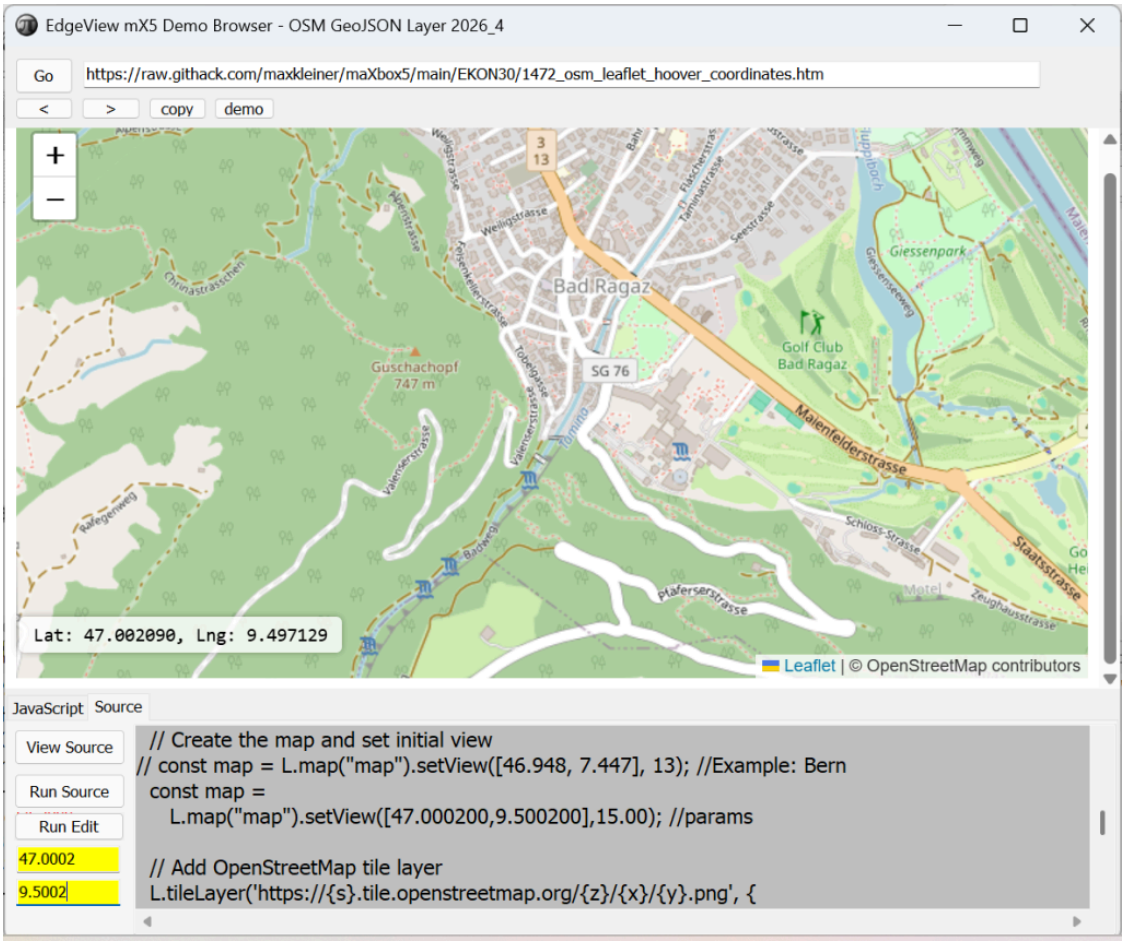
Guess Traffic Ways Norge Test

WebView Injection

This type of procedure function shows the injection of a JavaScript OSM leaflet library at runtime in a rich client of the EdgeView2 component. This will be the maXbox Starter Tutorial 192.

This component can embed web content (HTML, CSS, and JavaScript) in your native applications with Microsoft Edge WebView2 runtime dll.



TEdgeViewForm with geo coordinates parameters to inject in JavaScript

This procedure is the `OnClick` handler for a button (`Button1`) that lives on an `TEdit` control (`Edit1`) in Delphi. Its job is to take latitude/longitude/zoom values from the UI, build/update some HTML/JavaScript content for a map view, and then refresh that view.

The button on the form above is called Run Edit which passes the latitude and longitude to find and set on a OpenStreetMap:

```
procedure TEdit1Button1Click(Sender: TObject);
   //var Image1: TImage;
 var htmlgeojs2: string;
 begin
```

```
    writ('debug c: '+edt1lat.text+' '+edt2long.text)
    if SetSCRIPT_PARAMS(Format('%.6f',[strtofloat(edt1lat.text
                                    Format('%.6f',[strtofloat(
                                    format('%.2f',[zoomf])) th
       htmlgeojs2:= utf8encode(stringreplace((HOVER_JS_SCRIPT2
    edgeviewfrm.memoHTML.Text:= utf8toansi(htmlgeojs2); ////GE
    edgeviewfrm.btnSetSourceClick(self);
  end;
```

Step-by-step behavior

`writ('debc: '+edt1lat.text+' '+edt2long.text)` writes the current text of two edit boxes (`edt1lat`, `edt2long`) to some debug log/output. This is just for troubleshooting.

- Set script parameters

  It calls `SetSCRIPT_PARAMS` with three formatted strings:

  - Latitude: `Format('%.6f',[strtofloat(edt1lat.text)])`
  - Longitude: `Format('%.6f',[strtofloat(edt2long.text)])`
  - Zoom: `format('%.2f',[zoomf])`

    So it converts the latitude and longitude edit texts to `float`, formats them to 6 decimal places, formats the zoom to 2 decimals, and passes them into `SetSCRIPT_PARAMS`. The `if` means the following block only runs if that call returns `True`.

- Build HTML/JS string

  If `SetSCRIPT_PARAMS` succeeds:

  - It takes a constant/template string `HOVER_JS_SCRIPT2`.
  - Replaces all line feeds (`LF`) with carriage return + line feed (`CRLF`) using `stringreplace(..., LF, CRLF, [rfReplaceAll])`.
  - Wraps the result with `utf8encode`, assigning it to `htmlgeojs2`.

    This suggests `HOVER_JS_SCRIPT2` is UTF-8 text (likely HTML/JS for a map) that needs Windows-style line endings.

- Push HTML into a memo and refresh the view

  - `edgeviewfrm.memoHTML.Text := utf8toansi(htmlgeojs2);`

    Converts the UTF-8 string to ANSI and puts it into `memoHTML` on a form called `edgeviewfrm`. That memo presumably holds the HTML source used by some embedded browser or viewer.
  - `edgeviewfrm.btnSetSourceClick(self);`

    Programmatically triggers the `OnClick` handler of `btnSetSource` on `edgeviewfrm`. That handler likely tells the browser control to (re)load the HTML from `memoHTML`, thus updating the displayed map with the new lat/lon/zoom data.

So originally this button may have been used to load an image file, but that functionality is disabled now; the procedure currently only updates the map view based on the coordinate/zoom inputs for OpenStreetMap.

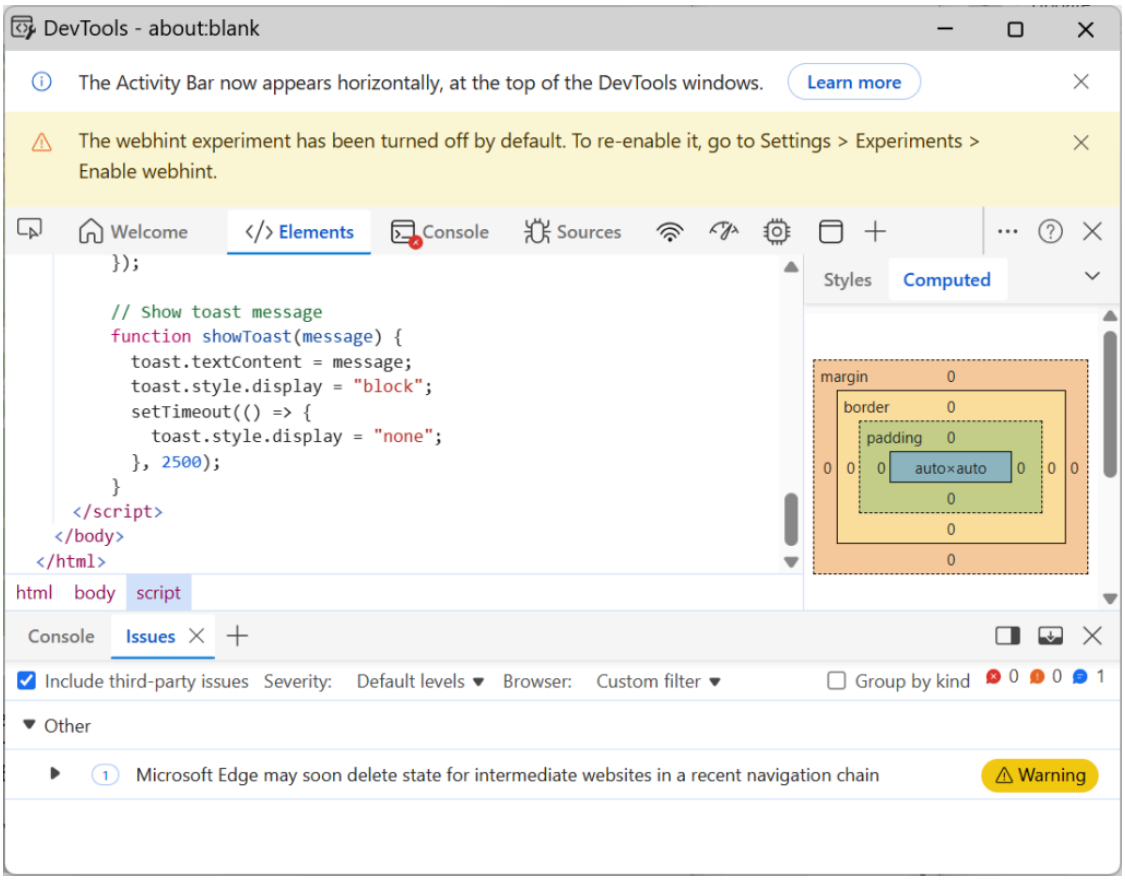
The embeed JScript is a function called SetSCRIPT_PARAMS() which injects the string html-template HOVER_JS_SCRIPT2. which returns the updated script at runtime:

```
function SetSCRIPT_PARAMS(vlat,vlong,vzoom: string): boolean
begin
 result:= false;
 HOVER_JS_SCRIPT2:=
'<!DOCTYPE html>
'<html>
'<head>
'  <meta charset="utf-8" />
'  <title>Leaflet OSM Hover Coordinates on maXbox52</title>
'  <meta name="viewport" content="width=device-width, initial
'
'  <!-- Leaflet CSS -->
'  <link rel="stylesheet" href="https://unpkg.com/leaflet/dis
'
'  <style>
'    #map {
'      height: 100vh; /* Full screen height */
'      width: 100%;
'    }
```

The adapted jscript will then executed with edgeviewfrm.btnSetSourceClick(self);

What is the Inspect tool?

Inspect is a tool that allows you to view HTML and CSS elements on a webpage. The HTML or JavaScript code creates animation, and CSS handles web design. It can help you understand how a webpage is constructed, what font a site may be using, see what elements of a website are broken, and how long a site takes to load. You can also use Inspect to see how temporary changes to the code may look on your device.



The Inspect tool can be started with right hand click on the webview map.

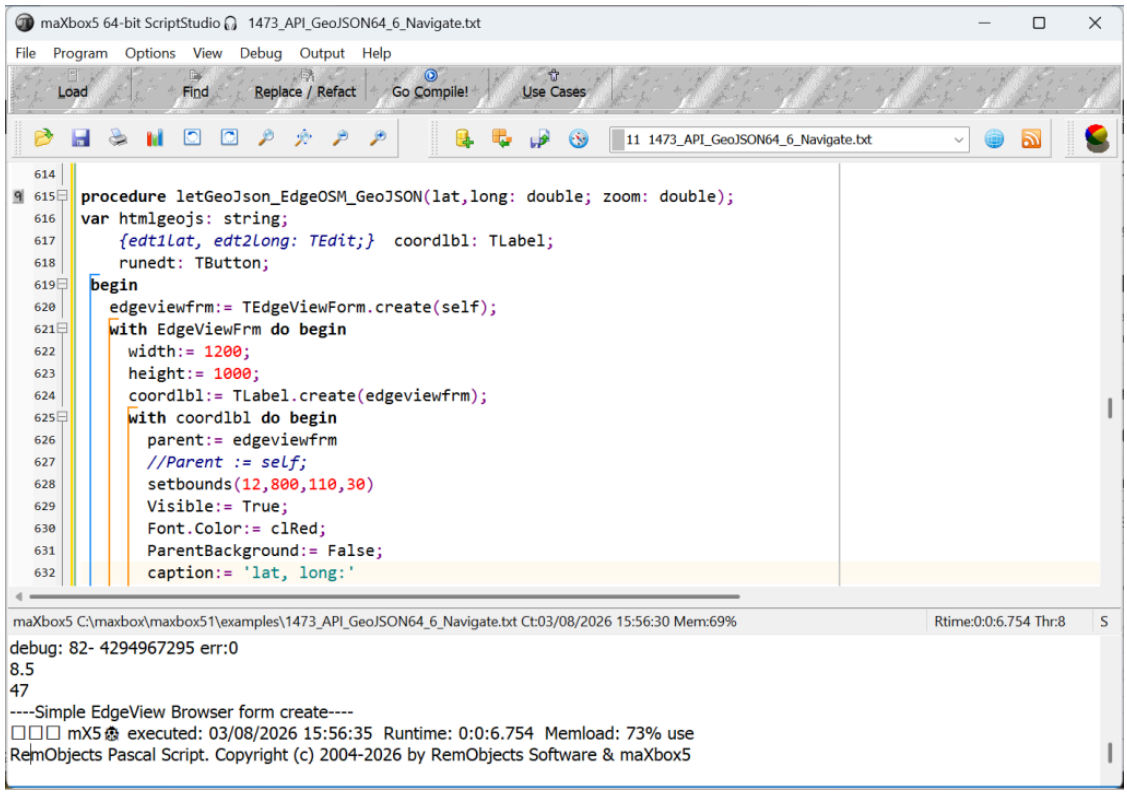
Conclusion

`TEdit1Button1Click` reads lat/lon/zoom from edit controls, passes them to a script-parameter function, regenerates an HTML/JavaScript snippet for a map viewer, puts it into a memo on another form, and then forces that viewer to reload the content in maXbox script engine.

The script can be found at:

Download 1473_API_GeoJSON64_6_Navigate.txt (maXbox5)

JavaScript injection is a technique where a user inserts their own JavaScript code into a web page. This can be done for various purposes, such as debugging, testing, or even malicious activities. In our example the script runs on a client side web form which isolates or delegates server-side scripting to a desktop rich client. There's no evidence that WebView2 Runtime is categorically more secure than Edge. It shares the same core engine and receives similar engine-level patches, but it lacks some browser-specific protections and depends on app developers to use it safely.



https://sourceforge.net/projects/maxbox5/files/examples/1473_API_GeoJSON64_6_Navigate.txt/download

Download 1473_API_GeoJSON64_6_Navigate.txt (maXbox5)

To get more experiences with geo mapping I can recommend the following site:

GPS Coordinates, Latitude and Longitude with Interactive Maps

4 responses to “EKON 30”

breitsch breitsch
November 23, 2025 Edit

Welcome to EKON 30 in 2026

★ Like

Reply