

# Generating the Text Element from the Reference Point of Symbol (Cell)



**GEOVAP**

## CONTENTS

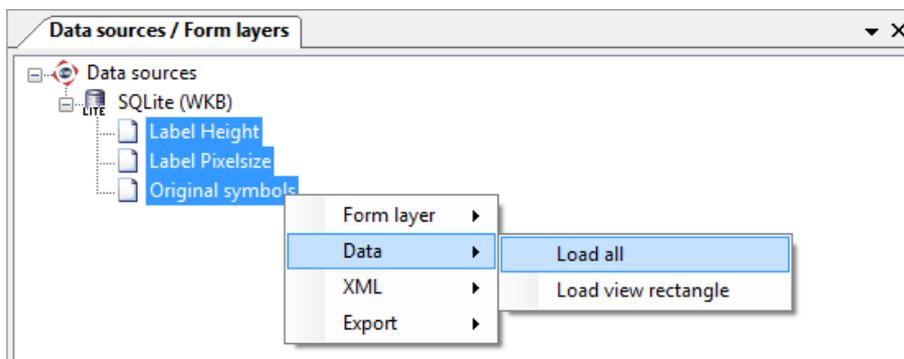
1	AIM OF THE EXAMPLE.....	2
2	WORKING WITH EXAMPLE .....	2
3	DIALOG BOX SAMPLE.....	3
4	A BRIEF DESCRIPTION OF THE EXAMPLE IN MARUSHKADESIGN .....	5

## 1 Aim of the Example

In this example, we will demonstrate how to generate a text element from the reference point of the symbol (cell) 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 **TextFromSymbolRefPoint\_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 **TextFromSymbolRefPoint\_EN.xml** in MarushkaDesign environment.
- Select all three form layers, in context menu select Data – Load all:



- In map window choose "Fit all":



- Launch the local web server:



### 3 Dialog Box Sample

Fig 1: Setting of the parameter *DBCColumnsToClient* of the form layer *Label Height*:

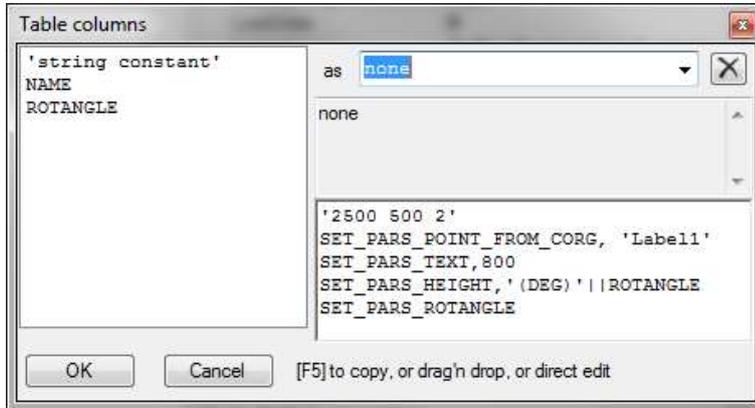


Fig 2: Setting of the parameter *DBCColumnsToClient* of the form layer *Label Pixelsize*:

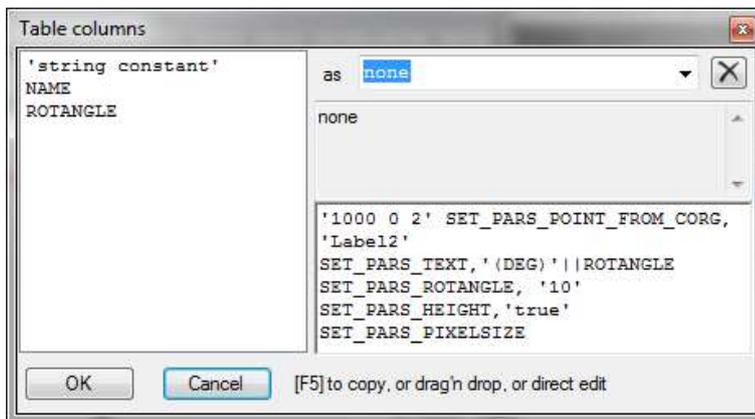


Fig 3: Sample of displaying two kinds of labels in the local web environment

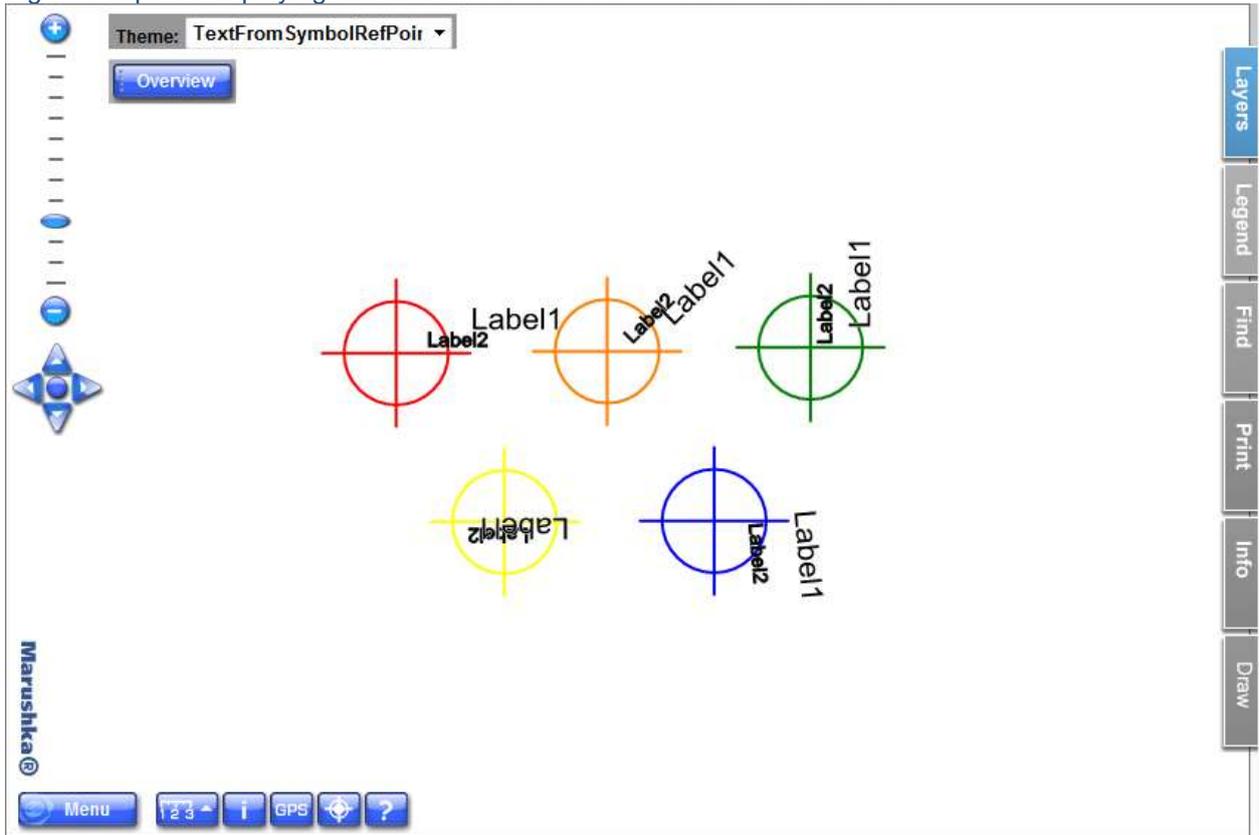
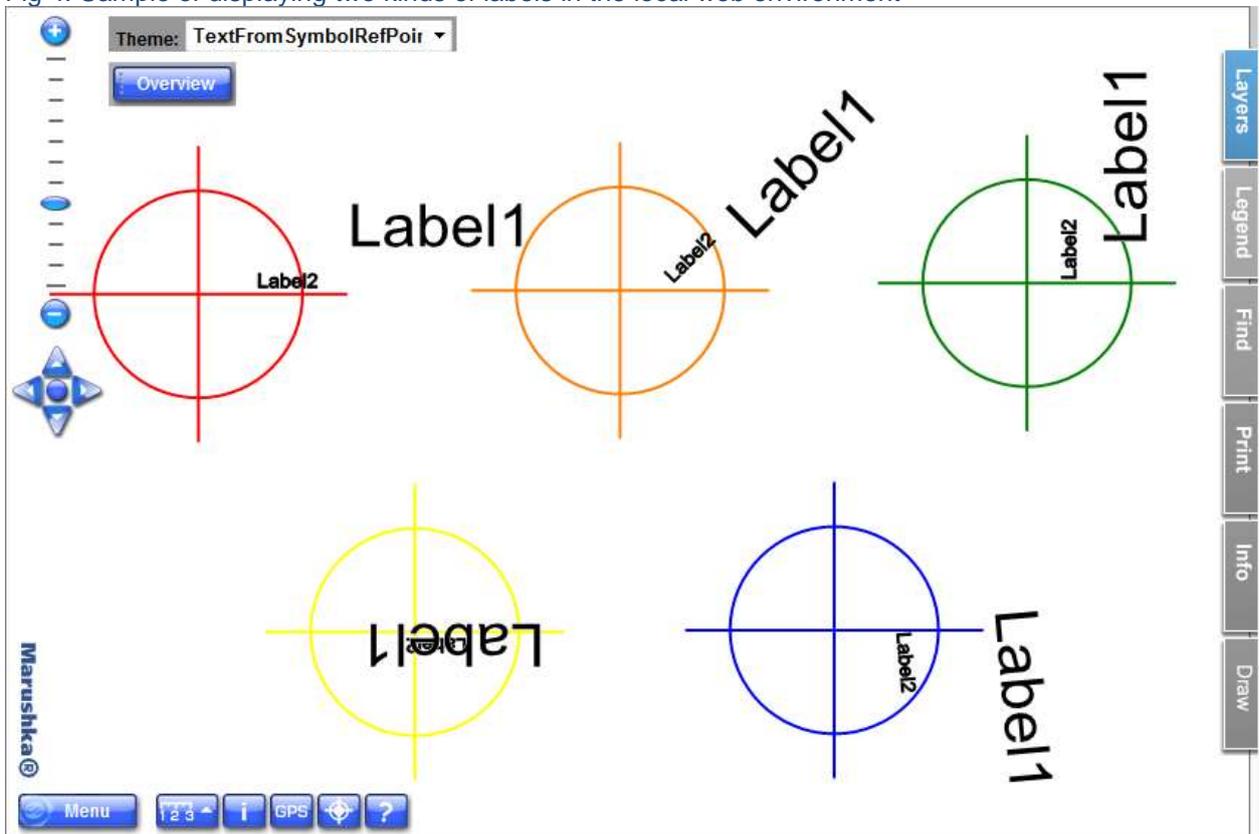


Fig 4: Sample of displaying two kinds of labels in the local web environment



## 4 A Brief Description of the Example in MarushkaDesign

This test example contains a SQLite database with one publish layer. In the data source are three form layers related to physical layer (database table). Data bearer is a physical layer "GS\_TABLE".

- A) Form layer **Original symbols** displays five graphical elements from GS\_TABLE in unchanged form.
- B) Form layer **Label Height** produces from the graphical elements in the table GS\_TABLE point texts, sets them size in real units and then turns these texts according to the angle defined by the column ROTANGLE. Size in real units' means that the text size varies depending on the scale.

In the property *DBCColumnsToClient* are used following commands:

**SET\_PARS\_POINT\_FROM\_CORG** produces from the source element a point. The parameters in quotes '2500 500 2' before the actual command means a shift: 2500 – movement along the x axis by 2500 source units, in our case meters, 500 – movement along y axis by 500 source units, in our case meters, 2 – text justification to left bottom.

The command **SET\_PARS\_TEXT** sets the text of point element (here is the value set to "Label1").

**SET\_PARS\_HEIGHT** command sets the real height of text (label), here is the height set to 800 source units (in our case meters).

Using the command **SET\_PARS\_ROTANGLE** is set rotation of text (the first text has a value of 0°; others are variously rotated according to their values ROTANGLE in table GS\_TABLE). It was necessary to connect '(DEG)' before this command, which allows entering the value of rotation in degrees, otherwise it would be necessary to specify the rotation value in radians.

- C) Form layer **Label Pixelsize** like the previous layer produces from graphic elements in table GS\_TABLE texts and rotates them. However, it sets the resulting elements size **in PIXELS**, so that regardless of the scale, the text will always have the same size.

In the property *DBCColumnsToClient* are used the same commands as in the layer *Label Height*, specifically **SET\_PARS\_TEXT** and **SET\_PARS\_ROTANGLE**, which are set to the same parameters of those commands.

The command **SET\_PARS\_POINT\_FROM\_CORG** is nearly the same, it just have set different shift, so that the texts do not overlap.

There is in contrast to layer *Label Height* used different command **SET\_PARS\_PIXELSIZE**, which is set to 'True', and defines that the value **SET\_PARS\_HEIGHT** is not defined in source units, but in pixels.

Pictures above show a distinct difference between displaying texts of layer *Label Height* and *Label Pixelsize*. For texts of layer *Label Height* is set value **Label1**, for texts of layer *Label Pixelsize* is set value **Label2**. When looking at two images from the local web environment it is apparent that the label *Label1* that is set to a fixed value in source units of the data store. When zooming, it increases in order to keep its fixed size (here 800 meters). As you zoom out, the label would obviously dwindle. While zooming, *Label2* maintains a constant pixel size (here 10 pixels), the pixel size is still the same at any scale.