

# MapInfo2Leaflet

---

## User Guide

TWIAV.NL

## Table of Contents

1. Introduction .....	2
2. Export a MapInfo Table to a Web Page using Leaflet.....	3
2.1. Step 1: Select Table to Export .....	3
2.2. Step 2: Set Layer Settings.....	4
2.3. Step 3: Set Title for Web Page .....	5
2.4. Step 4: Set html file .....	6
2.5. Step 5: Set Style Settings.....	6
2.6. Step 6: Click OK to Export .....	7
3. The web page – technical details.....	9
3.1. Leaflet 0.7.3 .....	9
3.2. Background map: OpenStreetMap .....	9
3.3. Projection: EPSG:4326 (WGS84) .....	9
3.4. Styling.....	9
4. Installation .....	10
4.1. MapInfo2Leaflet32Verion02.zip .....	10
4.2. Run MapInfo2Leaflet: MapInfo2Leaflet32.MBX.....	10
4.3. Test data.....	11
4.4. Unload the tool .....	11
5. Sample Output .....	12
5.1. html file .....	12
5.2. Javascript file.....	14

## 1. Introduction

With MapInfo2Leaflet you can export a MapInfo table to a web page using Leaflet - the output consists of an html file, a js file (containing a GeoJSON FeatureCollection) and a css file for the styling.

Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps.

More information: <http://leafletjs.com/>

This version is compatible with the 'classic' 32 bit version of MapInfo Professional (9.5 or higher)

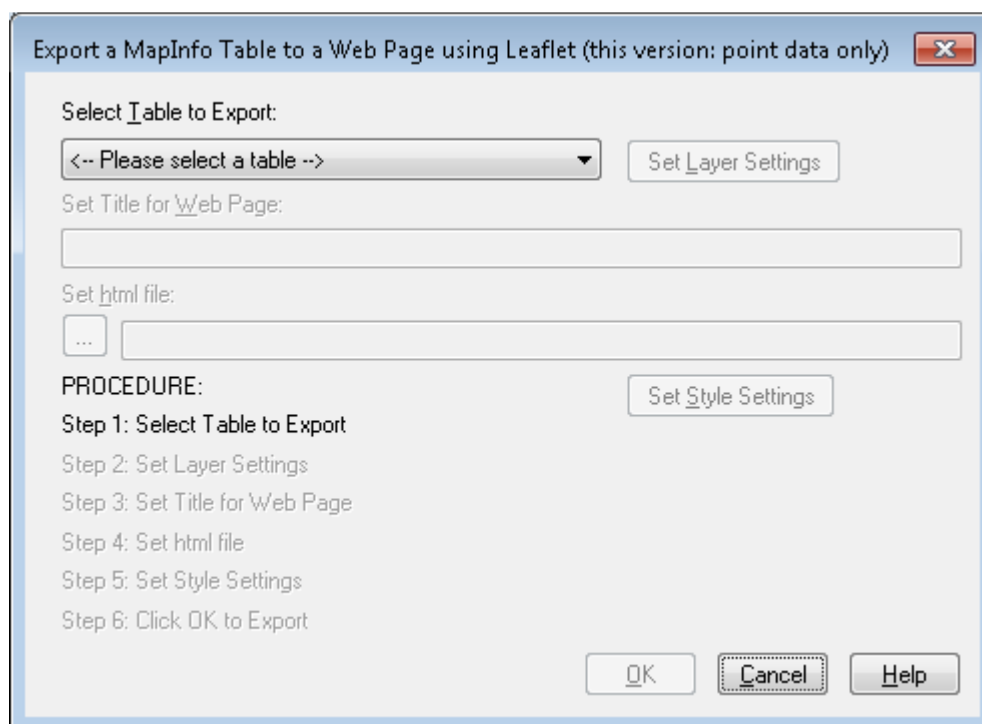
## 2. Export a MapInfo Table to a Web Page using Leaflet

In MapInfo Professional open the table you want to export to a web page. (This version of MapInfo2Leaflet only supports point data.)

Next go to **MapInfo2Leaflet > MapInfo2Leaflet** or click the **MapInfo2Leaflet** button

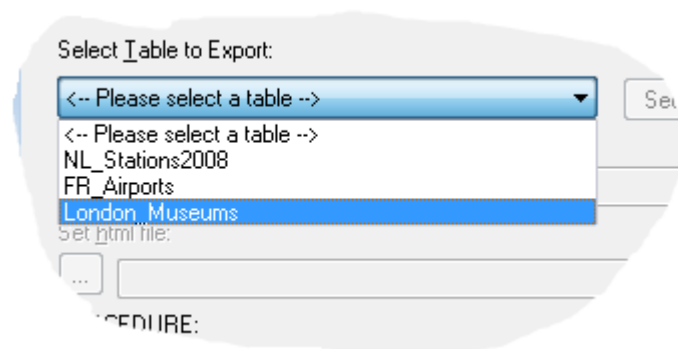


MapInfo2Leaflet

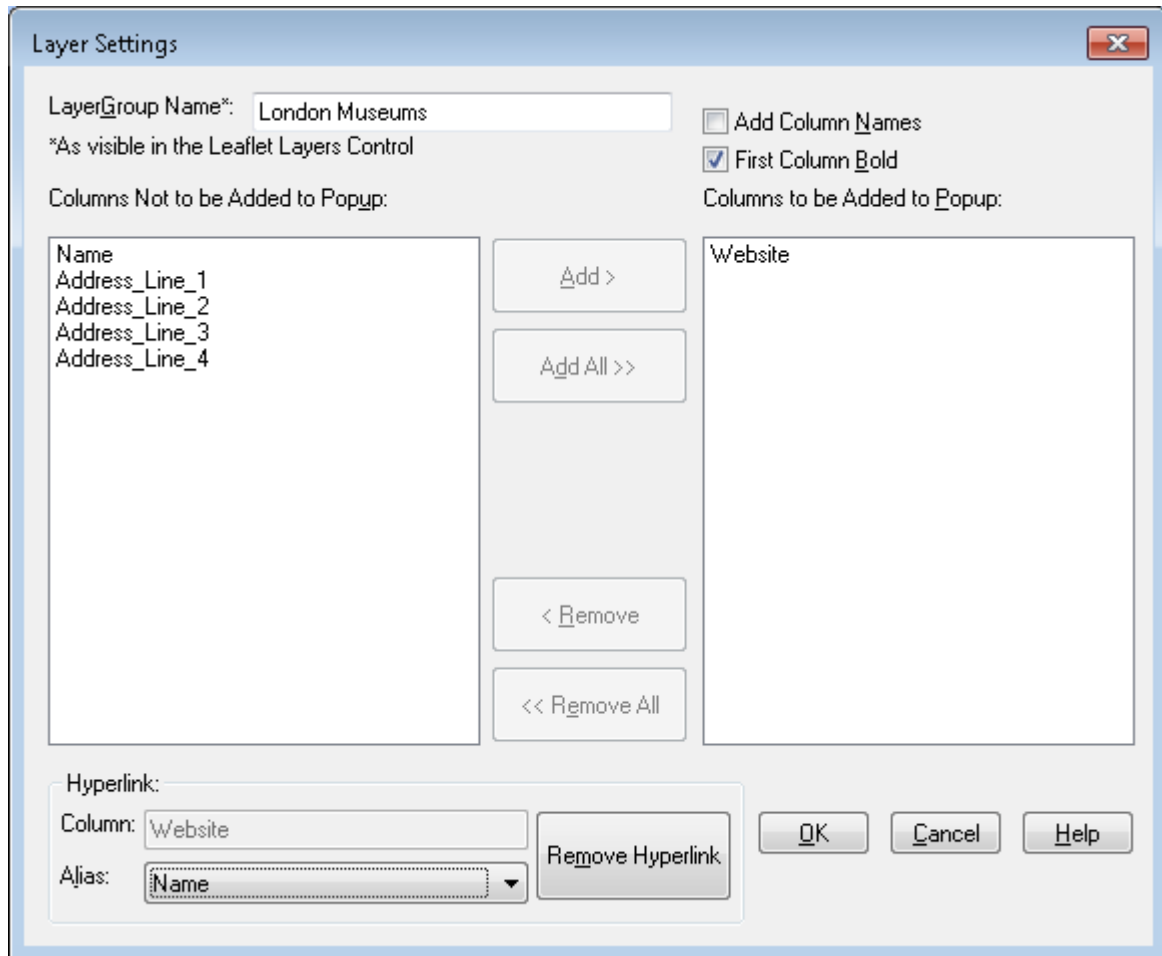


### 2.1. Step 1: Select Table to Export

Select your table from the available tables, e.g. **London\_Museums**. This will activate the **Set Layer Settings** button.



## 2.2. Step 2: Set Layer Settings



The screenshot shows the 'Layer Settings' dialog box. At the top, the title bar says 'Layer Settings'. Below it, there's a text field for 'LayerGroup Name\*' containing 'London Museums'. To its right are two checkboxes: 'Add Column Names' (unchecked) and 'First Column Bold' (checked). Below the name field is a note: '\*As visible in the Leaflet Layers Control'. Underneath, there are two lists. The left list, titled 'Columns Not to be Added to Popup:', contains 'Name', 'Address\_Line\_1', 'Address\_Line\_2', 'Address\_Line\_3', and 'Address\_Line\_4'. The right list, titled 'Columns to be Added to Popup:', contains 'Website'. Between these lists are four buttons: 'Add >', 'Add All >>', '< Remove', and '<< Remove All'. At the bottom left, there's a 'Hyperlink:' section with a 'Column:' dropdown set to 'Website', an 'Alias:' dropdown set to 'Name', and a 'Remove Hyperlink' button. At the bottom right are 'OK', 'Cancel', and 'Help' buttons.

When you click the **Set Layer Settings** button, the **Layer Settings** dialog will appear.

In this dialog you can set the following options:

- **LayerGroup Name**

This name will be shown in the **Leaflet Layers Control** on your web page, e.g. **London Museums**.



- **Columns to be Added to Popup**

This defaults to all columns in the table, but you can **Remove** columns from the list. To reorder the columns in the popup: **Remove All**, and manually **Add** in the right order.



- **Add Column Names**

When you check this option, column names will also be shown in the popup

- **First Column Bold**

When you check this option, the first column value will be shown bold in the popup

- **Hyperlink**

When processing the table **MapInfo2Leaflet** will automatically detect whether the table does contain a column with URLs, i.e. links to web pages. To be detected these URL values should start with "http".

If a column with URLs is present a hyperlink will be created automatically.

You can set an **Alias** (the value of one of the other columns).

Click **OK** to return to the **Export** dialog

## 2.3. Step 3: Set Title for Web Page

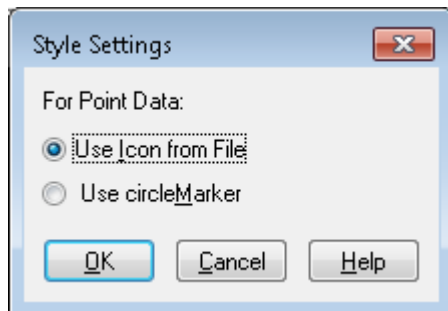
You can modify the title of the web page

## 2.4. Step 4: Set html file



You can set the name and the output folder for the html file to be created. The other output files (the css and js files) will be created in or copied to the same folder.

## 2.5. Step 5: Set Style Settings

When you click the **Set Style Settings** button, the **Style Settings** dialog will appear.



With the current version of **MapInfo2Leaflet** you can only export point data. For these points you can choose between two styles:

- **Use Icon from File** 
- **Use circleMarker** 

At the moment both options come with a default value, a blue arrow and a green dot with a red border respectively.

You can edit the html file after it has been created to modify the point style, i.e. use your own icon or change the colours of the circle.

To use another **Icon from File** change or replace the variable *blueArrowIcon*:

```
var blueArrowIcon = L.icon({
  iconUrl: 'blue-arrow.png',
  iconSize: [32, 37],
  iconAnchor: [16, 37],
  popupAnchor: [0, -28]
});
```

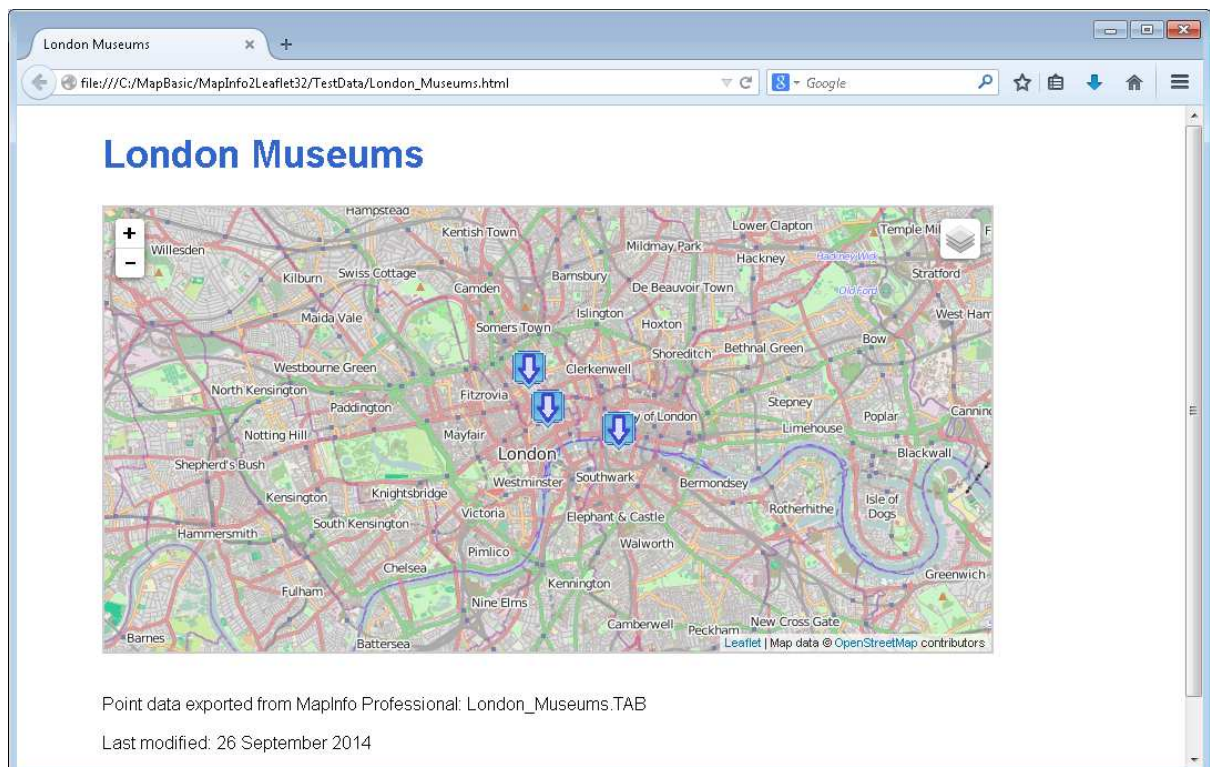
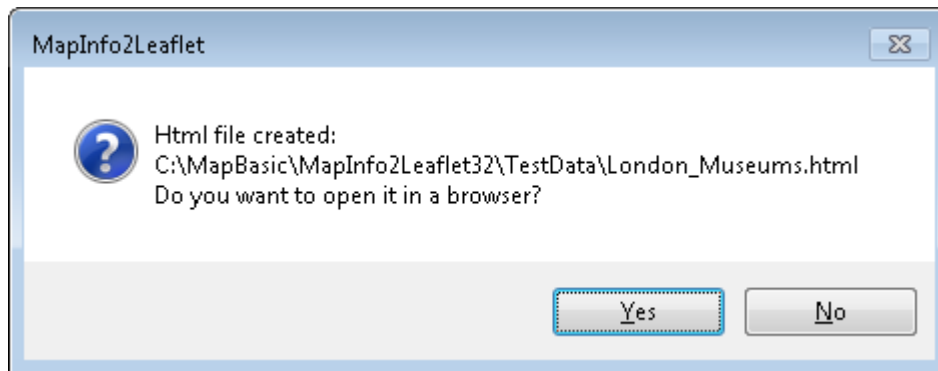
**Please note: do not remove or rename the file *blue-arrow.png* in the application folder because the application needs this file to create the web page. Only replace the \*.png file in the destination folder.**

To modify the appearance of the **circleMarker**, change the values in the variable *mapinfoMarkerOptions*:

```
var mapinfoMarkerOptions = {  
    radius : 6,  
    fillColor : "#008000",  
    color : "#FF0000",  
    weight: 2,  
    opacity: 1,  
    fillOpacity : 0.9  
};
```

## 2.6. Step 6: Click OK to Export

When all settings are set you can click **OK** to create the web page.





The output consists of 3 (or 4) files:

- An \*.html file, e.g. **London\_Museums.html**. This file contains the basic layout of the website – see chapter Sample Output for an example.
- A \*.js file, e.g. **London\_Museums.js**. This file contains the geometry, i.e. the point data (a GeoJSON FeatureCollection) – see chapter Sample Output for an example.
- A cascading style sheet (css) file: **default.css**. This file contains some basic style settings for the web page.
- The icon file **blue-arrow.png** (optional)

## 3. The web page – technical details

### 3.1. Leaflet 0.7.3

In the web page we use Leaflet (<http://leafletjs.com/>).

Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps.

The website uses the version of Leaflet hosted on the Leaflet CDN (*content delivery network*):

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-
0.7.3/leaflet.css" />
```

### 3.2. Background map: OpenStreetMap

The background map is OpenStreetMap:

```
var osmAttrib='Map data © <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
osmUrl='http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png';
```

### 3.3. Projection: EPSG:4326 (WGS84)

The projection used in the map is EPSG:4326.

Please note: your input MapInfo Table can be in any projection. **MapInfo2Leaflet** will extract the coordinates in WGS84 (EPSG:4326), independent of the projection of the input table.

### 3.4. Styling

Some basic styling of the web page is provided by a cascading style sheet (css) file: **default.css**. You can modify (or replace) this style sheet to suite your needs.

**Please note:** the style sheet also contains the styling of the `<div id="map"></div>`:

```
#map {
  width: 800px;
  height: 400px;
  border: 2px solid #CCCCCC;
  margin-top: 25px;
}
```

You can modify this, but please do not remove it (otherwise the map will not show in the web page).

## 4. Installation

### 4.1. MapInfo2Leaflet32Verion02.zip

To install **MapInfo2Leaflet version 0.2** unzip **MapInfo2Leaflet32Verion02.zip** in a folder of your choice.

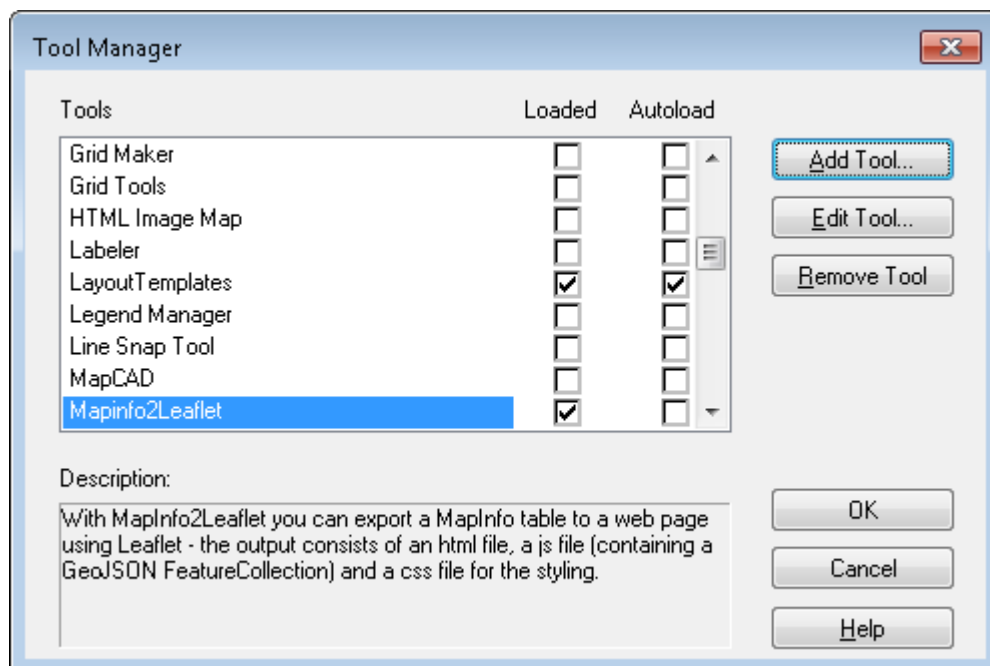
The following files and folders will appear:

29/09/2014	11:22	<DIR>	Documentation
29/09/2014	11:24	<DIR>	TestData
26/09/2014	09:17	941	blue-arrow.png
26/09/2014	12:17	459	default.css
26/09/2014	13:50	5,751	MapInfo2Leaflet32.def
26/09/2014	14:19	39,474	MapInfo2Leaflet32.mb
26/09/2014	14:20	28,686	MapInfo2Leaflet32.MBX
25/09/2014	18:19	4,530	MBExtensions.cs
25/09/2014	18:19	5,120	MBExtensions.dll
18/09/2014	06:25	13,824	MIPro2LFLTIcons.dll

### 4.2. Run MapInfo2Leaflet: MapInfo2Leaflet32.MBX

To run **MapInfo2Leaflet** double-click **MapInfo2Leaflet32.MBX**

Optionally you can add the tool to the Tool Manager.



### 4.3. Test data

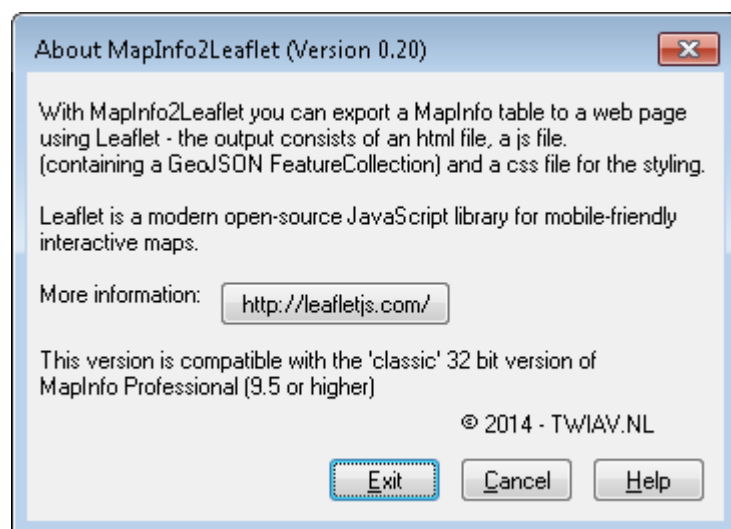
The two tables which have been used during development and testing are included.

**London\_Museums.TAB** and **FR\_Airports.TAB**. Both tables are in a local projection, British National Grid and French Lambert-93, respectively.

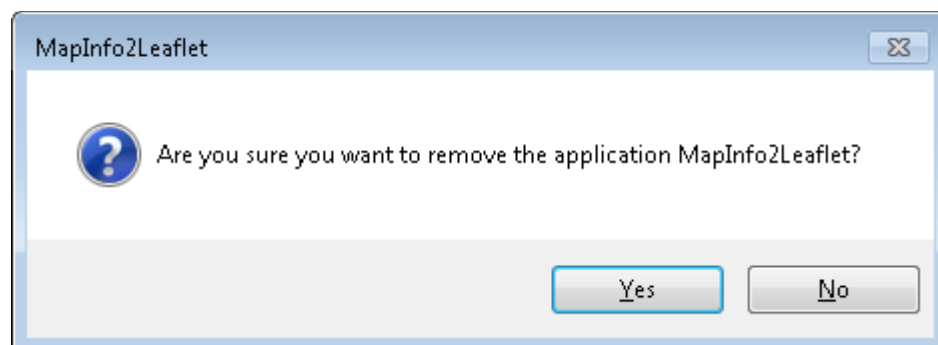
17/09/2014	17:03	16,204	FR_Airports.DAT
17/09/2014	17:01	456	FR_Airports.ID
17/09/2014	17:01	6,656	FR_Airports.MAP
17/09/2014	17:03	385	FR_Airports.TAB
25/09/2014	19:02	1,429	London_Museums.DAT
24/09/2014	12:47	12	London_Museums.ID
24/09/2014	12:47	2,048	London_Museums.MAP
25/09/2014	19:02	426	London_Museums.TAB

### 4.4. Unload the tool

To unload the tool from MapInfo Professional, go to **MapInfo2Leaflet > About MapInfo2Leaflet**



In the **About MapInfo2Leaflet** dialog, click the **Exit** button.



## 5. Sample Output

### 5.1. html file

C:\MapBasic\MapInfo2Leaflet32\TestData\London\_Museums.html

```
<!DOCTYPE html>
<html>
<head>
  <title>London Museums</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-
0.7.3/leaflet.css" />
  <link rel="stylesheet" href="default.css" />
</head>
<body>
  <header>
    <h1>London Museums</h1>
  </header>
  <div id="main">
    <div id="map"></div>
    <br>
    <p>Point data exported from MapInfo Professional:
London_Museums.TAB</p>
    <p>Last modified: 26 September 2014</p>
  </div>
  <script src="London_Museums.js" type="text/javascript"></script>
  <script src="http://cdn.leafletjs.com/leaflet-
0.7.3/leaflet.js"></script>
  <script>
    var map = L.map('map').setView([51.5119, -0.121304], 12);

    var osmAttrib='Map data © <a
href="http://openstreetmap.org">OpenStreetMap</a> contributors',
    osmUrl='http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png';

    var osm = L.tileLayer(osmUrl, {attribution: osmAttrib}).addTo(map);

    function onEachFeature(feature, layer) {
      if (feature.properties && feature.properties.popupContent) {
        layer.bindPopup(feature.properties.popupContent);
      }
    }

    var blueArrowIcon = L.icon({
      iconUrl: 'blue-arrow.png',
      iconSize: [32, 37],
      iconAnchor: [16, 37],
      popupAnchor: [0, -28]
    });

    var london_museumsLayer = L.geoJson(london_museums, {
      pointToLayer: function (feature, latlng) {
        return L.marker(latlng, {icon: blueArrowIcon});
      },
      onEachFeature: onEachFeature
    }).addTo(map);

    var baseLayers = {
      "OpenStreetMap": osm
```

```
};

var overlays = {
  "London Museums": london_museumsLayer
};

L.control.layers(baseLayers, overlays).addTo(map);
</script>
</body>
</html>
```

## 5.2. Javascript file

C:\MapBasic\MapInfo2Leaflet32\TestData\London\_Museums.js

```
//Point data exported from MapInfo Professional: London_Museums.TAB
//Last modified: 26 September 2014
var london_museums = {
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "popupContent": "<b>Website: <a
href=\"http://www.britishmuseum.org/\">British Museum</a></b>"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [-0.126981, 51.5195]
      }
    },
    {
      "type": "Feature",
      "properties": {
        "popupContent": "<b>Website: <a
href=\"http://www.tate.org.uk/visit/tate-modern\">Tate Modern</a></b>"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [-0.099381, 51.5076]
      }
    },
    {
      "type": "Feature",
      "properties": {
        "popupContent": "<b>Website: Unknown</b>"
      },
      "geometry": {
        "type": "Point",
        "coordinates": [-0.121304, 51.5119]
      }
    }
  ]
};
```