WMS Service Client Connection Possibilities in Marushka Design

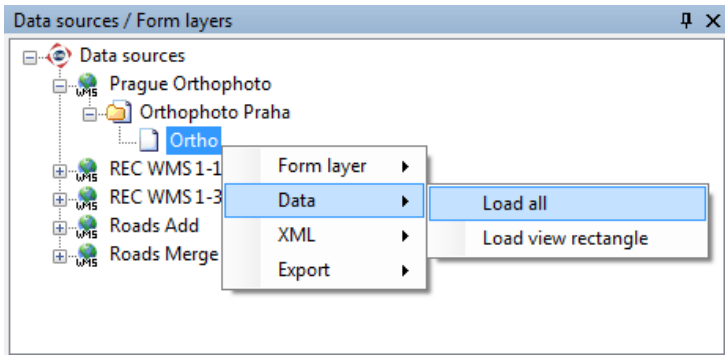
CONTENTS

1 Aim of the Example

In this example we will demonstrate WMS connection possibilities and some tricks 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 the Example

- Unzip the **WMSClient_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 the example.
- Open the **WMSClient_EN.xml** in MarushkaDesign environment.
- Select form layer *Ortho* in Prague Ortophoto data store, 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: Symbology settings set for layer *All REC 1.1.1*

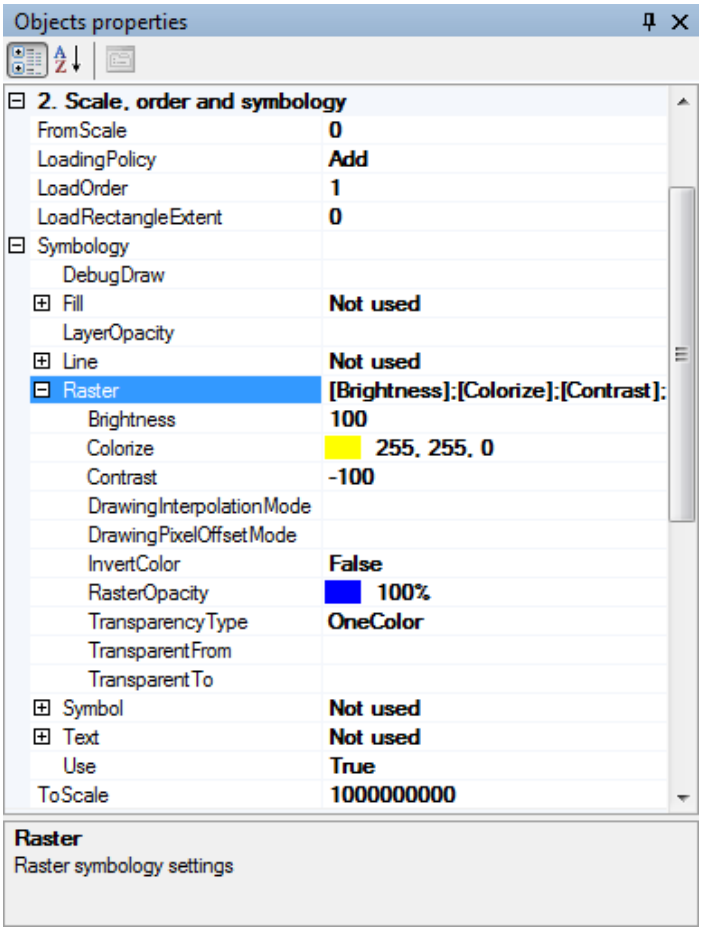
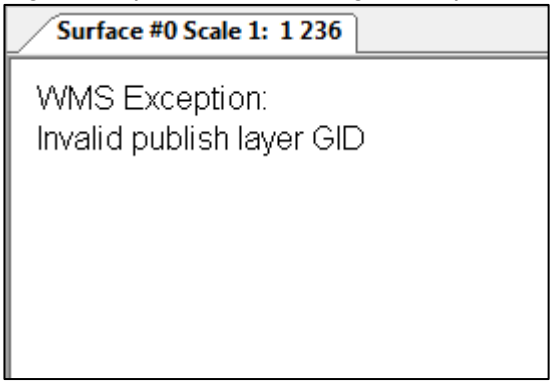


Fig 2: Sample of Error message in map window (feature *ShowErrorInImage*)



4 A Brief Description of the Example in MarushkaDesign

In this project, there is a total of 5 connected WMS data stores. An aerial photography of Prague is used as an underlying map in Prague area. In this project is also connected twice the same data store REC (Real Estate Cadaster), once via WMS 1.1.1, once via WMS 1.3.0. This is due to demonstration of differences between these two WMS versions. Another pair of data stores is also connected to this project – *Roads*, which is due to the demonstration of the difference between loading policies – *Add* and *Merge*.

The aim of this tutorial is to show the various possibilities of WMS connection from the client position and to teach some useful tricks with WMS connection.

4.1 WMS Connection Property Settings Possibilities

4.1.1 Form layer property settings

Below is a list of setting properties of individual layers; there are listed only properties, which are important for displaying layers connected via WMS. Individual items are divided into categories for clarity, as it is in the form layer properties in MarushkaDesign.

Category 2. Scale, order and symbology

- **FromScale/ToScale –** lower, respectively the upper limit of the scale for displaying form layers, it is important to reckon with the fact that it is not possible to exceed the scales set in WMS layer settings (in GetCapabilities file), respectively it is possible, but the layer would then return no data, or an error message.
- **Loading policy –** strategy of loading form layers
  1. Add – when sending a query to the relevant WMS server during publication, for each form layer is sent a separate request – in more complicated requests and also when having larger quantities of form layers, this strategy is time-consuming and computationally demanding. Try this policy when you are retrieving data from data store *Roads Merge* using *Clear and load*  policy. For monitoring of sent queries, you have to enable *Debug console* and function *Trace all*  . For each form layer is sent one request on the server.
  2. Merge – when sending a request to the relevant WMS server for publication, for all the merged form layers is sent just one request (in the query, in the parameter *LAYERS* is inserted list of all form layers, respectively comma separated *Name* values). When using this loading policy, it is not possible to modify symbology of data. In more complicated queries is in this way saved both time and computer performance. Try this policy when you are retrieving data from data store *Roads Add* using *Clear and load*  policy. For all the merged form layers is sent just one request on the server. You can verify it in *Debug console*, you have to enable function *Trace all* 
  3. You can achieve similar functionality by combining multiple layers into one form layer, see 4.2.2.
- **LoadOrder** – order (priority) of loaded form layers
- **Symbology** – the possibility of setting symbology for individual layers

Category 3. Request properties

- **RequestImageFormat** – requested graphical format in which the server returns the map image (works only for web services)
  1. PNG – this format supports transparency, it is relatively economical for vector drawing, each server defines this format slightly differently, but most common are PNG, PNG8bit and PNG24bit
  2. JPEG – it is a compressed format, it does not support transparency, it is suitable for photographs or ortophoto, in these cases it is economical; if it is used for vector drawing, there will be „blurred spots“ in the resulting image
  3. BMP – uncompressed format, very data demanding, it has a 24 bit color depth, images in this format are large sized, it does not support transparent drawing and it is not appropriate for publication
  4. GIF – uses lossless compression, but can contain only 256 colors, it is particularly suitable for vector data; it achieves the same results as PNG8bit, it is a commercial format and a lot of servers does not generate it correctly, due to this fact, the PNG format is more appropriate

4.1.2 WMS source property possibilities

Below are WMS sources property settings, in this particular case, it is necessary to mention only category 4. of these settings. Individual items are sorted in categories here, as well as in WMS source properties in MarushkaDesign, moreover here are mentioned the differences between WMS version 1.1.1 and 1.3.0.

Category 4. WMS source properties

A) Common for both versions

- **RequestTransparentImage** – if its value is set to "*True*", so into the http request is added parameter *TRANSPARENT=True*. In this case (if it is implemented), it returns images on the transparent background. Of course, the requested format must support transparent images (PNG, GIF). If you select a format that does not support transparency, request to the server can end with an error or by unexpected result.
- **ShowErrorInImage** – server displays error messages in the map window, see example in Fig 2.
- **TryTransformRemoteServer –** if for WMS source is set transformation between two different coordinate systems and if the server in the *GetCapabilities* list contain EPSG identical to the clients and if the item *TryTransformRemoteServer* is set to "*True*", the image will return already transformed from the server and Marushka won't have to recalculate it (it slows down the process and the resulting picture is also more distorted). When using this parameter, all the data store form layers are transformed, so if you need to transform some of the layers into different EPSG, you have to connect the WMS again as a new data store and set a different projection. This makes sense, especially when you publish the project as another WMS. In this case, the target coordinate system is not fixedly defined as in MarushkaHTML client, but it is defined in request as a parameter SRS (thus the server can assign transformation to a remote server).

B) Just version 1.3.0

- **TestWmsBboxBeforeRequest** – before evaluating the query is Bbox (limiting rectangle) compared with *GetCapabilities* data, because some WMS did after leaving their Bbox evoked complicated error messages.

4.2 Other Tips for Work with WMS

4.2.1 WMS multicolor form layer symbology recolor

If there is a case that you have a multi-colored layer connected via WMS source and you need to display it in only one color, use the following procedure:

1. You have a form layer *All REC 1.1.1*, which display the basic drawing in black, overview map in purple, respectively green and in the lower right corner is a gray logo ČÚZK 2010.
2. In the WMS source settings, in category *4. WMS properties*, set item *RequestTransparentImage* to "*True*", otherwise you should get a colored square.
3. For layer *All REC 1.1.1* select in the layer *Properties* category *2. Scale, drawing and symbology*, item *Symbology ~ Raster*. Set attribute *Brightness* to maximum (100), item *Constrast* to minimum (-100, as in the Fig 1.
4. This will prepare the form layer, so that except for transparent areas it is completely white, which is the best color for recolor (opposite *Brightness* adjustment would recolor drawing to black, but it is not possible to recolor black, because its value is 0 0 0 and if any operation is performed on a matrix with the vector of this color, the result will be always the same – black color).
5. Further, set attribute Colorize value RGB 255;255;0 by direct insert into box next to the attribute, respectively ARGB 255;255;255;0 after clicking on color palette.
6. Now the only action left is *Clear all* (it is necessary, because otherwise the original layer would be loaded instead), and then select *Data* / *Load view rectangle.*
7. WMS layer should now display in selected color.

4.2.2 Joining multiple form layers as one

If you need to load multiple form layers as one, create a new form layer in WMS data store. Insert a list of form layers you want to join into field *Name* and separate them with commas. This form layer does during communication with the server act as a layer loaded with *Merge* policy, i.e. for all the joined form layers is sent just one request on the server.

You can have a look at form layers joined like this in data store *Roads Add,* layer *Joined*. Try to load this form layer using function *Clear and load by policy* . Therefore, if you want to see the called request, you have to enable *Debug console* and function *Trace all*

This procedure is much more used than the *Merge* policy, for which it is good to know how it works. When using *Merge* policy, it is not possible to change the symbology of data, which is in many cases almost inevitable.