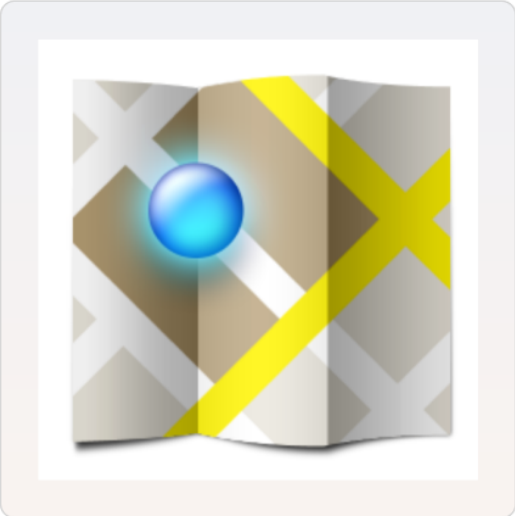
MapsGMap Control Reference

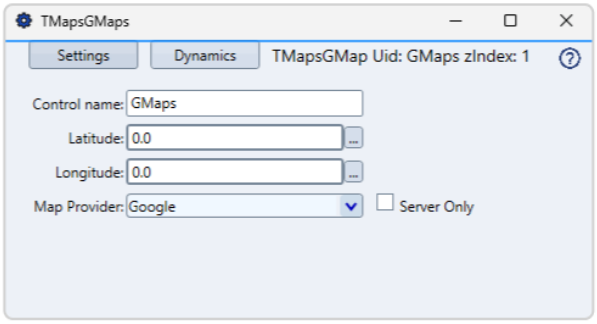
Show a map by a provider like Google Maps or Bing Maps.

Reference **Controls** **Viewer** MapsGMap

The MapsGMap external component is used to create GIS (Geographical Information System) projects. This need is a common requirement in projects with assets geo-location.

Requirements
Configuration
Code Behind Callback (Only for Advanced Customization)
Runtime Execution
Related





Requirements

To use this feature, the following system requirements need to be matched:

- Product version v8.1.39 or newer.

Configuration

The GMapControl is a simple window with a map that uses coordinates and gets map information by a provider, like Google Maps or Bing Maps.

1. Go to **Displays / Draw**.
2. Select **Viewer**, then **MapsGMap**.
3. Add it to the Draw area by clicking or dragging and dropping it.
4. Double-click the object to open the setting window.

GMap configuration	
Field	**Description**
Control Name	Defines a name for the control so it can be accessed in the CodeBehind script. See Display CodeBehind.
Latitude	The latitude of the coordinates.
Longitude	The longitude of the coordinates.

Map Provider	Select the correct provider: • Bing Maps • Cloud Made • Google • Open Cycle • WikiMapia • Yahoo Maps
Server Only	Enable GMap working mode to use only server data.

Code Behind Callback (Only for Advanced Customization)

First, import all GMap proprieties and set some variables.

```
*        C#:
using GMap.NET;
using GMap.NET.MapProviders;
using GMap.NET.WindowsForms;
using GMap.NET.WindowsForms.Markers;
using GMap.NET.Internals;

double end_zoom;
TGMapControl mapControl = null;
GMapControl gmap = null;
GMarkerGoogle marker = null;
```

Then, configure the mouse and other settings.



All these codes are in method DisplayOpening () {} and in C#.

```
mapControl = this.CurrentDisplay.GetControl("TGMapControl") as TGMapControl;
gmap = mapControl.GetControlRef();
gmap.DragButton = System.Windows.Forms.MouseButtons.Left;
gmap.CanDragMap = true;
gmap.MarkersEnabled = true;
gmap.Zoom = 11;
end_zoom = gmap.Zoom;
gmap.MaxZoom = 22;
gmap.MinZoom = 1;
gmap.AutoScroll = true;

// Choose the provider:

gmap.MapProvider = GMap.NET.MapProviders.BingMapProvider.Instance;
```

Set the position from the Map and configure the Markers.

```
//Set longitude and latitude

gmap.Position = new GMap.NET.PointLatLng(-15.793888, -47.883487);

//Creating Markers

gmap.Overlays.Add(new GMapOverlay("Markers"));
gmap.Overlays[0].Markers.Clear();
gmap.ReloadMap();

addMarker(new PointLatLng(-15.7118595,-47.9107536), GMarkerGoogleType.purple_pushpin,
"DC", String.Format("{0}\n{1}", "DC", "Digital City"));

addMarker(new PointLatLng(-15.81164,-47.950911), GMarkerGoogleType.purple_pushpin,
"B", String.Format("{0}\n{1}", "B", "Brasilia"));

addMarker(new PointLatLng(-15.7837743,-47.8957009), GMarkerGoogleType.purple_pushpin,
"NS", String.Format("{0}\n{1}", "EN", "Nacional Stadium"));

addMarker(new PointLatLng(-15.7507488,-47.9370287), GMarkerGoogleType.purple_pushpin,
"NB", String.Format("{0}\n{1}", "NB", "North Brasilia"));
```

To connect one point to another in the map, use the code below:

```
GMapOverlay routes = new GMapOverlay("routes");
List<PointLatLng> points = new List<PointLatLng>();
points.Add(new PointLatLng(-15.712199, -47.911240));
points.Add(new PointLatLng(-15.722334, -47.910331));
points.Add(new PointLatLng(-15.723553, -47.909480));
points.Add(new PointLatLng(-15.724644, -47.907717));
points.Add(new PointLatLng(-15.7411402, -47.9177745));
points.Add(new PointLatLng(-15.756816, -47.927832));
points.Add(new PointLatLng(-15.754628, -47.932072));
points.Add(new PointLatLng(-15.755097, -47.933931));
points.Add(new PointLatLng(-15.752753, -47.934598));
points.Add(new PointLatLng(-15.7507488,-47.9370287));

GMapRoute route = new GMapRoute(points, "DC_NB");
route.Stroke = new System.Drawing.Pen(System.Drawing.Color.Green, 3) ;
routes.Routes.Add(route);
gmap.Overlays.Add(routes);

points.Clear();
```

To use small markers in the map without a Big Marker, use this code:

```
addMarker(new PointLatLng(-15.772571, -47.937260), GMarkerGoogleType.green_small, "RE_Y",
String.Format("{0}\n{1}", "RE_Y", "Circuit: yyyy"));

addMarker(new PointLatLng(-15.775202, -47.938166), GMarkerGoogleType.green_small, "RE_z",
String.Format("{0}\n{1}", "RE_z", "Circuit: zzzz"));
```

Finally, implement the method *addMarker() {}*

```
public void addMarker(PointLatLng position, GMarkerGoogleType tipoPin, string tag, string tooltipText)
{
        marker = new GMarkerGoogle(position, tipoPin);

        marker.Tag = tag;
        marker.ToolTipText = tooltipText;
        marker.ToolTip.Fill = System.Drawing.Brushes.Blue;
        marker.ToolTip.Foreground = System.Drawing.Brushes.White;
        marker.ToolTip.Stroke = System.Drawing.Pens.Black;
        marker.ToolTip.TextPadding = new System.Drawing.Size(20,
20);

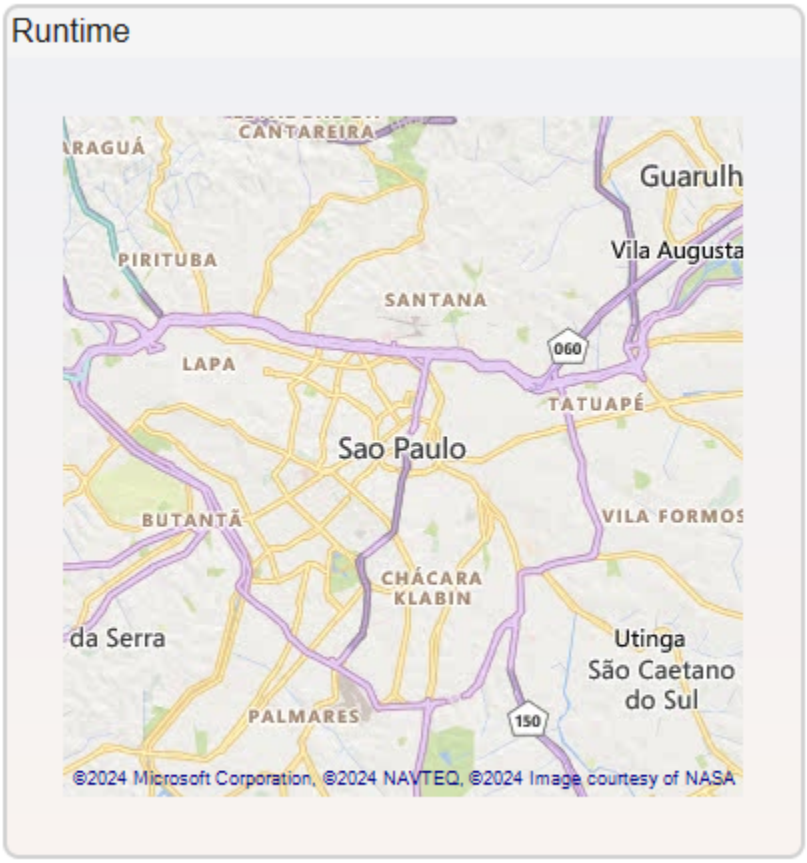
        marker.ToolTipMode = MarkerTooltipMode.OnMouseOver;

        gmap.Overlays[0].Markers.Add(marker);

  }
```

Runtime Execution

During runtime, the MapsGMap retrieves and displays map data based on the specified coordinates. This allows users to visualize and interact with geographical information, making it easier to manage and analyze geo-located assets. The map interface dynamically updates as users navigate or modify the coordinates, providing a seamless and interactive GIS experience.



Related

- See **WPF Client overlay limitations (HWND-hosted controls)** for the shared airspace constraint — on the WPF Client, WPF visuals (buttons, popups, headers) cannot overlay this control.

In this section...
