Application Note

Maps and Web-Cam Integration

In SpiderControl HMI

Muttenz,	April 2019
Autor:	Mike Keizer / Peter Brügger
Referenz:	
Datei:	AppliationNode_MapsIntegration.doc
Version:	1.0

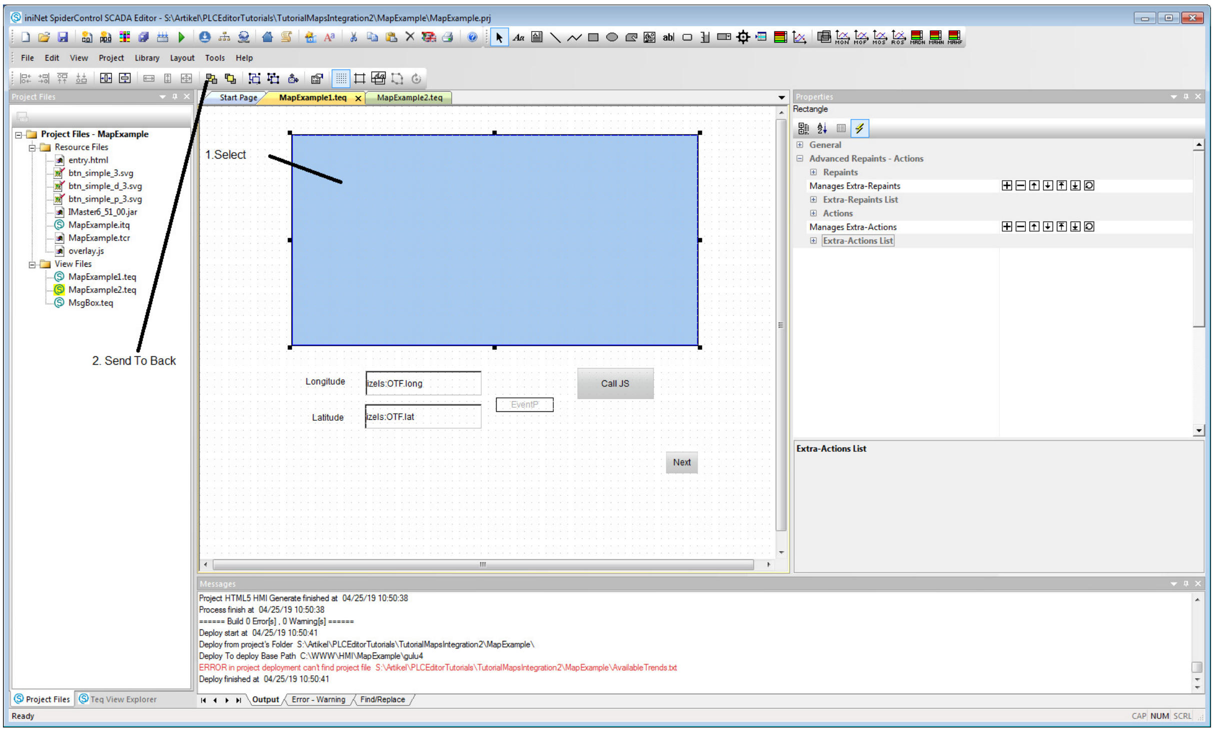
Contents

Overview

This application note describes how to integrate external HTML sources like maps or a webcam.
To integrate these services into a teq-view which is designed with the SpiderControl HMI editor, some modifications to the html and a js file need to be made manually.

1. HMI Project

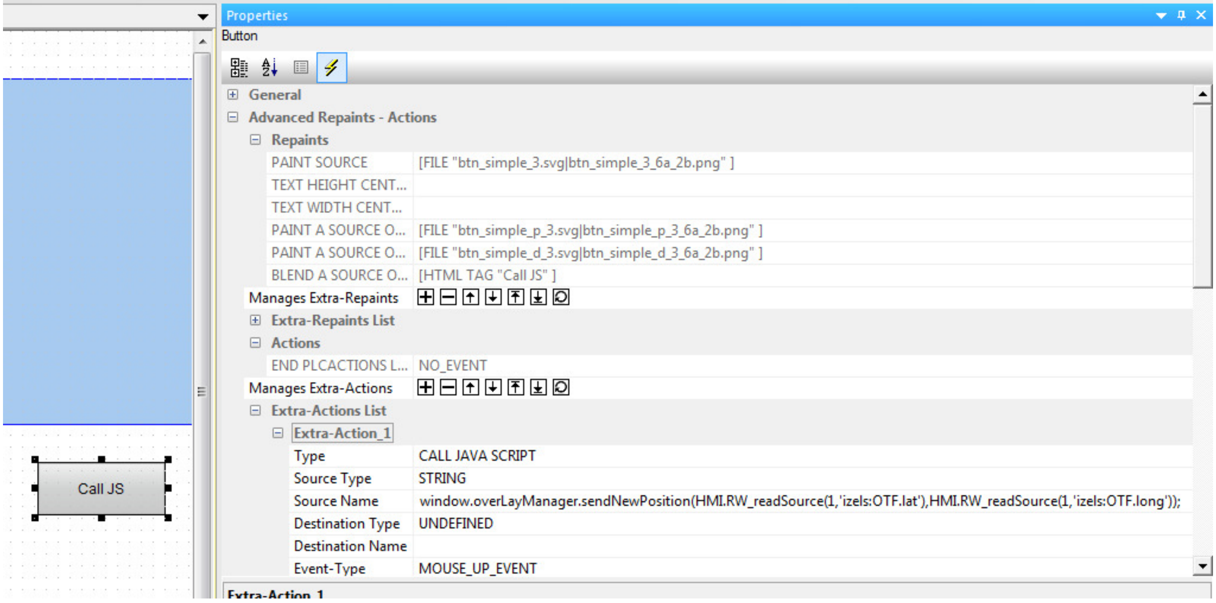
Open the project MapExample/MapExample.prj with the SpiderControl HMI Editor.



```
Project HTML5 HMI Generate finished at  04/25/19 10:50:38
Process finish at  04/25/19 10:50:38
====== Build 0 Error[s] , 0 Warning[s] ======
Deploy start at  04/25/19 10:50:41
Deploy from project's Folder  S:\Artikel\PLCEditorTutorials\TutorialMapsIntegration2\MapExample\
Deploy To deploy Base Path  C:\WWW\HMI\MapExample\gulu4
ERROR in project deployment can't find project file  S:\Artikel\PLCEditorTutorials\TutorialMapsIntegration2\MapExample\AvailableTrends.txt
Deploy finished at  04/25/19 10:50:41
```

The blue rectangle will be used as a marker to position the map-overlay.
Select this rectangle and send it to the background. The object in the background will be the first object in the file and can therefore be found by the overlay manager.

To re-position the map control during runtime, an external JavaScript (js) function can be called.
In this example, the position is taken from 2 PPOs ‘izels:OTF.lat’ and izels:OTF.long’.
The external js function is called from the button labeled ‘Call JS’. To do so, an extra action needs to be added, following the notation below:



General	
Advanced Repaints - Actions	
Repaints	
PAINT SOURCE	[FILE "btn_simple_3.svg|btn_simple_3_6a_2b.png"]
TEXT HEIGHT CENT...	
TEXT WIDTH CENT...	
PAINT A SOURCE O...	[FILE "btn_simple_p_3.svg|btn_simple_p_3_6a_2b.png"]
PAINT A SOURCE O...	[FILE "btn_simple_d_3.svg|btn_simple_d_3_6a_2b.png"]
BLEND A SOURCE O...	[HTML TAG "Call JS"]
Manages Extra-Repaints	
Extra-Repaints List	
Actions	
END PLCACTIONS L...	NO_EVENT
Manages Extra-Actions	
Extra-Actions List	
Extra-Action_1	
Type	CALL JAVA SCRIPT
Source Type	STRING
Source Name	window.overLayManager.sendNewPosition(HMI.RW_readSource(1,'izels:OTF.lat'),HMI.RW_readSource(1,'izels:OTF.long'));
Destination Type	UNDEFINED
Destination Name	
Event-Type	MOUSE_UP_EVENT

This action can be added to any object, e.g. also an event painter which would call this action on a condition.

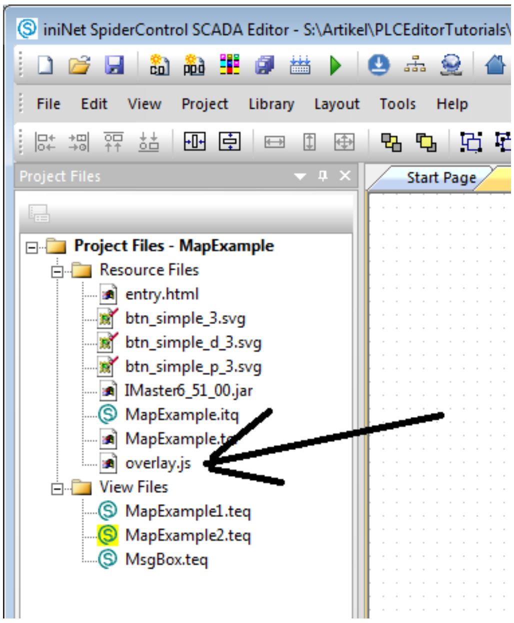
2. HTML Page

The HTML page (in this project: entry.html) which is generated by the HMI Editor needs to be modified by adding this line marked red (at line 48):

In “entry.html”:

```
<div class="SCViews" id="SC_MsgBox"></div>
<div class="SCViews" id="SC_View_001"></div>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="MapExample_itq.js"></script>
<script type="text/javascript" src="hmi_min.js"></script>
<script type="text/javascript" src="overlay.js"></script>
<script type="text/javascript">
```

The source file overlay.js has to be added to the resources of the project in order to deploy it to the target web-server:



2.1 AutomationBrowser

If you want to run this sample inside an AutomationBrowser, you need to remove the following section below the comment /* - DO NOT REMOVE… */ (in order to force the AutomationBrowser to use the HTML5 viewer instead of the MicroBrowser component):

```
/* - DO NOT REMOVE THIS COMMENT - */



</script>
</body>
</html>
```

The file entry.html should look like this in the end.

3. Overlay.js

This js file contains the code to control the overlay. The sources that you want to display must be indicated in this file. The demo project contains a link to a map service and a link to a public web-cam.
Theses sources can be modified here:

In file ‘overlay.js’ at line 221:

```
var createOverlayManager = function(){
       if(typeof overLayManager !== "undefined")
               return;
       overLayManager = new COverLayManager();
    overLayManager.Add({view: "MapExample1", attr: [ { src:"/OSM_JS/osm.html",
                                                           class: "map",
                                                           on:{ newPosition: newPosHandler }
                                                                                                }
                                                                                            ]});
    overLayManager.Add({view: "MapExample2", attr: [ { src:"/OSM_JS/osm.html",
                                                          class: "map",
                                                          on:{ newPosition: newPosHandler }
                                                                                                 },
                                                                                                 {
src:"https://webtv.feratel.com/webtv/?design=v3&cam=4181&c2=0",

class: "video"
                                                                                                 }
 ]});
```

Please mind that you need to indicate the name of the teq-page to which you want to link the overlay.
In this example, there is both a reference to page MapExample1.teq and MapExample2.teq.
If you want to apply the overlay manager to a different teq-page, you need to modify this section.

4. Sub Directory ‘OSM_JS’

You need to manually copy once the sub directory ‘OSM_JS’ to the deployment www folder of your Web-server, e.g.
‘www/HMI/OSM_JS’

This directory contains all the sources to display the maps (from open street map).
No changes need to be made here.

5. Remarks

Google Maps or a camera view. Not all web sites, how ever, allow inclusion in other websites by iframe. Yes, we are using an iframe. Some web sites for that reason set the **X-Frame-Options** to: „**sameorigin**“. For example Google Maps does this. To be able to display a Google Map in an HTML5 iframe the developer / company has to get a „key“ from Google. For this look here:
`https://developers.google.com/maps/documentation/javascript/get-api-key?hl=en`