Google Map Integration in MarushkaDesign

GEOVAP

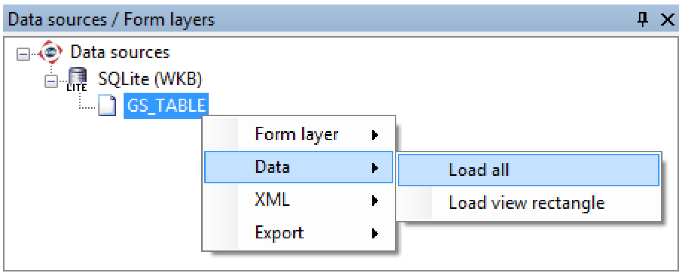
CONTENTS

1 Aim of the Example

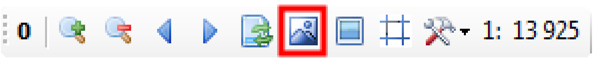
In this example, we will demonstrate integration of Google maps in MarushkaDesign. This example was created in version 4.0.1.0, so it does not have to be compatible with older versions.

2 Working with Example

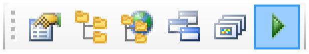
- Unzip the **GoogleMapsIntegration_EN.zip** into **c:\MarushkaExamples\** folder. The target folder must be respected due to interconnection of paths with the project. In case of placing the files in the different folder, it would not be possible to work with an example.
- Open the project **GoogleMapsIntegration_EN.xml**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



- In map window choose "Fit all":



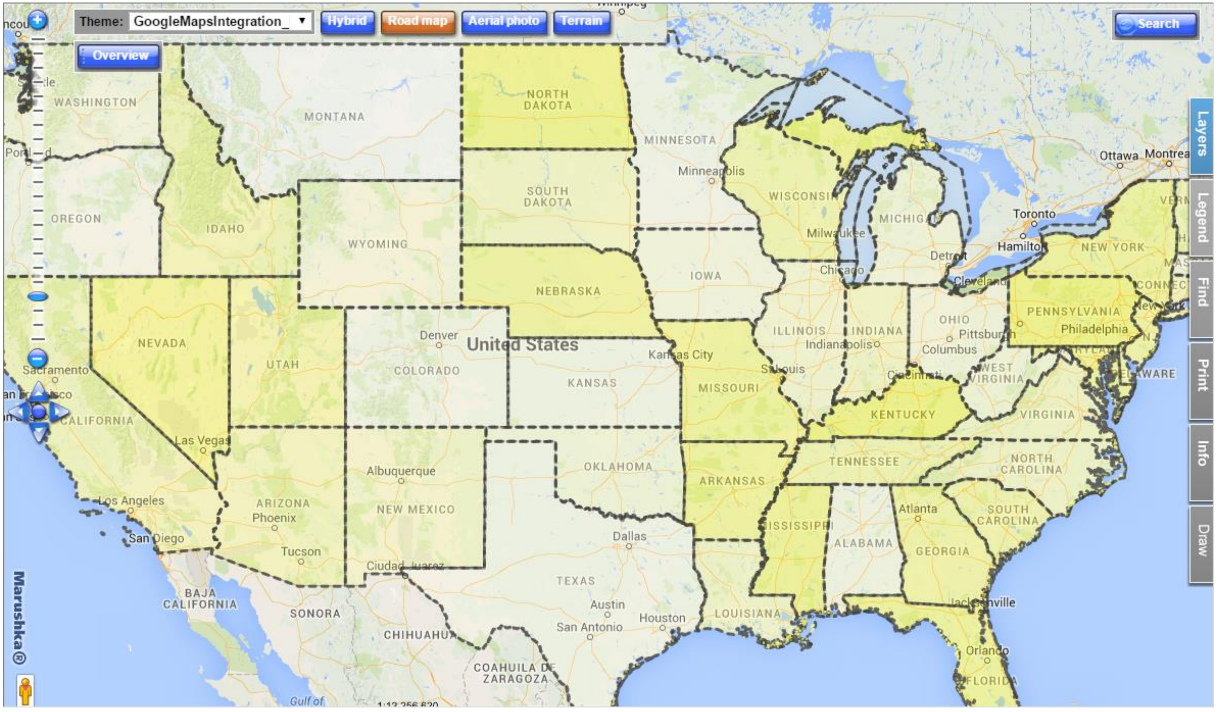
- Launch the local web server:



..

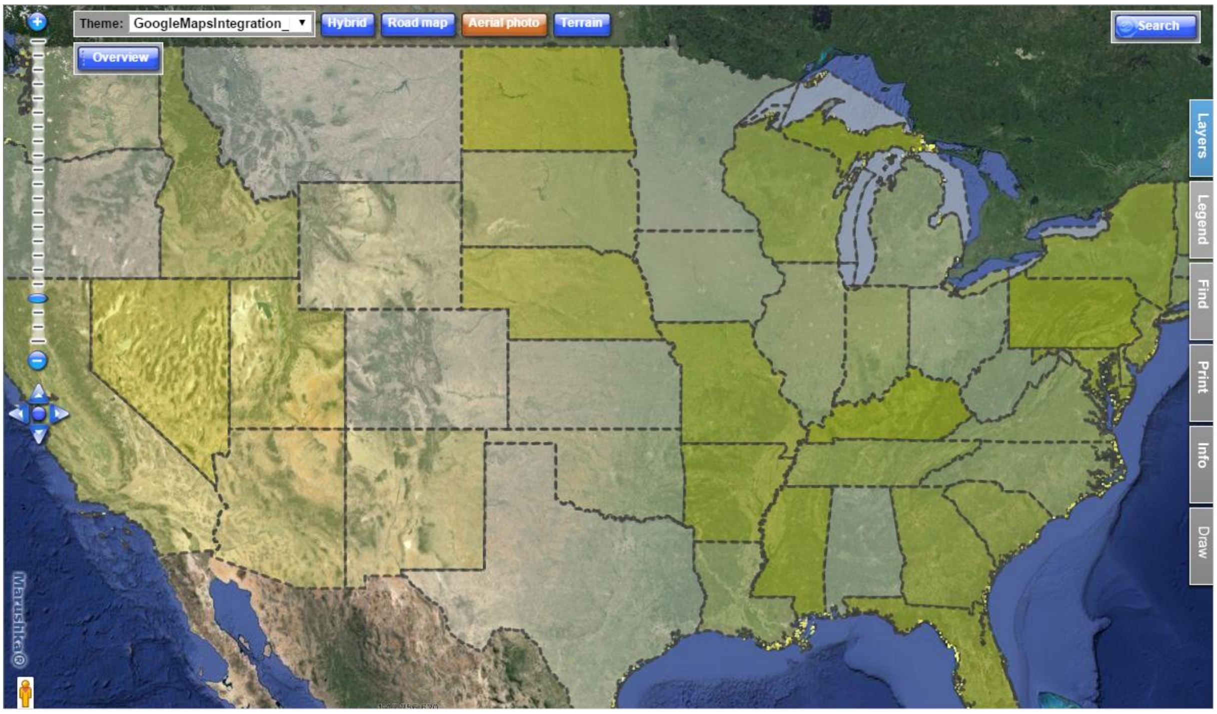
3 Dialog Box Sample

Fig 1: Google maps integration in local web server – Road map in Marushka



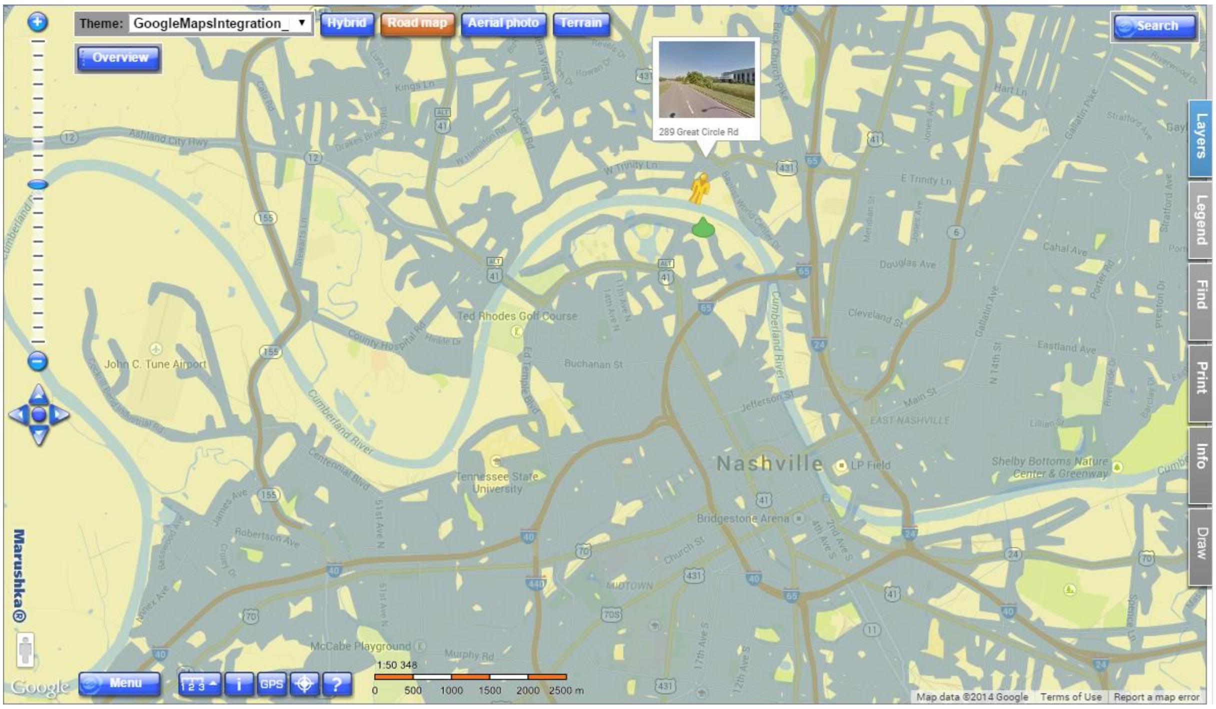
Theme: GoogleMapsIntegration_

Fig 2: Google maps integration in local web server - Aerial photo in Marushka



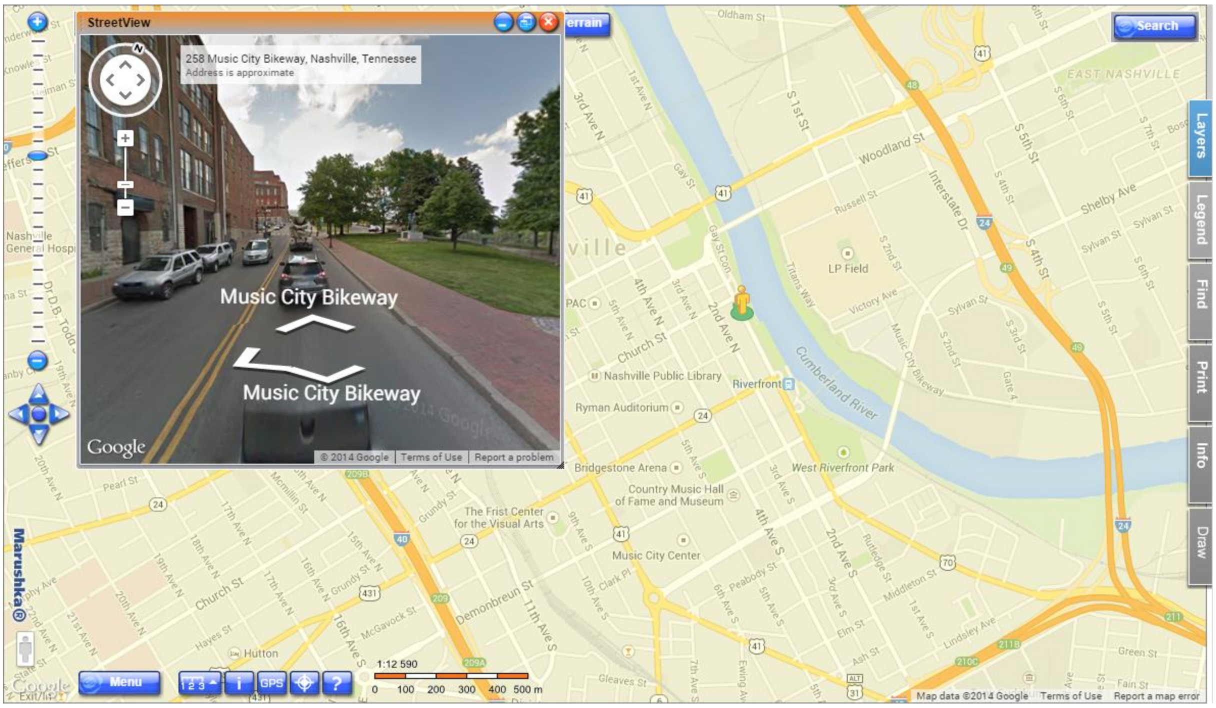
Theme: GoogleMapsIntegration_

Fig 3: Startup of StreetView in Marushka



Theme: GoogleMapsIntegration_

Fig 4: StreetView in Marushka



StreetView

258 Music City Bikeway, Nashville, Tennessee
Address is approximate

Fig 5: StreetView in Marushka - maximalized

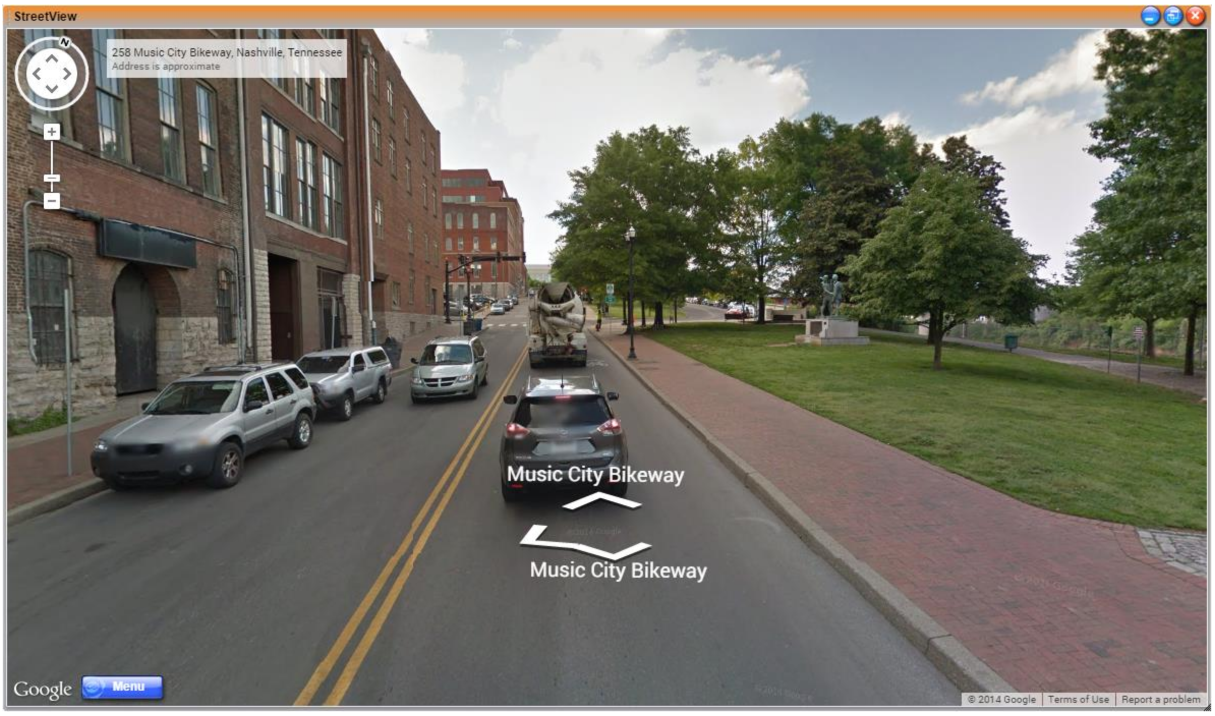
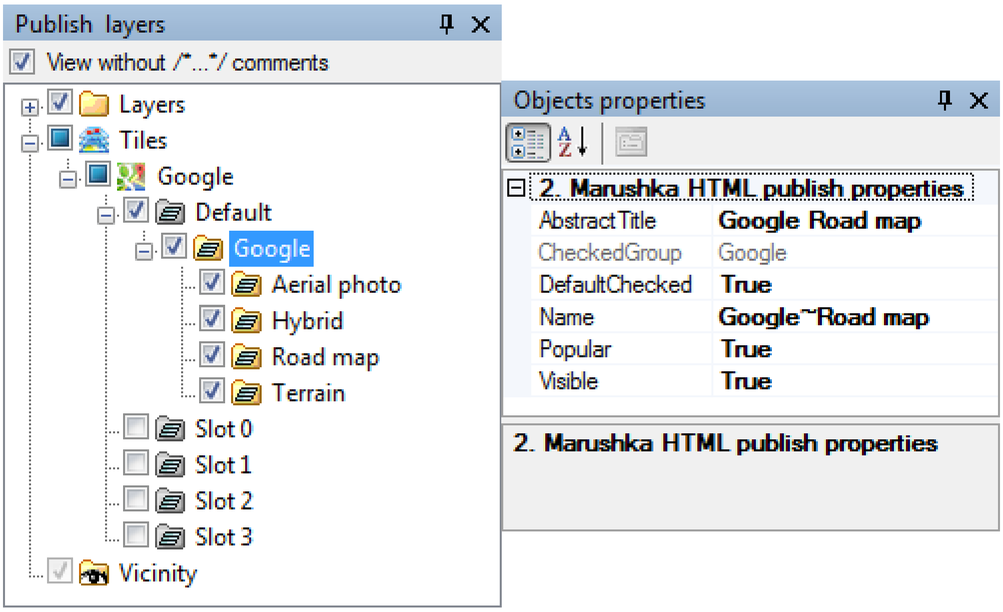


Fig 6: Google layers and its properties in Marushka



☑ View without /*...*/ comments

- ☑ Layers
- ■ Tiles
  - ■ Google
    - ☑ Default
      - ☑ Google
        - ☑ Aerial photo
        - ☑ Hybrid
        - ☑ Road map
        - ☑ Terrain
    - □ Slot 0
    - □ Slot 1
    - □ Slot 2
    - □ Slot 3
- ☑ Vicinity

2. Marushka HTML publish properties

AbstractTitle	Google Road map
CheckedGroup	Google
DefaultChecked	True
Name	Google~Road map
Popular	True
Visible	True

2. Marushka HTML publish properties

4 A Brief Description of the Example in MarushkaDesign

This test example contains a SQLite database with one publish layer. In the data source is one form layer related to physical layer (database table). Data bearer is a physical layer "GS_TABLE".

Form layer GS_TABLE displays the boundaries of the U.S. states. Using the command `'125 255 255 ' ||abs (random ()% 255) "SET_PARS_RGBFCOLOR"` are set fill colors for polygons of individual states, with the fact that for part of the color code (**ARG**) are set values (`'125 255 255'`). And the remaining part - **B** - variable value is generated using string `||abs(random()%255)`, which is the absolute value of a number in the range from 0 to 255. This number is generated automatically for each polygon and changes each time the map is redrawn. Thus, each time redrawing the map changes the color of each polygon.

Google map integration in MarushkaDesign environment is set in publish layer dialogue box, in item *Tiles*. Right mouse click on item *Tiles* invokes a menu, from which choose *Google maps tiles engine*. It will open individual Google layers items. Here check item *Default* and click on *Google*. In this folder are subfolders, representing individual Google layers, see Fig 6. There are 4 Google map layers available in MarushkaDesign (hybrid, road, aerial and terrain).

After clicking on folder Google it is possible to set in bulk values of items *Popular* – if layers will be displayed as a popular buttons in the local web server environment and *Visible* – if the items will be visible.

When clicking on individual setting items of Google layers, it is possible to set the previously mentioned two values also to individual layers. In addition, it is possible to set a hover text of the popular layer (*AbstractTitle*), if the layer will be displayed as default (*DefaultChecked*) and layer name (*Name*).

At one moment can be switched on just one from popular layers. And not all the layers are available in all the scales. For Google map integration it is necessary to have target data store in map projection Mercator (EPSG:3395) in units meter, otherwise the underlying map would not be displayed correctly.

While displaying polygons above underlying Google map, it is necessary to set the value **A** (**transparency**) in ARGB code using the command `SET_PARS_RGBFCOLOR` to value less than 255, to be visible underlying Google map. Otherwise it would not be visible and we would only see the map we created. It is the best to test the value of A, because it depends on individual color scheme of each map. In this example A is set to value 125.

In the local web server environment is a possibility to use application **StreetView**, which is activated by the figurine button at left bottom.

StreetView application is started so that we drag the figurine button by the left mouse click and drop it on the place in the map that we want to see it (more accurately a circle below the figurine). It runs in the new dialog box. StreetView offers insight into the streets scanned by Google and provides panoramic shots with a range of 360° horizontally and 290° vertically. Photos taken for StreetView are interconnected in a row every few meters so you can rotate view in all directions. StreetView is controlled by mouse, by Google cross (top left) or by keyboard arrows. It displays appoximate address of the current position and orientation is facilitated by displaying street names with directional arrows. Under the Google cross is a slider to adjust the zoom.