- ***QueryBuf*** – Maximum number of results per query. The default value is 1.
- ***QueryParameters*** – Definition of parameters that are used as search parameters in SQL query.
- ***ResultTemplate*** – This query feature may contain the complete HTML (voluntarily), including javascript styles (here specifically red, yellow and purple color query results). Everything between ~DETAIL~ and ~DETAIL_END~ will be repeated according to the number of results. The names prefixed by ~~ (tilde) are named aliases from an SQL query.

Following is an example of property *ResultTemplate* containing HTML form in the attached project, note: notes in purple are just supplementary information, they are not mentioned in the HTML form itself:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Test</title>
Information as of the date: ~SYSDATE~
</head>
<body>
```

If the template returns at least one row, you can use its values also in the header – it is used for example for constants, when for all the rows is returned SYSDATE (current date), and in the header is displayed text: Information as of date: ~DATE~, where the value ~DATE~ will be replaced by the current date.

Here is the beginning of the page; a full HTML can be inserted here, including javascript styles etc. Everything between ~DETAIL~ and ~DETAIL_END~ will be repeated according to the number of results. The value of the first row:

```
~DETAIL~
<hr>
Municipality: <span style="color: red">~municipality~</span>
<BR>
Municipality ID: <span style="color: yellow">~municipality id ~</span>
<BR>
Municipality part: <span style="color: red">~municipality part~</span>
<BR>
Municipality part ID: <span style="color: yellow">~municipality part
id~</span>
<BR>
Street: <span style="color: red">~street~</span>
<BR>
Street ID: <span style="color: yellow">~street id ~</span>
<BR>
Date: <span style="color: green">~SYSDATE~</span><BR>
~DETAIL_END~

<hr>
<BR>
<br>
```

Here is the bottom of the page, which is also a complete HTML

```
</body>
</html>
```

- ***SlqStmtTemplate*** – SQL query template

Following is an example of SQL query *SlqStmtTemplate* from the attached project:

```
SELECT A.NAZEV "street", A.ULICE_ID "street id" , B.NAZEV "municipality
part", B.COBCI_ID "municipality part id", C.NAZEV "municipality", C.OBCE_ID
"municipality id"
FROM
UIR_ULICE A, UIR_COBCI B, UIR_OBCE C
WHERE
A.NAZEV LIKE '~3~%'
AND A.OBCE_ID = B.OBCE_ID AND B.NAZEV LIKE '~2~'
AND B.OBCE_ID = C.OBCE_ID AND C.NAZEV LIKE '~1~'
ORDER BY
A.NAZEV ASC
```

Using the query above is defined which information from the database are displayed in the *Common info* query result. There are also defined column aliases; these are then substituted into HTML form *ResultTemplate*.

The query contains parameters ~(string)1~ … ~(string)3~, where 1 … 3 are indexes of order of the parameters listed in item *QueryParameters* of the associated query *Common info*.

4.1.2 Category 4. Displaying of result

- ***ResultHeight*** – height of info bubble (in pixels)
- ***ResultWidth*** – width of an info bubble (in pixels)
- ***TableStyle*** – type of info query view in client (*grid* – with grid / without grid), see Fig 2
- ***TargetName*** – query will display in new window with a given name
- ***ViewStyle*** – type of info query view in client (*InNewWindow* – result is displayed in new window (see Fig 1), *InPanel* – result is displayed in new window, *InPopUpBubble* – result is displayed in pop up bubble)
- ***WindowLeft*** – indentation from the left side of window when opening a new window
- ***WindowOrigin*** – determines corner of the client for calculation of its position
- ***WindowTop*** – indentation from the top while opening in a new window