Localization in WSN

Software Recommended: NetSim Standard v12.1/v12.2 (32 bit/ 64 bit), Microsoft Visual Studio 2019

Follow the instructions specified in the following link to clone/download the project folder from GitHub using Visual Studio:
https://tetcos.freshdesk.com/support/solutions/articles/14000099351-how-to-clone-netsim-file- exchange-project-repositories-from-github-

Other tools such as GitHub Desktop, SVN Client, Sourcetree, Git from the command line, or any client you like to clone the Git repository.

Note: It is recommended not to download the project as an archive (compressed zip) to avoid incompatibility while importing workspaces into NetSim.

Secure URL for the GitHub repository:

v12.1: https://github.com/NetSim-TETCOS/Localisation_in_WSN_v12.1.git
v12.2: https://github.com/NetSim-TETCOS/Localisation_in_WSN_v12.2.git

Note: The cloned project directory will contain the documentation specific to the NetSim version (v12.1/v12.2).

Localization is the process of finding the physical or relative location of a sensor node as data and information are useless if the nodes have no idea of their geographical positions. GPS (global positioning system) is the simplest method for localization of nodes, but it becomes very expensive if large number of nodes exists in a given network.

Anchor Nodes:

Sensor nodes with known location information are called “Anchor nodes". Typically, anchor nodes obtain their location information by using a global positioning system (GPS), or by manually being placed at defined coordinates.

Unknown Nodes:

Sensor nodes with unknown location information are called “Non-Anchor nodes" or “Unknown nodes”. Localization is estimated through communication between localized node and unknown node for determining their geometrical placement or position. Location is determined by means of distance and angle between nodes.

Trilateration:

Location of node is estimated through distance measurement from three nodes. In this concept, intersection of three circles is calculated, which gives a single point which is a position of unknown node.

Use the distance equation. If your unknown point is (x, y) and known points are (x_i, y_i) which are at distances r_i from unknown point, then you get three equations:

$$(x - x1)^2 + (y - y1)^2 = r1^2$$

$$(x - x2)^2 + (y - y2)^2 = r2^2$$

$$(x - x3)^2 + (y - y3)^2 = r3^2$$

To calculate the distance between to sensors we have used NetSim API

```
DEVICE_DISTANCE(d1,d2)
```

Expand out the squares and subtract the second equation from the first and third equation from second, we get

$$2(x2 - x1) + 2(y2 - y1)y = r1^2 - r2^2 + x2^2 - x1^2 + y2^2 - y1^2$$

$$2(x3 - x2) + 2(y3 - y2)y = r2^2 - r3^2 + x3^2 - x2^2 + y3^2 - y2^2$$

This is a system of two equations with two unknowns:

$$Ax + By = C$$

$$Dx + Ey = F$$

The values of x and y is obtained from the below equations:

$$x = (CE - FB) / (EA - BD)$$

$$y = (CD - AF) / (BD - AE)$$

Localisation in NetSim:

1. To implement Localisation, we have added **Localisation.c** file in Zigbee project. The file contains the following functions:

   - `int fn_NetSim_localisation()`

     This function is used to find the anchor nodes based on the highest received powers received at unknown sensors from anchor nodes.

   - `int fn_NetSim_trilateration_method()`

     This function is used to implement the trilateration method to calculate the position / location of the unknown sensor.

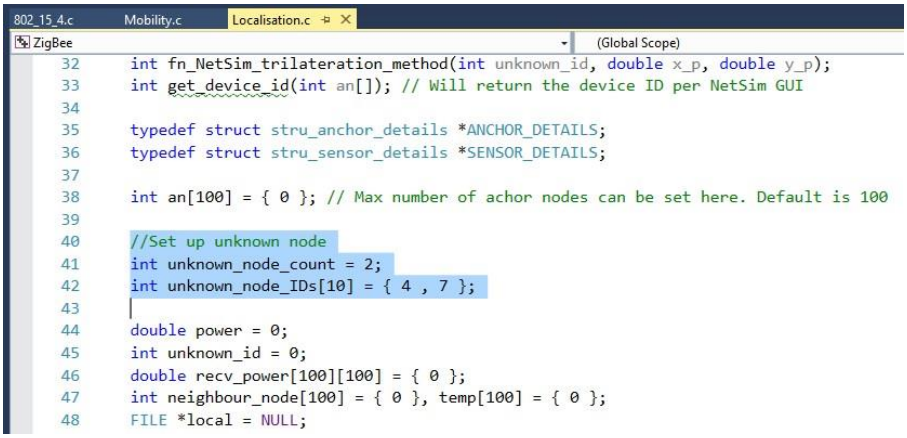
   - `bool IsUnknownNode()`

     This function is used to check whether the given node is unknown node or not.

   - `bool determine_anchor_node()`

     This function is used to check whether the given node is anchor node or not.

2. Users can give their own unknown node IDs and unknown node count in **Localisation.c** file. NetSim knows all the positions of sensor nodes. Localisation is used to find the position of unknown nodes and then comparing this position with NetSim sensor positions.

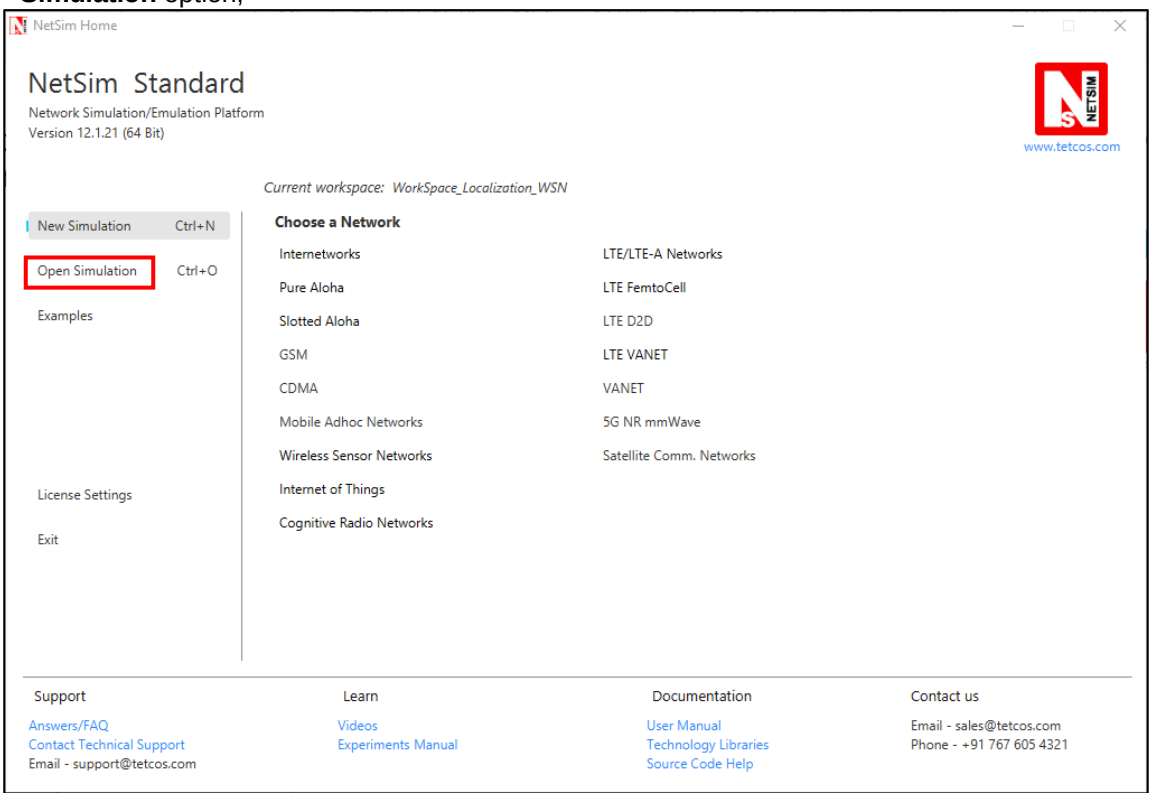


3. Since the unknown nodes are mobile, we have added a call to localisation in **fn_NetSim_Mobility_Run()** function present in **mobility.c** file inside Mobility project to calculate the new positions of the unknown node whenever a node moves.

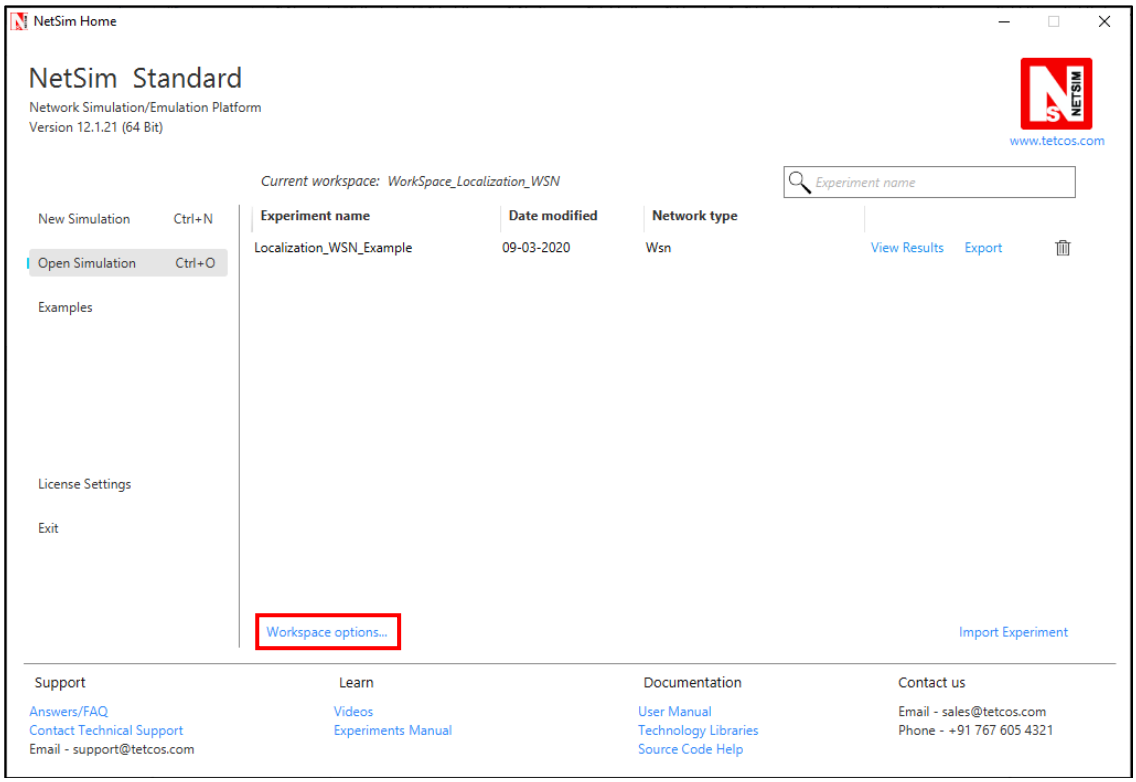
```
Mobility.c | Localisation.c
Mobility | (Global Scope) | fn_NetSim_Mobility_Run()
467     memcpy(cor, ncor, sizeof* cor);
468     if(pstruMobilityVar->dLastTime+pstruMobilityVar->dPauseTime*SECOND<pstruEventDetails->dEventTime+pstruMobilityVar->dCalculati
469     {
470         fn_NMo_RandomPoint(&X, &Y, vel, pstruMobilityVar->dCalculationInterval, &pstruMobilityVar->ulSeed1, &pstruMobilityVar->ul
471         while (cor->corrType == CORRTYPE_CARTESIAN &&
472             (X > dSimulationArea_X || X < 0 || Y < 0 || Y > dSimulationArea_Y))
473         {
474             X = cor->X;
475             Y = cor->Y;
476             fn_NMo_RandomPoint(&X, &Y, vel, pstruMobilityVar->dCalculationInterval, &pstruMobilityVar->ulSeed1, &pstruMobilityVar
477         }
478         ncor->X = X;
479         ncor->Y = Y;
480         //store the last time
481         pstruMobilityVar->dLastTime = pstruEventDetails->dEventTime+pstruMobilityVar->dCalculationInterval;
482     }
483     //update the device position
484     fn_NetSim_localisation();
485     memcpy(pos,cor,sizeof* pos);
486
487     mobility_pass_position_to_animation(pstruEventDetails->nDeviceId,
488                                         pstruEventDetails->dEventTime,
489                                         pos);
490
491     //Add event for next point
492     pstruEventDetails->dEventTime+=pstruMobilityVar->dCalculationInterval;
493     fnpAddEvent(pstruEventDetails);
494     pstruEventDetails->dEventTime-=pstruMobilityVar->dCalculationInterval;
495 }
```

Steps:

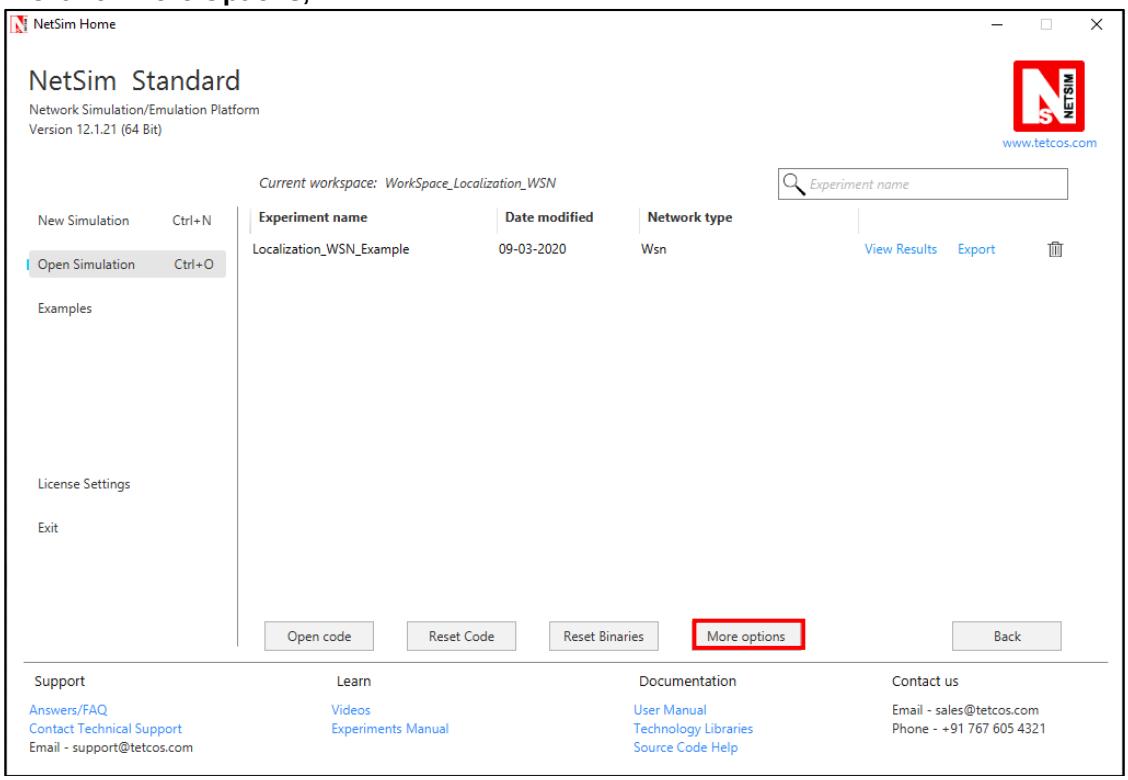
- After you unzip the downloaded project directory, Open NetSim Home Page click on **Open Simulation** option,



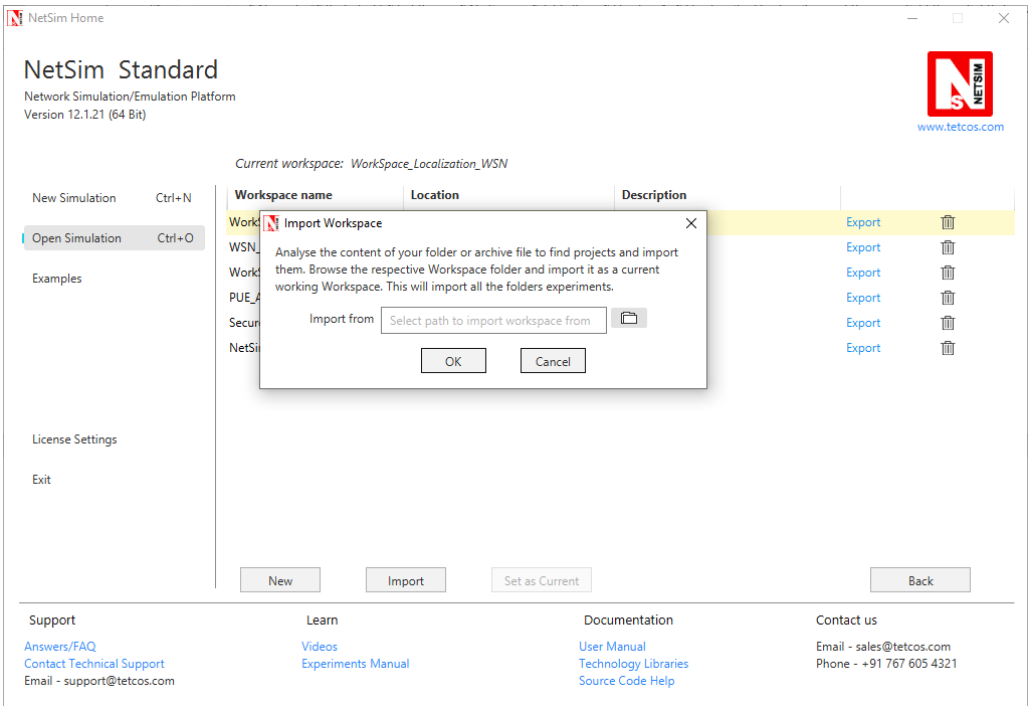
- Click on **Workspace options**



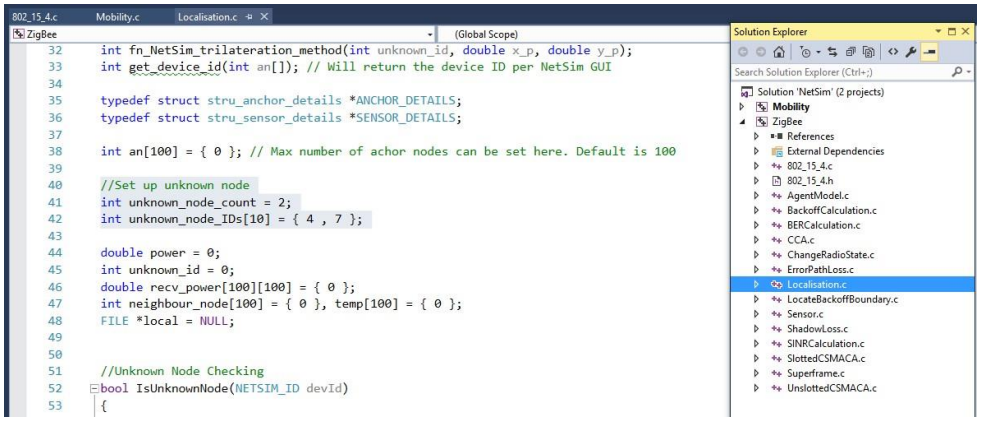
- Click on **More Options,**



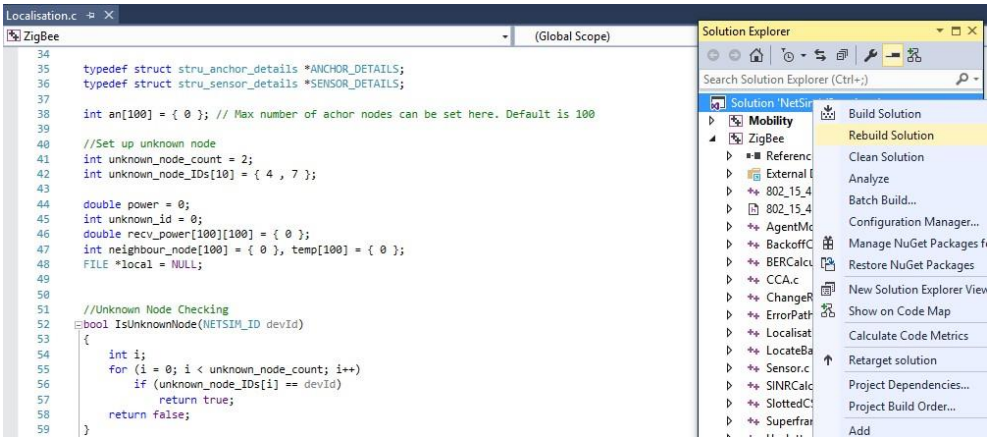
- Click on **Import**, browse the extracted folder path and go into the WorkSpace_Localization_WSN directory. Click on the Select folder button and then on **OK.**



- Go to home page, Click on **Open Simulation → Workspace options → Open code**

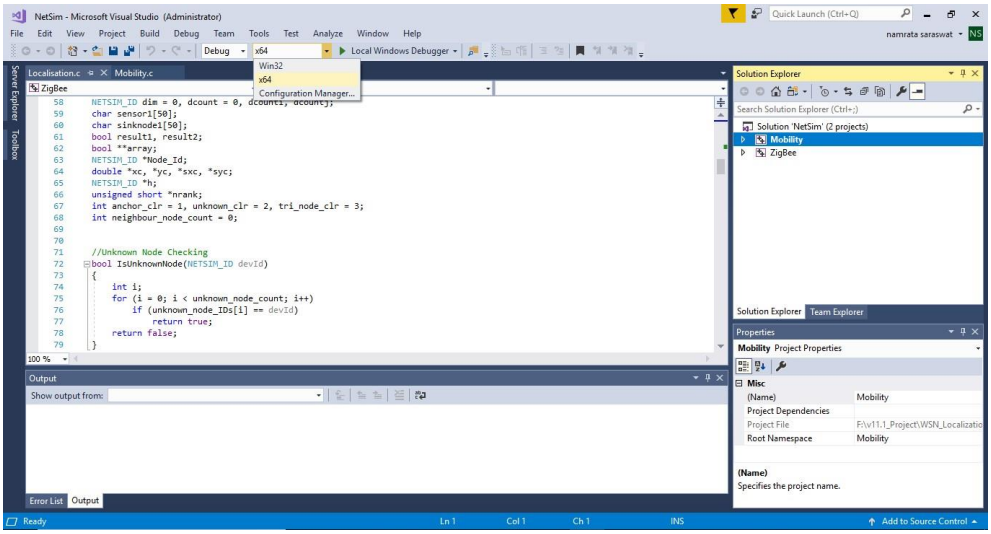


- Right click on Solution in Solution Explorer and select rebuild solution.

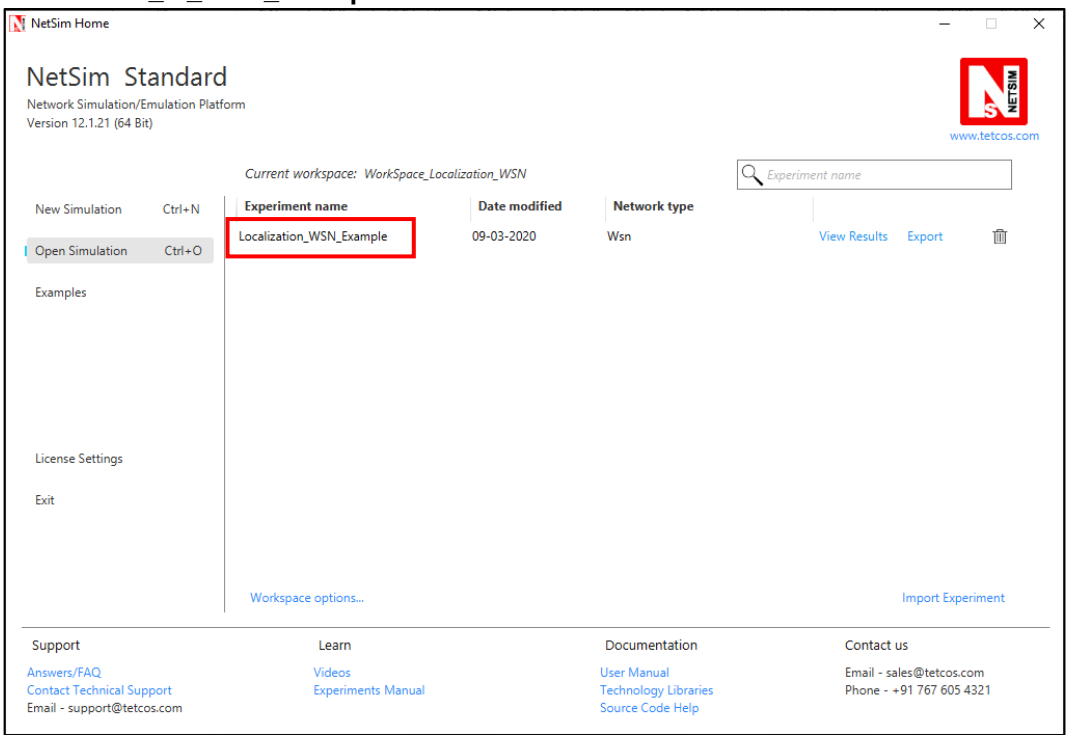


- Upon rebuilding, **libZigbee.dll** and **libMobility.dll** will automatically get updated in NetSim binary folder.

Note: Based on whether you are using NetSim 32 bit or 64 bit setup you can configure Visual studio to build 32 bit or 64 bit Dll files respectively as shown below:



- Go to NetSim home page, click on **Open Simulation**, Click on **Localization_in_WSN_Example**.



- Run simulation after the network scenario gets loaded.
- After simulation, localisation.txt file will get created in the **bin** folder of NetSim. Right click on the NetSim shortcut icon in your desktop and select Open file location to go to NetSim bin folder. The localisation.txt file logs the unknown node IDs, received powers from all anchor nodes to unknown nodes, anchor node IDs based on highest received powers and the position or coordinates of the unknown nodes with variation in time as shown below.

localisation - Notepad

File Edit Format View Help

```
From 3 to 7 is: -84.7827469788 dbm
From 5 to 7 is: -85.1956026853 dbm
From 6 to 7 is: -87.7656517820 dbm
From 8 to 7 is: -89.4904372973 dbm
From 9 to 7 is: -86.0518999457 dbm
Unknown node = 4
Anchor nodes = 1, 2, 9,
The position of Unknown node 4 at time 24000000.000000µs = 7, 12
Unknown node = 7
Anchor nodes = 1, 2, 3,
The position of Unknown node 7 at time 24000000.000000µs = 62, 9

Unknown nodes
4
7
Received powers
From 1 to 4 is: -80.3528052692 dbm
From 2 to 4 is: -83.1987039075 dbm
From 3 to 4 is: -86.4787581036 dbm
From 5 to 4 is: -87.2647955930 dbm
From 6 to 4 is: -89.2354282518 dbm
From 8 to 4 is: -90.6437199782 dbm
From 9 to 4 is: -86.4472286436 dbm
From 1 to 7 is: -77.8310409707 dbm
From 2 to 7 is: -79.5792536503 dbm
From 3 to 7 is: -84.7827469788 dbm
From 5 to 7 is: -85.1956026853 dbm
From 6 to 7 is: -87.7656517820 dbm
From 8 to 7 is: -89.4904372973 dbm
From 9 to 7 is: -86.0518999457 dbm
Unknown node = 4
Anchor nodes = 1, 2, 9,
The position of Unknown node 4 at time 24000000.000000µs = 7, 12
Unknown node = 7
Anchor nodes = 1, 2, 3,
The position of Unknown node 7 at time 24000000.000000µs = 62, 9
```