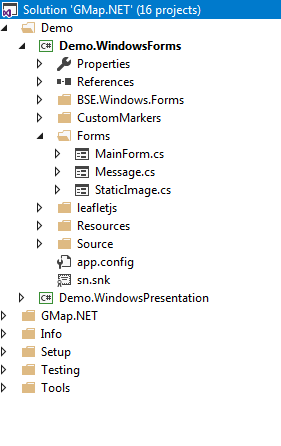
Working with OSM Map

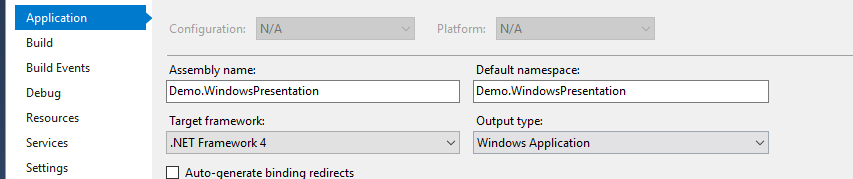
Download GMAp from here:

<https://github.com/radioman/greatmaps>

I tried it from the original web site <https://archive.codeplex.com/?p=greatmaps> but it does not work.



The default freamework is 2.0 , change it to 4

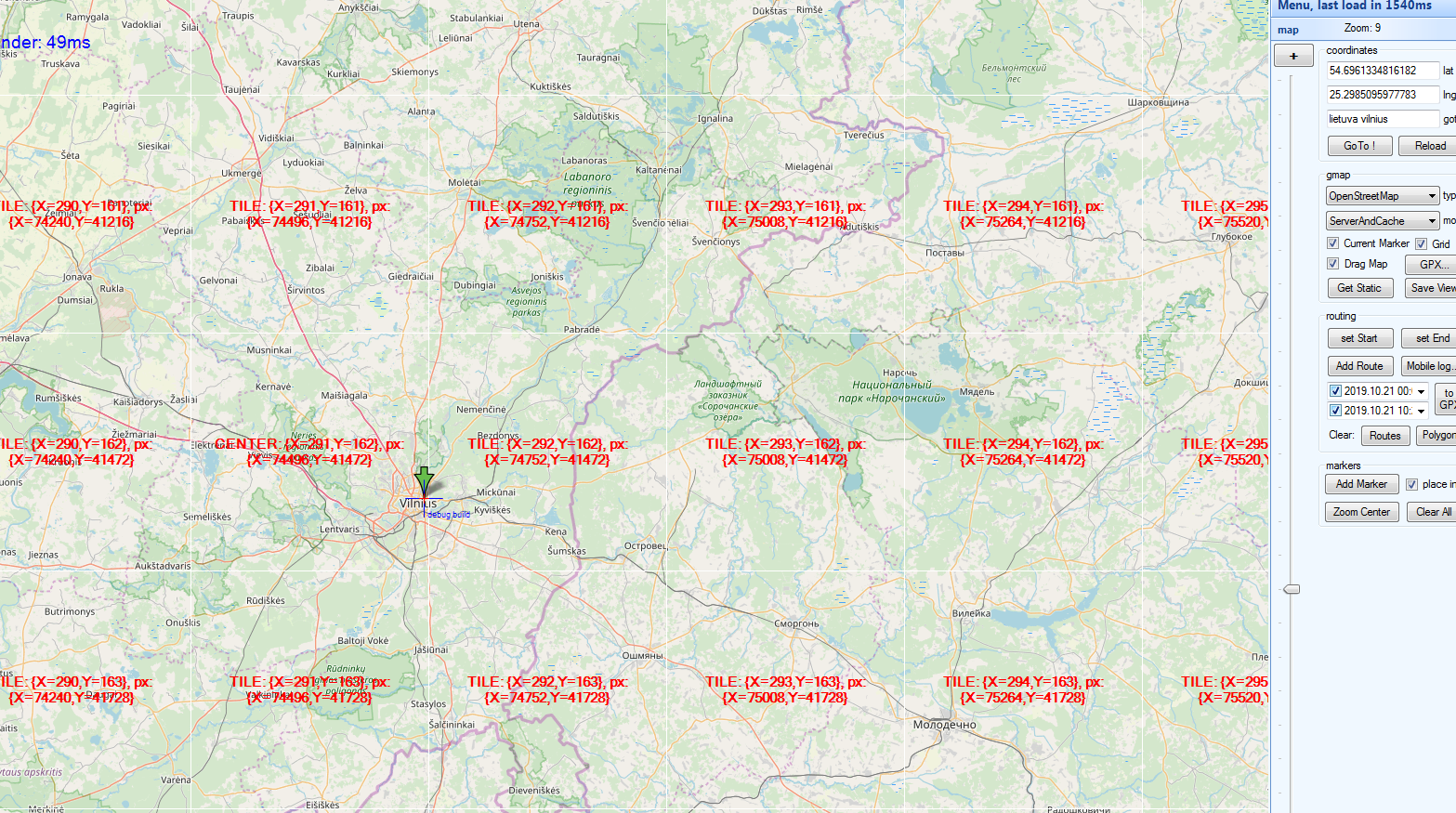


For all projects , then you will be able to compile and run.

Run the Windows Forms demo

You should see initial map

Means you are in the right track



# To understand where you are from the lat lon position , use the Tile Calculator BETA



<https://tools.geofabrik.de/calc/#type=geofabrik_standard&bbox=-74.039364,40.687727,-73.951351,40.732949&tab=1&proj=EPSG:4326&places=2>

search in google, Tile calculator beta

You can search location at the search bar

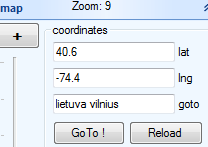


At the bottom right you will see the lon lat



First number is longitude and the second number is latitude

Copy those numbers in the windows form GMAp application



And press Go To!

To the application add the zoom level



private void trackBar1\_ValueChanged(object sender, EventArgs e)

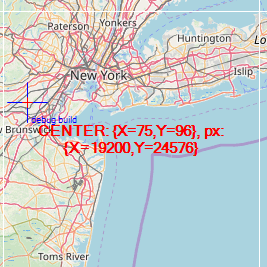
{

MainMap.Zoom = trackBar1.Value / 100.0;

lblZoom.Text = "Zoom: " + MainMap.Zoom;

}

Change the zoom to 8 and you will see new York tile



Whats good in this tool is that it gives you the x, y corrdinates



We will now need to download those tiles

I wrote a small program

That you need to specify the rectangle you want

public struct Tile

{

public int x;

public int x\_size;

public int y;

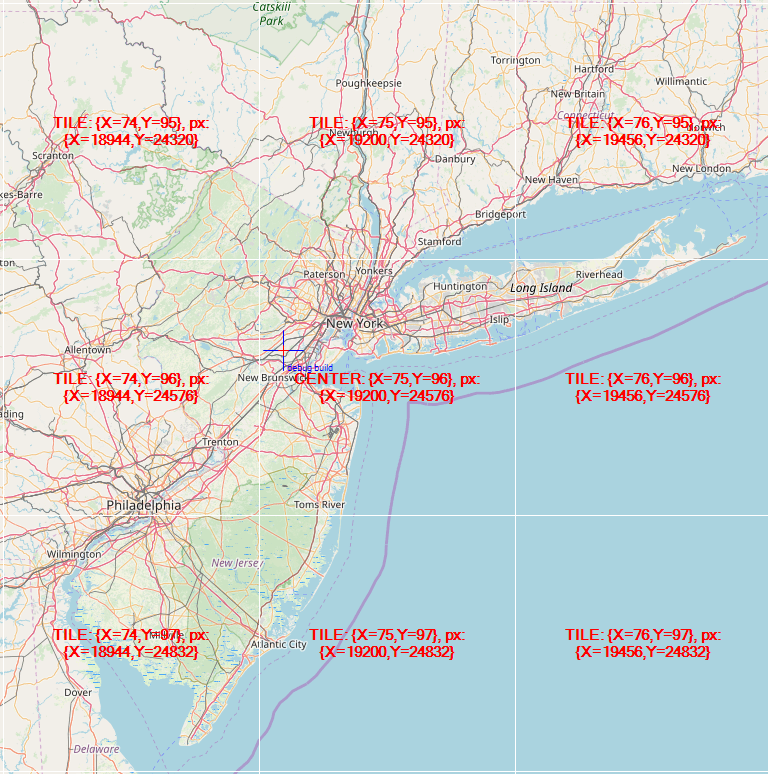
public int y\_size;

public int zoom;

public string name;

}

Lets download 9 tiles of new York , first in zoom 8



So, lets set the tile structure to:

Tile t2 = new Tile

{

x = 74,

x\_size = 3,

y = 95,

y\_size = 3,

name = "new york",

zoom = 8

};

And call to Download Tiles

private readonly string[] \_serverEndpoints = { "a", "b", "c" };

public async void DownloadTiles(Tile tile)

{

Directory.CreateDirectory("c:\\OSM\_Tiles");

HttpClient client = new HttpClient();

var random = new Random();

for (int x = tile.x; x < (tile.x + tile.x\_size); x++)

{

for (int y = tile.y; y < (tile.y + tile.y\_size); y++)

{

try

{

var url = $"http://{\_serverEndpoints[random.Next(0, 2)]}.tile.openstreetmap.org/{tile.zoom}/{x}/{y}.png";

var data = await client.GetByteArrayAsync(url);

File.WriteAllBytes("c:\\OSM\_Tiles\\" + tile.name + "\_" + x + "\_" + y + "\_" + tile.zoom + ".png", data);

}

catch (Exception err)

{

MessageBox.Show(err.Message);

}

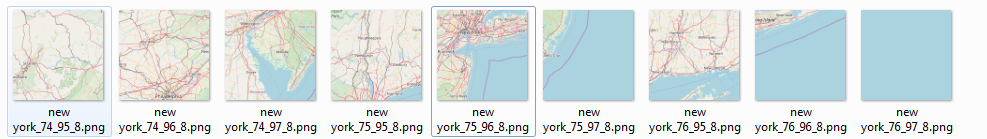
}

}

MessageBox.Show("Finished");

}

Downloading the tiles:



Part 2

Here you can find function to convert from Lat Lon to OSM pixel and tile number

<https://stackoverflow.com/questions/45718374/get-lat-lon-of-pixel-with-in-tile-in-google-maps-in-c-sharp>

Modify a bit the function

static void LatLongToPixelXYOSM(float latitude, float longitude, int zoomLevel, out int pixelX, out int pixelY)

to have the tilex and tiley as return values in the function

void LatLongToPixelXYOSM(float latitude, float longitude, int zoomLevel,

out int pixelX,

out int pixelY,

out int tilex,

out int tiley)

{

float MinLatitude = -85.05112878f;

float MaxLatitude = 85.05112878f;

float MinLongitude = -180;

float MaxLongitude = 180;

float mapSize = (float)Math.Pow(2, zoomLevel) \* 256;

latitude = Clip(latitude, MinLatitude, MaxLatitude);

longitude = Clip(longitude, MinLongitude, MaxLongitude);

float X = (float)((longitude + 180.0f) / 360.0f \* (float)(1 << zoomLevel));

float Y = (float)((1.0 - Math.Log(Math.Tan(latitude \* (Math.PI / 180.0)) + 1.0 / Math.Cos(latitude \* (Math.PI / 180.0))) / Math.PI) / 2.0 \* (1 << zoomLevel));

tilex = (int)(Math.Truncate(X));

tiley = (int)(Math.Truncate(Y));

pixelX = (int)ClipByRange((tilex \* 256) + ((X - tilex) \* 256), mapSize - 1);

pixelY = (int)ClipByRange((tiley \* 256) + ((Y - tiley) \* 256), mapSize - 1);

}

We need the tilex and tiley as input to the download function.