Rendering Image from the Database Column to Referenced Geometry Point

GEOVAP

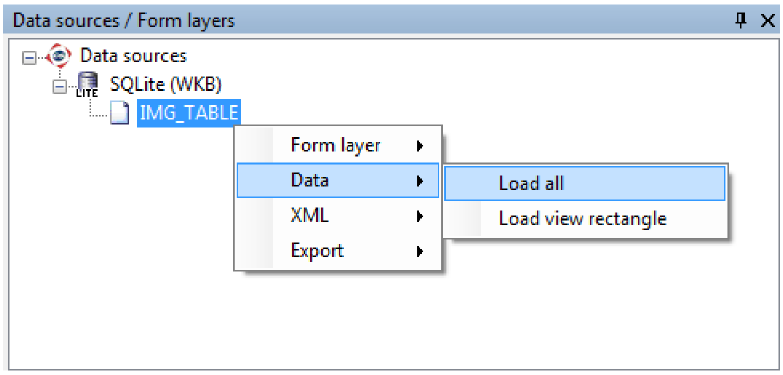
CONTENTS

1 Aim of the Example

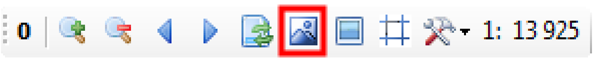
In this example we will demonstrate how to replace a standard geometric element (POINT/LINE/POLYGON etc.) in its reference point with the image (PNG/JPEG/GIF). Image is stored in SQLite database, in column IMG type BLOB of the geometric table. This example was created in version 4.0.3.14, so it does not have to be compatible with older versions.

2 Working with Example

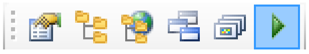
- Unzip the **ImgFromPoint_EN.zip** into **c:\MarushkaExamples\** folder. The target folder must be respected due to interconnection of paths with the project. In the case of placing the files in the different folder, it would not be possible to work with an example
- Open the **ImgFromPoint_EN.xml**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



- In map window choose “Fit all”:

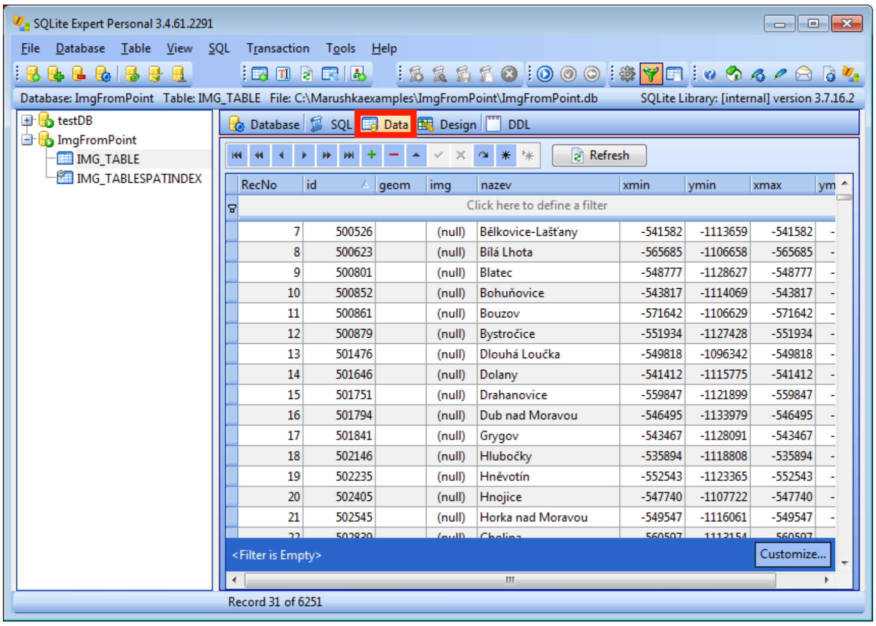


- Launch the local web server:



3 Dialog Box Sample

Fig 1: Switching to the data mode in SQLite Expert Personal manager


SQLite Expert Personal 3.4.61.2291
File Database Table View SQL Transaction Tools Help
testDB
ImgFromPoint
IMG_TABLE
IMG_TABLESPATINDEX
Database SQL Data Design DDL
Refresh
Click here to define a filter
Customize...

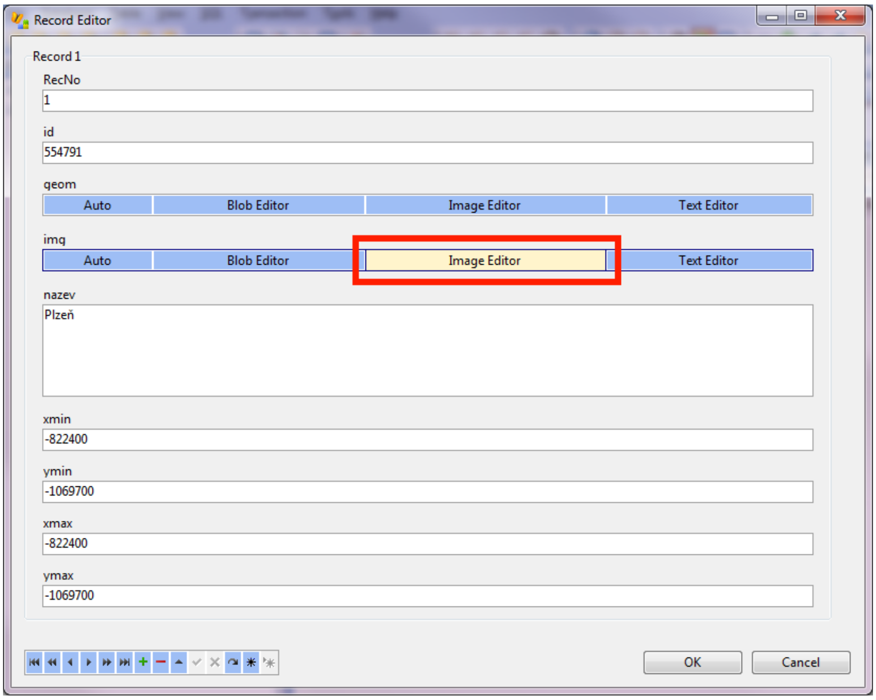
Database: ImgFromPoint Table: IMG_TABLE File: C:\Marushkaexamples\ImgFromPoint\ImgFromPoint.db SQLite Library: [internal] version 3.7.16.2

RecNo	id	geom	img	nazev	xmin	ymin	xmax	ym
7	500526		(null)	Bělkovice-Lašťany	-541582	-1113659	-541582	-
8	500623		(null)	Bílá Lhota	-565685	-1106658	-565685	-
9	500801		(null)	Blatec	-548777	-1128627	-548777	-
10	500852		(null)	Bohuňovice	-543817	-1114069	-543817	-
11	500861		(null)	Bouzov	-571642	-1106629	-571642	-
12	500879		(null)	Bystročice	-551934	-1127428	-551934	-
13	501476		(null)	Dlouhá Loučka	-549818	-1096342	-549818	-
14	501646		(null)	Dolany	-541412	-1115775	-541412	-
15	501751		(null)	Drahanovice	-559847	-1121899	-559847	-
16	501794		(null)	Dub nad Moravou	-546495	-1133979	-546495	-
17	501841		(null)	Grygov	-543467	-1128091	-543467	-
18	502146		(null)	Hlubočky	-535894	-1118808	-535894	-
19	502235		(null)	Hněvotín	-552543	-1123365	-552543	-
20	502405		(null)	Hnojice	-547740	-1107722	-547740	-
21	502545		(null)	Horka nad Moravou	-549547	-1116061	-549547	-
22	502830		(null)	Cholina	-560507	-1113154	-560507	

<Filter is Empty>

Record 31 of 6251

Fig 2: Selection of Image Editor in SQLite Manager



Record 1

RecNo
1

id
554791

geom

img

nazev
Plzeň

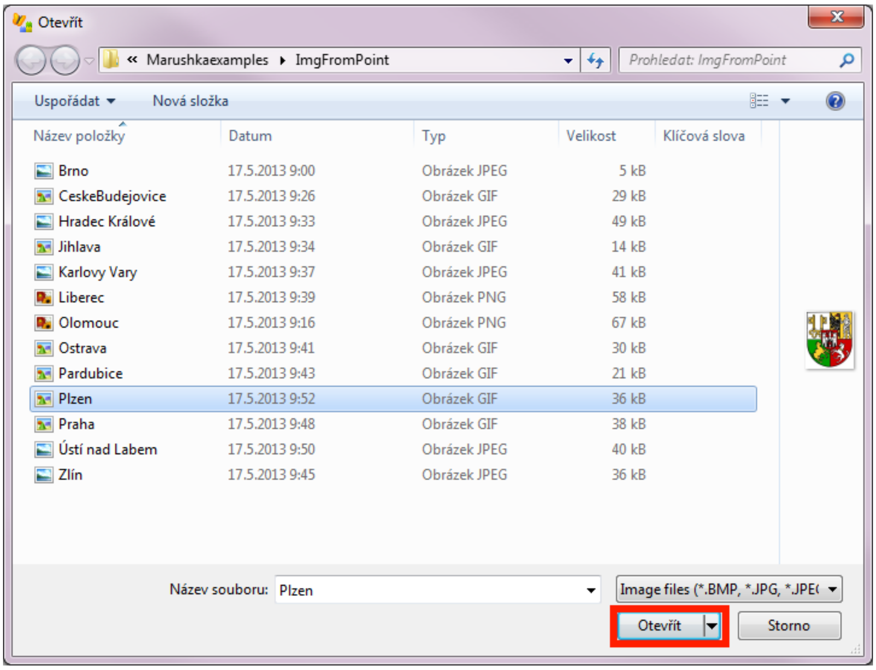
xmin
-822400

ymin
-1069700

xmax
-822400

ymax
-1069700

Fig 3: Loading image in SQLite Manager

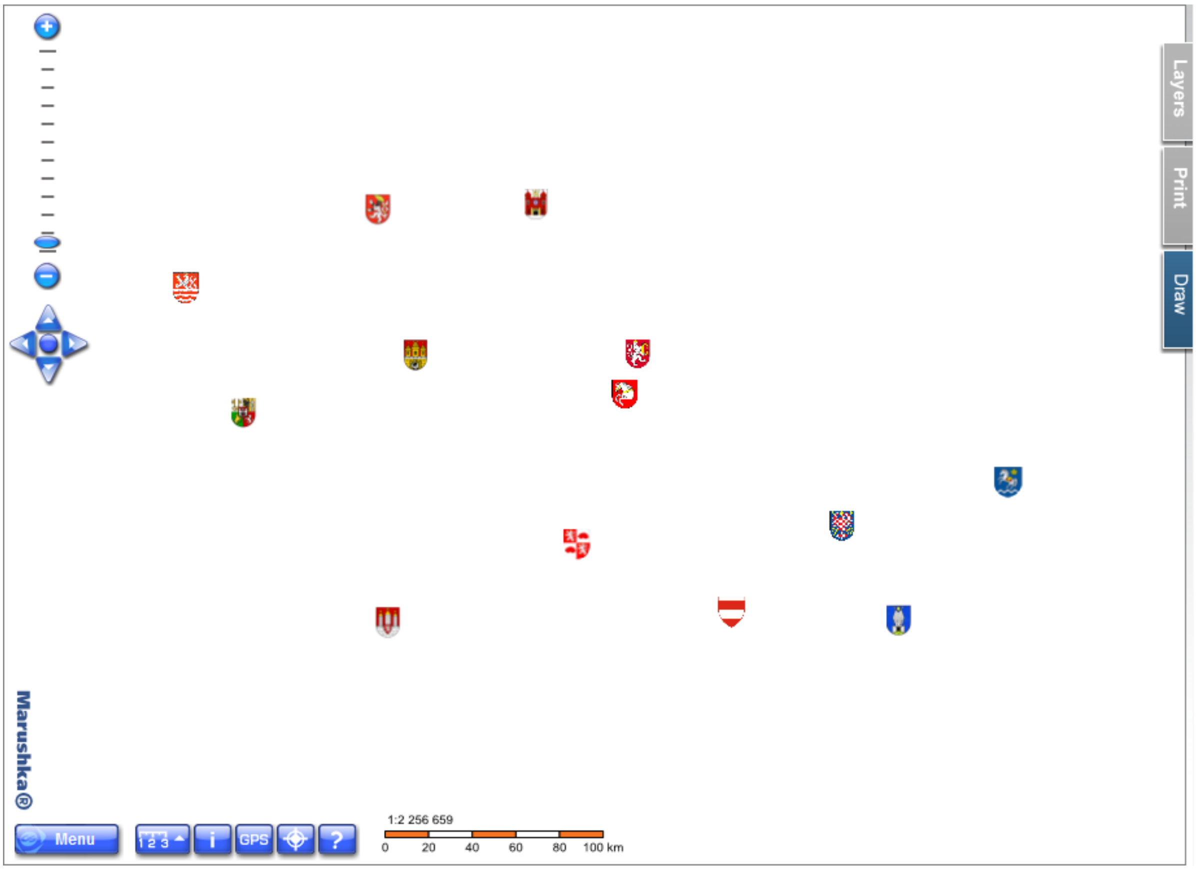


Název položky	Datum	Typ	Velikost	Klíčová slova
Brno	17.5.2013 9:00	Obrázek JPEG	5 kB	
CeskeBudejovice	17.5.2013 9:26	Obrázek GIF	29 kB	
Hradec Králové	17.5.2013 9:33	Obrázek JPEG	49 kB	
Jihlava	17.5.2013 9:34	Obrázek GIF	14 kB	
Karlovy Vary	17.5.2013 9:37	Obrázek JPEG	41 kB	
Liberec	17.5.2013 9:39	Obrázek PNG	58 kB	
Olomouc	17.5.2013 9:16	Obrázek PNG	67 kB	
Ostrava	17.5.2013 9:41	Obrázek GIF	30 kB	
Pardubice	17.5.2013 9:43	Obrázek GIF	21 kB	
Plzen	17.5.2013 9:52	Obrázek GIF	36 kB	
Praha	17.5.2013 9:48	Obrázek GIF	38 kB	
Ústí nad Labem	17.5.2013 9:50	Obrázek JPEG	40 kB	
Zlín	17.5.2013 9:45	Obrázek JPEG	36 kB	

Název souboru: Plzen

Image files (*.BMP, *.JPG, *.JPE(

Fig 4: Preview of the example result in the local web server in MarushkaDesign



4 A Brief Description of the Example in MarushkaDesign

The aim of this example is to demonstrate how to save binary representation of the image into a column in SQLite database and then retrieve it using *pseudo column* `SET_PARS_IMAGE_FROM_CORG` instead of point elements and render these images into map composition.

This example contains the form layer IMG_TABLE, which is related to the database table IMG_TABLE. This table contains images, whose binary representation is stored in the column IMG.

In this example, these images are specifically emblems of the regional cities in the Czech Republic, which are located in the coordinates of corresponding cities. Images are rendered in the size of 20 000 meters.

The database table in SQLite contains in addition to the standard columns also column IMG, which has the same type as column GEOM – BLOB.

Into this column will be in SQLite manager using Image Editor saved binary representation of image, which will be rendered in the place of reference point of the point element. Instead of the reference point of the point element, the image can be also displayed in the place of reference point of element type LINESTRING, POLYGON or GEOMETRYCOLLECTION.

The procedure is as follows:

1. ***Data preparation***

   First, we will **prepare data in SQLite manager** (there was used SQLite Expert Personal 3.5.89.205, hereinafter referred to as SQLite manager). In SQLite manager, go to *Data* mode (*see Fig 1*), find the record for which you want to change the image. At this record double-click on column IMG. The *Record Editor* will display, then click on option *Image Editor* at column IMG (*see Fig 2*). Next click on *Load*, select the image you want to display in the given coordinates and click on *Open* (*see Fig 3*). Then click on *OK*. Now, the image is saved in the database in binary format and it can be rendered.

2. ***Rendering (presentation) of data***

   In **MarushkaDesign** create a form layer, which will refer to the table with stored images (in this case IMG_TABLE). In the form layer properties of the layer *DBColumnsToClient* enter the string:

   ```
   IMG,'IMG 20000' SET_PARS_IMAGE_FROM_CORG, 'TRUE' SET_PARS_PIXELSIZE
   ```

   The string says that from the column IMG will be loaded images of size 20 000 meters.

   Instead of the value IMG in the string, it is possible to enter any name of the column, in which is stored binary representation of the image, instead of value 20 000 can be chosen any other image size.

   Pseudo column `SET_PARS_IMAGE_FROM_CORG` has two parameters: a) name of the database column in which is the image stored; this column must be loaded from the database through *DBColumnsToClient*, b) image height.

If the database column IMG contains also other records than just pictures, but we do not want to display these other records, it is possible to insert a string condition into form layer properties *DBWhereClause*: `IMG IS NOT NULL`. It will assure, that records, that has in column IMG value `NULL` (do not have assigned any image), will not be rendered. Otherwise, all the elements from the column IMG from the database table will be loaded.

Now it remains only to render the images in MarushkaDesign, do it in the same way as in the chapter 2 of this example (*Data – Load all*).