

Authorization Of Map Server



GEOVAP

CONTENTS

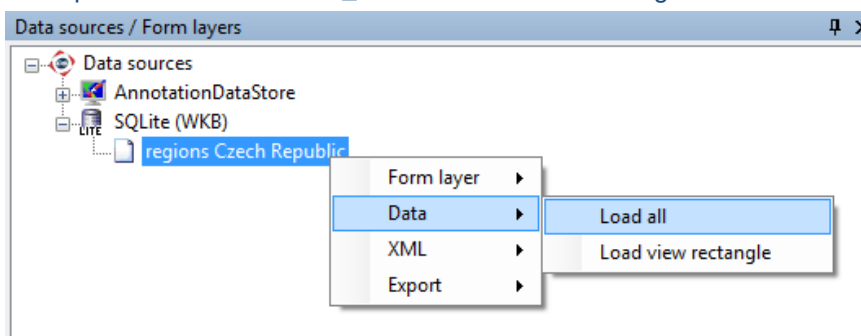
1	AIM OF THE EXAMPLE.....	2
2	WORKING WITH EXAMPLE	2
3	DIALOG BOX SAMPLE.....	3
4	A BRIEF DESCRIPTION OF THE EXAMPLE IN MARUSHKADESIGN	5

1 Aim of the Example

In this example we will demonstrate a few types of map server authorization. This example was created in version 4.1.0.1, so it does not have to be compatible with older versions.

2 Working with Example

- Unzip the **Authorization_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 project.
- Open the **Authorization_EN.xml** in MarushkaDesign environment.



- Select form layer *regions Czech Republic* in SQLite (WKB) data store, in the context menu choose Data – Load all:
- In map window choose “Fit All”:



- Edit or overwrite *Web.config* file, more [here](#).
- Launch the local web server:



- Enter login and password.

3 Dialog Box Sample

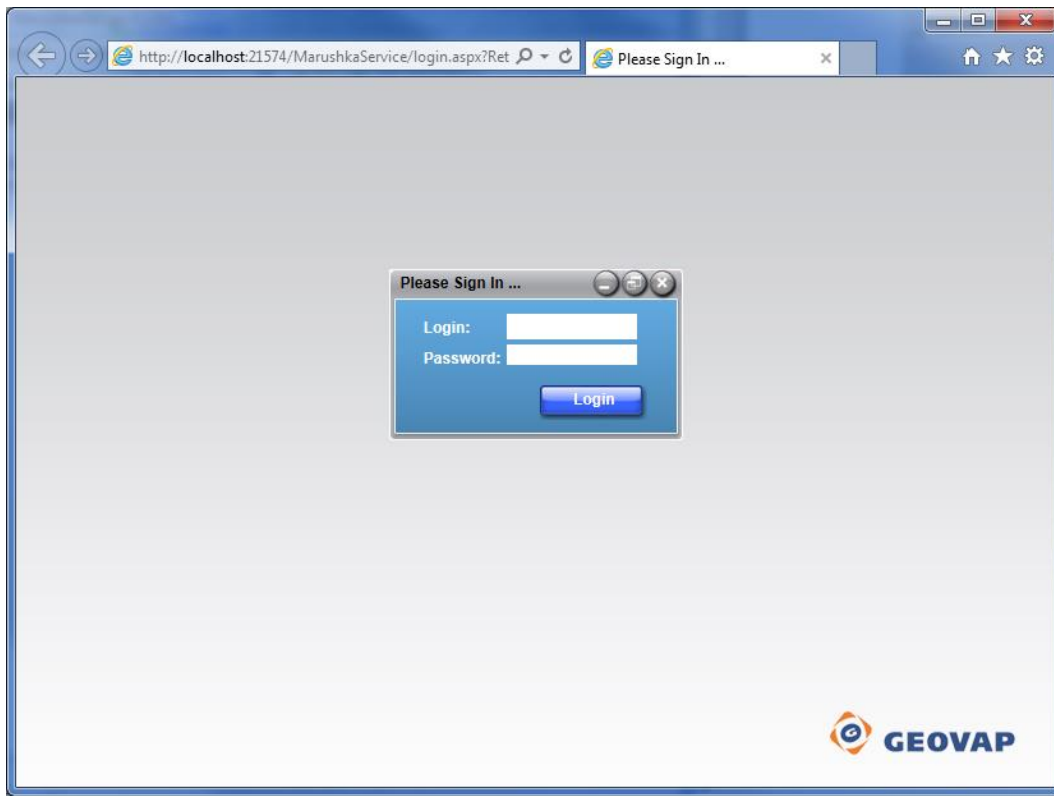


Fig 1: The login dialog in the local web server

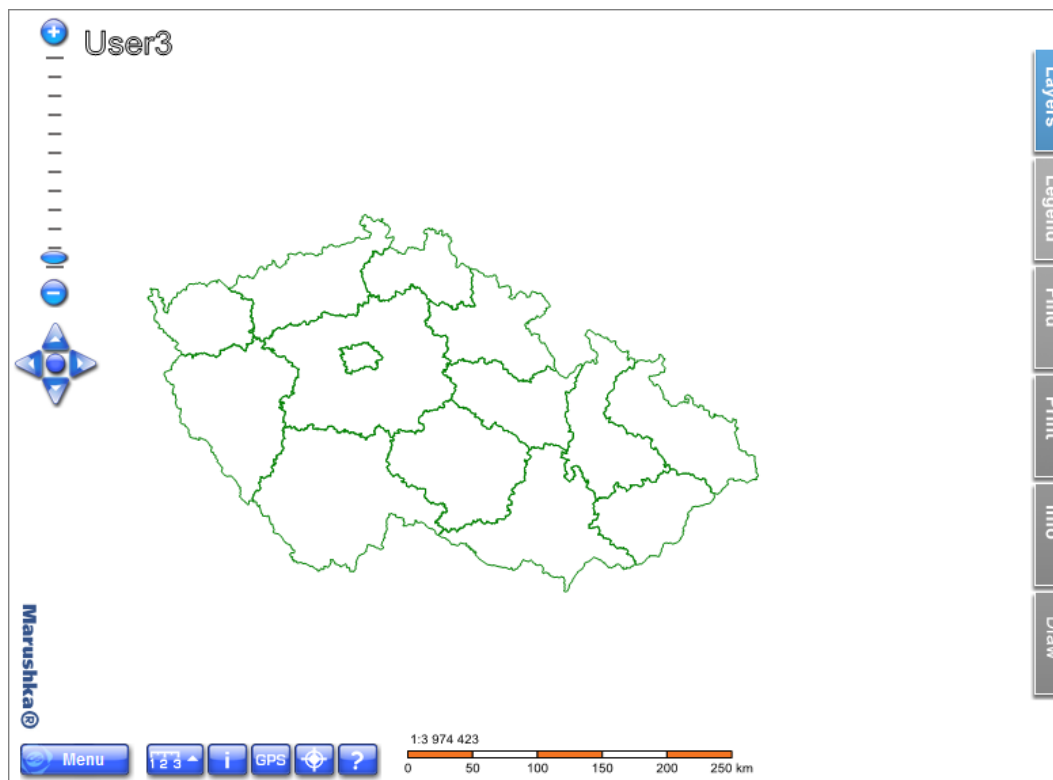


Fig 2: Sample of the query result in the local web server with logged on user User3

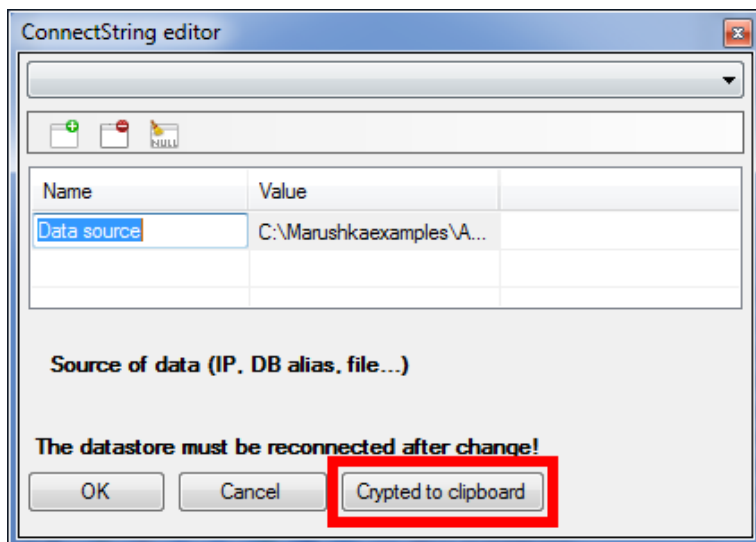


Fig 3: Copying data store ConnectString to clipboard

4 A Brief Description of the Example in MarushkaDesign

Example contains 4 *Web.config* files, each of which is preset to a different type of authorization. Before beginning work with this example it is recommended to back up the original *Web.config* file.

Caution! The *Web.config* file is not created until the first start of a Marushka local web server, so if you just installed Marushka, you should first run the local web server before trying this example!

Changing the type of authorization is done in a way, that you will either replace the original ***Web.config*** file (by default located in the path: "*C:/Program Files/Geovap/MarushkaDesign/4-1-0-1/service/*") by one of the preset files. Or you can hand edit this file. In this case, the bold version number in the path needs to be replaced by the number of the currently installed version of MarushkaDesign on your computer. The resulting file must be identical to the name and extension of the original file, i.e. *Web.config* and must be located in the same path.

Editing these files is possible e.g. in ***Free XML editor (Foxe)***, which is available at: http://www.firstobject.com/dn_editor.htm or using application ***Note Pad*** or many others.

When using one of the predefined *Web.config* files it is required for each version of Marushka other than 4.1.0.1 to replace the folder paths with which Marushka works ("*PathToRoot*", "*PathToLogFile*", "*PathToTemp*", "*LicenseDir*" and "*PathToThemeSettings*"), otherwise it would not work properly. This can be done e.g. by the program *Free firstobject XML editor* using the keyboard shortcut CTRL + H (or using Edit / Replace... in the main menu) that will replace all the paths to files. So by default you will replace "*C:\Program Files\Geovap\MarushkaDesign\4-1-0-1*" by the path to the current version folder, e.g. "*C:\Program Files\Geovap\MarushkaDesign\4-1-0-2*" and select *Replace All*.

If you are editing the *Web.config* file of the current Marushka version, it is not necessary to change these paths. This need would occur only when you install a new version of Marushka and then you continue modifying the *Web.config* file of an older version.

Authorization settings

The map publishing server allows several types of authorization access to the published project. Type of authorization, as already mentioned, is defined in the configuration file ***Web.config*** of the project.

1. Without Authorization


Public publication (i.e. accesses are not verified) is defined in the settings, so that in *Web.config* file in section `<appSettings>` set the following entry:

```
<!--Flag when authorization is in use-->
<add key="Authorization" value="false"/>
```

Next, in section `</system.web>` set:

```
<authorization>
  <allow users= "*" />
</authorization>
```

Finally, overwrite the original *Web.config* file by the modified version.

Then in MarushkaDesign choose *Data – Load all* and click on button  The local web server should start without any possibility or obligation of authorization.

2. Verifying Access to the Initial Form

Form authentication lies in the fact that the user and password are checked against a list of users defined in the `Web.config` file. Set the items as follows:

Allow authorization:

```
<!--Flag when authorization is in use-->
<add key="Authorization" value="true">
```

Set authorization type:

```
<!--Type of authorization component-->
<add key=AuthorizationComponent"
value="FrameWorkServer.Component.BasicFormAuthorization,FrameWorkServer"/>
<!--End Type of Authorization component-->
```


Set the list of users in the section:

```
<authentication mode="Forms">
  <forms loginUrl="login.aspx" name=".ASPXFORMSAUTH" timeout="10">
    <credentials passwordFormat="Clear">
      <user name="geovap" password="test/">
      <user name="test" password="1234"/>
      <user name="mike" password="test"/>
    </credentials>
  </forms>
</authentication>
```

In section `<system.web>` deny access to unauthorized users:

```
<authorization>
  <deny users="?" />
</authorization>
```

Finally, overwrite the original `Web.config` file by the modified version.

In `MarushkaDesign` then choose `Data – Load all` and click on button . It should start the local web server with a dialog box that will ask for Login and Password.

If you use the preset file, enter one of the pairs 1. Login: geovap, Password: test, 2. Login: test, Password: 1234, 3. Login: mike, Password: test.

If you followed instructions and edited `Web.config` file, you will have to use logins and passwords that you have defined yourself.

After entering a valid login and password your access will be authorized and the local web server will start.

3. Passwords Stored in Database Table

Form authentication using passwords stored in the table is performed so that the username and password are verified against the usernames and passwords stored in the database table.

In the `Web.config` file in section `<appSettings>` set the following items:

Allow authorization:

```
<!--Web page is (un)authorized-->
<add key="Authorization" value="true"/>
```

Set authorization component `FrameWorkServer` and item `UserSelect`:

```
<!--RequestBreak - Optional, break SQL statement, request is rejected.-->
<add key="AuthorizationComponent"
value="FrameWorkServer.Component.DataBaseUserListFormAuthorization,
FrameWorkServer"/>
<add key="UserSelect" value="SELECT COUNT(*) FROM USER_LIST WHERE
USER_NAME=~1~ AND USER_PASSWORD=~2~/>
```

Select the database type:

```
<!--add key="ServerBasePswd" value="ACC"/-->
<add key="ServerType" value="SQLITE"/>
```

Add a crypted database :

```
<
      add
      key
      ="AuthorizationConnect"
value="yZqUEED0Jy0bCp+J8zyYDNOKe57ncz4Fh9/UHKPBmN/kcstS0ZcAw00yM8YanG91K5u
35QodjLL00z4s06AMMw==" />
```


You can obtain this `ConnectionString`, so that in the data source in which are stored `UserNames` and `Passwords` click on *3. Connection Parameters ~ ConnectionString* and select the option *Crypted to clipboard*, as in *Img 3*.

Then enter copy this `ConnectionString` to `web.config` section above.

In section `<system.web>` deny access to unauthorized users:

```
<authorization>
  <deny users="?" />
```

Finally, overwrite the original *Web.config* file by the modified version.

In *MarushkaDesign* then choose *Data – Load all* and click on button . It should start the local web server with a dialog box that will ask for `Login` and `Password`.

If you used the preset file and if you did not modify the data in the table `USER_LIST`, enter one of the pairs: 1. `Login: User1, Password: 111`, 2. `Login: User2, Password: 222`, 3. `Login: User3, Password: 333`, 4. `Login: User4, Password: 444`, 5. `Login: User5, Password: 555`.

After entering a valid login and password, your access will be authorized and the local web server will start.

In versions older than 4.1.0.1 was used version of `web.config` (**passwords in table + AuthConnect**), which did not have encrypted database access. This `web.config` was for the backward compatibility also added into zip archive, but for safety reasons it is recommended to use a newer version.

4. Passwords Stored in Database Table, List of Themelds for Individual UserNames

Forms authentication using the list of allowed themes for individual users in the database table is performed so that the `UserName` and `Password` are validated against `UserNames` and `Passwords` stored in the table. For each `UserName` in this table is given a list of available `Themeids` for individual users, which is compared with current `Themeid` and `UserName`.

Setting of the `Web.config` file is the same as in the `Web.config` file (passwords in table), but there is in addition the following select:

```
<!--add key="RequestBreak" value="SELECT 100-COUNT(*) FROM REQUEST_USAGE
WHERE MACHINE_ID='~MACHINE_ID~' and request_time>sysdate - 1/1440"/-->
  <add key="UserAccessThemeSelect" value="select themeid from usertheme
where username = ~UserName~/>
```

If this select is not activated, the project would be verified only against the list of `UserNames` and `Passwords` as in the previous example *3. Passwords saved in the database table* and assigning

of Themes to available UserNames would not be performed. For each User would then be available all the existing themes.

Database USER_LIST.db contains tables USER_LIST and USERTHEME. Table USER_LIST includes a list of user names and passwords, table USERTHEME assigns available themes to individual users. For User1 are available Themeld 11 and 12, for User2 are available Themeld 13 and 14 and for User3 is available Themeld 15. Usernames, passwords and lists of Themelds for individual users can be changed in database USER_LIST.

When you define a list of Themes for individual users, it is necessary for each user to have assigned at least one Themeld! Otherwise, the user without a defined Themeld cannot log in to the project. Likewise, it is necessary to have a list of at least one user with a completed available THEMEID, otherwise the project would not start and it would return an error message: *Access was denied for the current theme.*

It is also necessary to set in Theme editor in parameter *LinksToTheme* a list of themes, among which you can switch; otherwise it would not be possible to call these themes.

In version older than 4.1.0.1 was used version of web.config (**themeid by user + AuthConnect**), which does not have encrypted database access. This web.config was for the backward compatibility also added into zip archive, but for safety reasons it is recommended to use a newer version.

5. Verifying Account against the Database Server

This type of authorization is described in the Manual.

6. NT authorization


This type of authorization is also described in the Manual.

- **Displaying the logged on user using SET_ENV_USER parameter**

In the case that you will need to substitute the user name into database queries, you can use SET_ENV_USER parameter, which is used to substitute the name of the current user. This parameter is dynamically replaced by the server therefore it cannot be changed from the outside. This parameter must be defined in PassedParams of the project.

In this example, the name of the logged on user is displayed using annotation layer in the top left map window next to the slider of the map scales.

- **Calling parameters in URL address in browser**

It is possible to run the project in Browser using button  You can then call the appropriate UserName, Password and Themeld directly in URL. When using this call method it is necessary to copy the local web server project address and insert it into the following URL. Depending on how many parameters you need to call, as many of them fill in the following URL, others are omitted. The resulting URL has the following form.

```
http://{local web server address}
/MarushkaService/default.aspx?ThemeId={theme id}&UserName={user
name}&Password={password}
```

The complete address calling Themeld number 15, UserName User3 and Password 333 has the following form:

```
http://localhost:21574/MarushkaService/default.aspx?ThemeId=15&UserName=User3&Password=333
```

You can read more about external call in Manual.