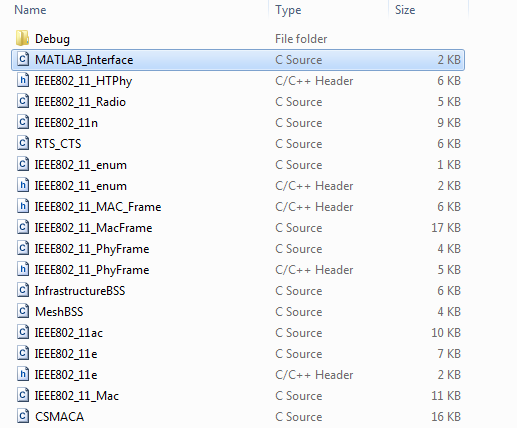
Implement Nakagami Distribution of MATLAB in NetSim without using .m file

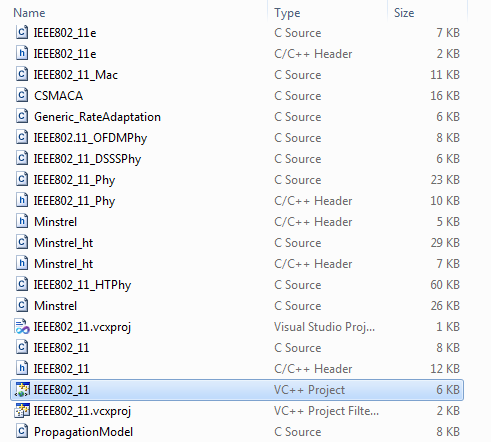
In this project we will replace the default Rayleigh Fading (part of the path loss calculation) used in NetSim, with a Fading Power calculated using the Nakagami Distribution from MATLAB.

# Steps to run MATLAB Interfacing Code in NetSim:

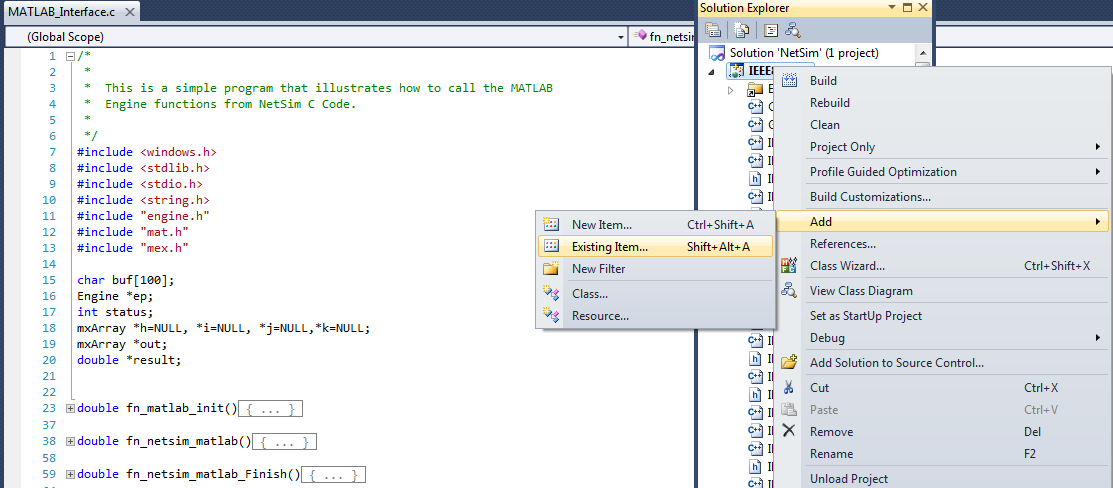
1. Create a MATLAB\_Interface.c file inside the IEEE802\_11 folder which can be found in the path <NetSim\_Install\_Direcotry>/src/Simulation/.



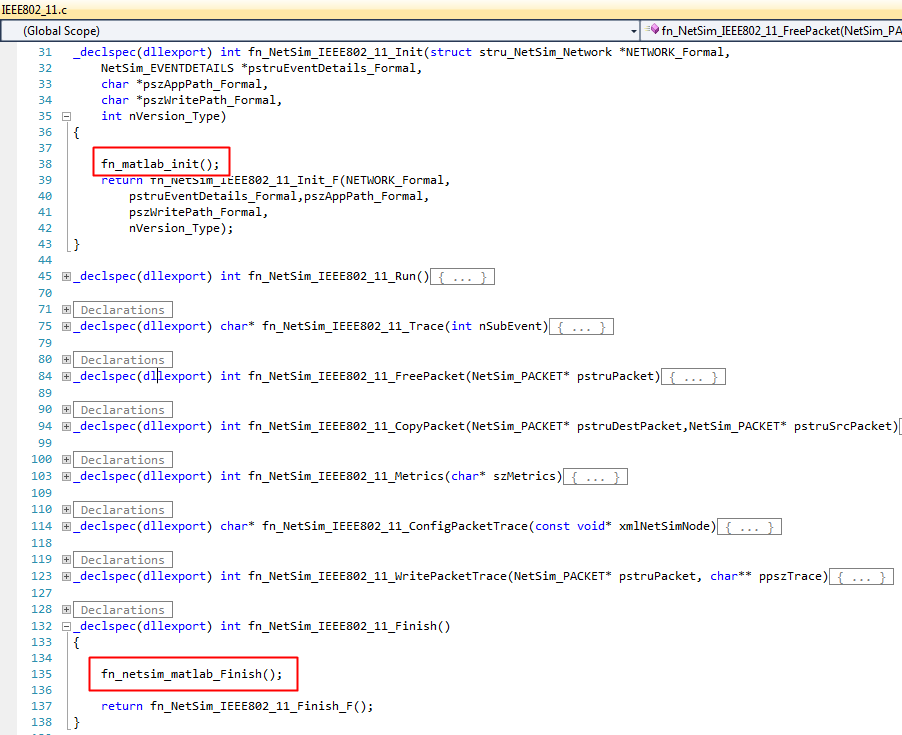
1. Write the following code inside the MATLAB\_Interface.c file:-
2. /\*
3. \*
4. \* This is a simple program that illustrates how to call the MATLAB
5. \* Engine functions from NetSim C Code.
6. \*
7. \*/
8. #include <windows.h>
9. #include <stdlib.h>
10. #include <stdio.h>
11. #include <string.h>
12. #include "engine.h"
13. #include "mat.h"
14. #include "mex.h"
15. char buf[100];
16. Engine \*ep;
17. int status;
18. mxArray \*h=NULL, \*i=NULL, \*j=NULL,\*k=NULL;
19. mxArray \*out;
20. double \*result;
21. double fn\_matlab\_init()
22. {
23. /\*
24. \* Start the MATLAB engine
25. \*/
26. if (!(ep = engOpen(NULL))) {
27. MessageBox ((HWND)NULL, (LPCWSTR)"Can't start MATLAB engine",
28. (LPCWSTR) "MATLAB\_Interface.c", MB\_OK);
29. exit(-1);
30. }
31. engEvalString(ep,"desktop");
32. return 0;
33. }
34. double fn\_netsim\_matlab()
35. {
36. //write your own implementation here
37. int nakagami\_shape=5,nakagami\_scale=2;
39. engPutVariable(ep,"h",h);
40. sprintf(buf,"h=ProbDistUnivParam('nakagami',[%d %d])",nakagami\_shape,nakagami\_scale);
41. status=engEvalString(ep,buf);
43. engPutVariable(ep,"i",i);
44. sprintf(buf,"i=random(h,1)");
46. status=engEvalString(ep,buf);
47. out=engGetVariable(ep,"i");
48. result=mxGetPr(out);
49. return \*result;
50. }
51. double fn\_netsim\_matlab\_Finish()
52. {
53. status=engEvalString(ep,"exit");
54. return 0;
55. }
56. Now open IEEE802\_11 project file, inside the IEEE802\_11 folder.



1. Right click on IEEE802\_11 project🡪Add🡪 Existing Item🡪 Select MATLAB\_Interface.c file.



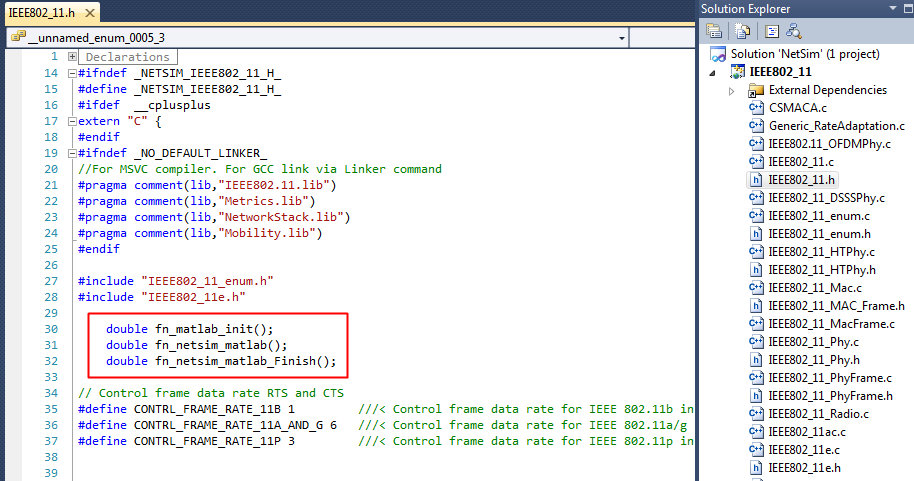
1. Open IEEE802\_11.c file and add fn\_matlab\_init() and fn\_netsim\_matlab\_Finish() inside fn\_NetSim\_IEEE802\_11\_Init() and fn\_NetSim\_IEEE802\_11\_Finish() respectively.



1. Add definitions of the following functions inside IEEE802\_11.h file

double fn\_matlab\_init();

double fn\_netsim\_matlab();

double fn\_netsim\_matlab\_Finish();

1. Go to the fn\_netsim\_matlab () function and write the following codes :

int nakagami\_shape=5,nakagami\_scale=2;

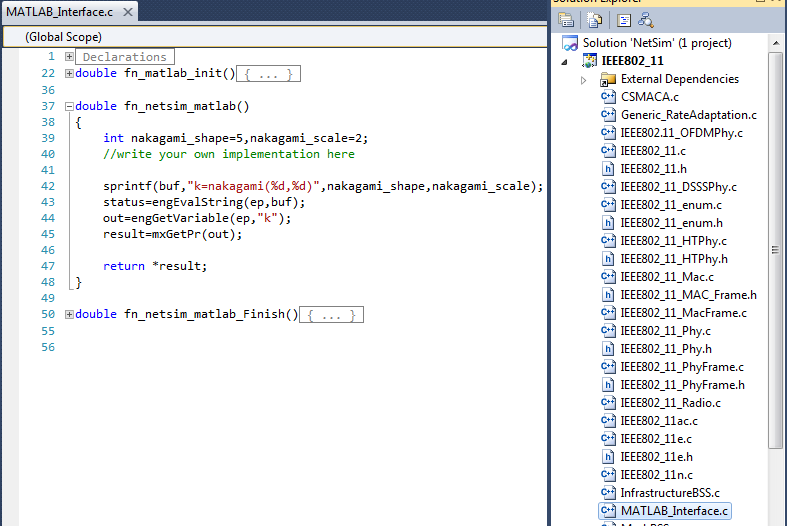
//write your own implementation here

sprintf(buf,"k=nakagami(%d,%d)",nakagami\_shape,nakagami\_scale);

status=engEvalString(ep,buf);

out=engGetVariable(ep,"k");

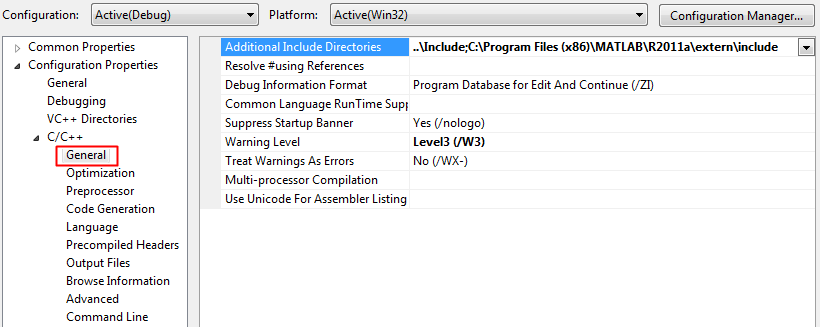
result=mxGetPr(out);



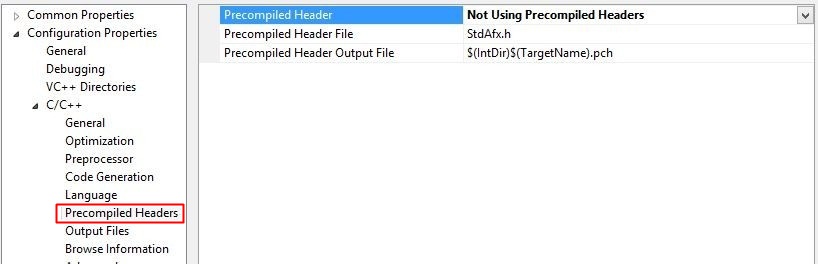
1. To compile a MATLAB engine application in the Microsoft Visual Studio 2010 environment, Right click on the IEEE802\_11 project and select PROPERTIES in the solution explorer. Once this window has opened, make the following changes:

**a.** Under C/C++ General, add the following directory to the field ADDITIONAL INCLUDE DIRECTORIES:

<Path where MATLAB is installed>\extern\include

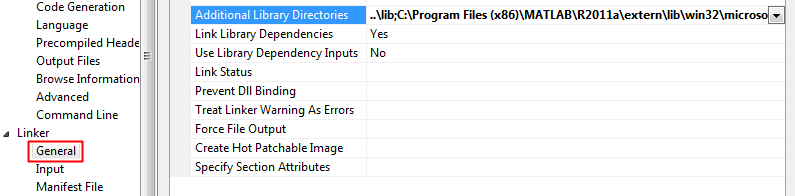
****

**b.** Under C/C++ Precompiled Headers, select "Not Using Precompiled Headers".



**c.** Under Linker General, add the directory to the field ADDITIONAL LIBRARY DIRECTORIES:

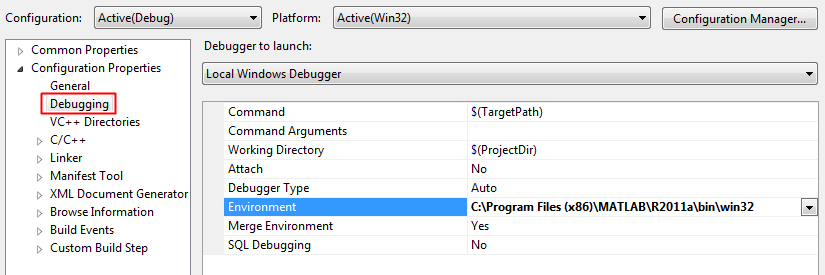
<Path where MATLAB is installed>\extern\lib\win32\microsoft



**d.** Under Configuration Properties ->Debugging

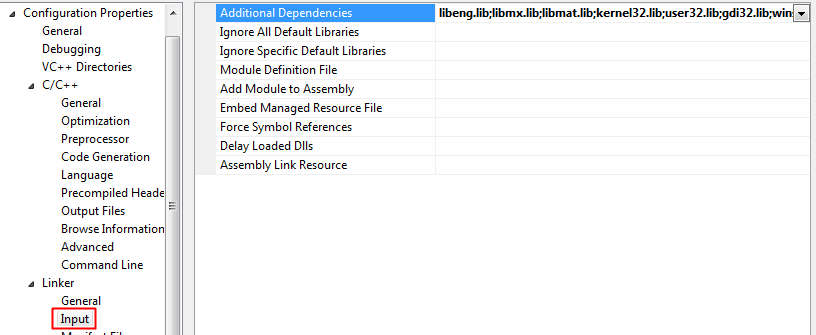
Add the following Target path in the *Environment*:

<Path where MATLAB is installed>\bin\win32



**e.** Under Linker Input, add the following names to the field marked ADDITIONAL DEPENDENCIES:

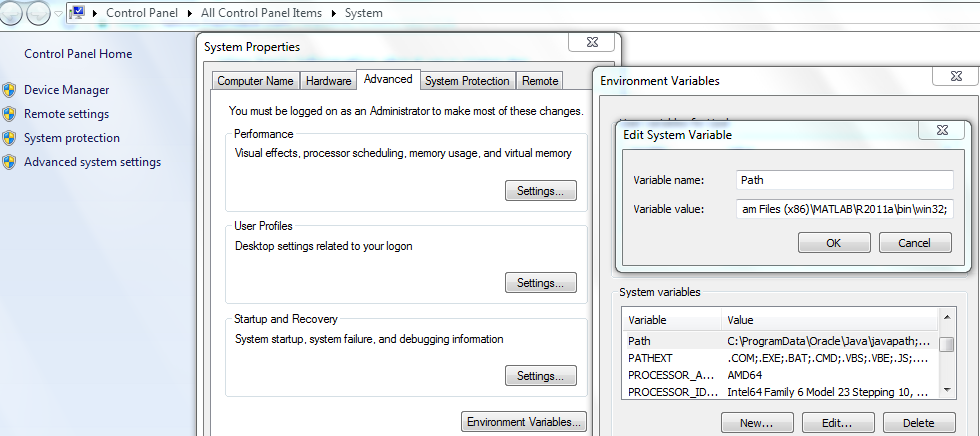
libmx.lib; libmat.lib; libeng.lib by separating them with a semicolon.



Click on Apply and then on ok.

**f.** Make sure that the following directory is in the PATH(Environment variable)

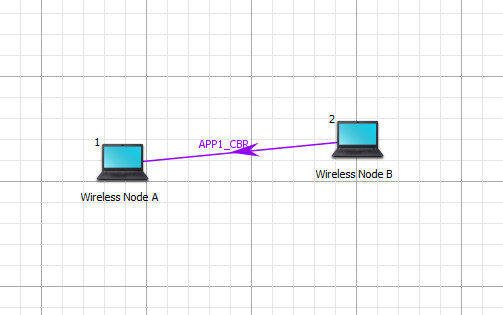
<Path where MATLAB is installed>\bin\win32



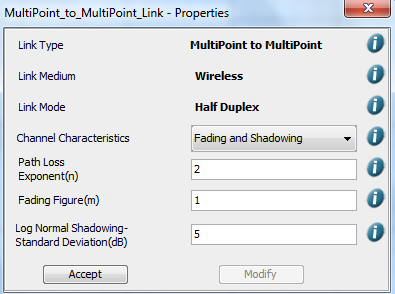
**NOTE:**

To run this code 32- bit version of MATLAB must be installed in your system.

1. Now Right Click on IEEE802\_11 project and select Rebuild.
2. Now replace the newly built libIEEE802.11.dll from the DLL folder, into the NetSim bin folder. Please ensure you rename the original libIEEE802.11.dll file to retain a copy of the original file.
3. Run NetSim in Administrative mode. Create a Network scenario involving IEEE802\_11 say MANET.



1. Set the Fading Figure value in the Multipoint to Multipoint Link properties to 1, to ensure that Rayleigh fading is set.



1. Perform Simulation. You will find that once the Simulation starts MATLAB command window starts and gets closed once the simulation is over.