**NetSim Multi-Parameter Sweeper MATLAB Script**

**Software:** NetSim v14.1 (64 bit), [DOT NET CORE SDK 3.1](https://dotnet.microsoft.com/download/dotnet-core/3.1), MATLAB (2022 or higher)

**Project Download Link:**

<https://github.com/NetSim-TETCOS/Multi-Parameter_Sweeper_v13.3/archive/refs/heads/main.zip>

**Introduction**

When users want to *sweep*one or more parameters, they change their values between simulation runs, and compare and analyse the performance metrics from each run. NetSim multi-parameter sweeper enables users to automate the sweep process.

Consider an example, where a user wishes to create and simulate a network scenario for all possible values of one or more parameters in combination and analyse a set of performance metrics across the simulation runs. This is extremely time consuming to do manually using the NetSim GUI.

The multi-parameter sweep program enables users to automate the sweep process across multiple input parameters, simulate each run, save each result, and compare specific output metrics via a spreadsheet software like MS Excel.

The sweep program runs NetSim via its CLI interface.

**File Organization**

The project directory consists of several binaries which are responsible for different tasks during a multi-parameter sweep:



**Figure 1:** project directory consists of several binaries

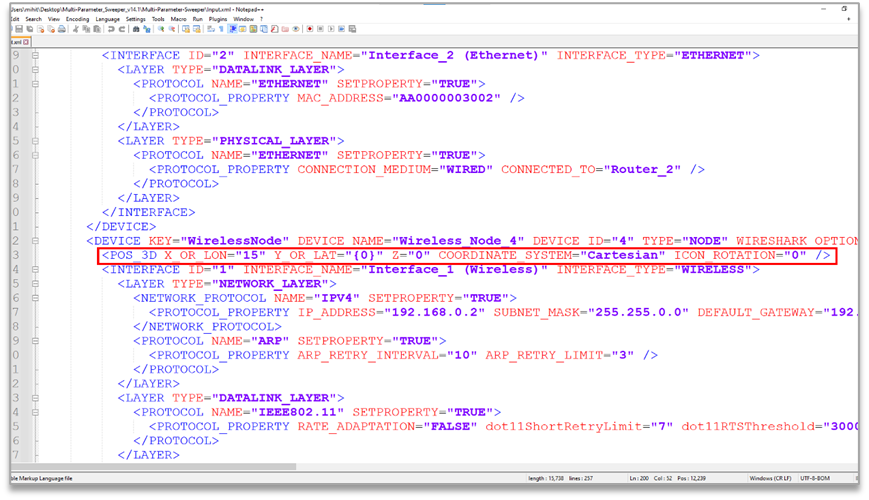
1. **input.xml:** This file contains the base NetSim network configuration that is to be simulated. This file is created by copy pasting the Configuration.netsim file that can be obtained by saving a network configuration in NetSim and renaming it to input.xml.

A screenshot of a computer

Description automatically generated

**Figure 2:** NetSim input Configuration.netsim file

* The values of parameters which are to be varied during each simulation run needs to be specified as {0}, {1}, {2}, etc. respectively.
* For Example, if the Y coordinate of a device is to be varied the values can be modified in the input.xml file as shown below:



**Figure 3:** Modify X and Y coordinates in input input.xml file

1. **Script**.**txt:** This file should be updated with the parameter from the output metrics of NetSim that is to be logged at the end of each simulation run for the purpose of analysis.

At the end of every simulation, NetSim generates a Metrics.xml file which contain the performance metrics written in a specific format based on which it is loaded in the results dashboard.

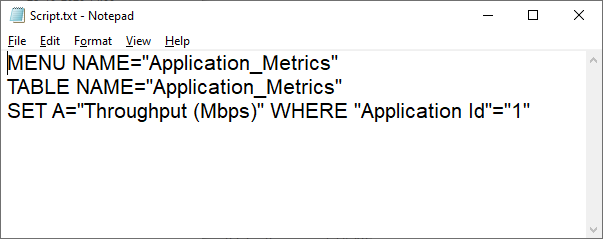
Each Metric is part of a results table which can be accessed using a menu in the results dashboard.

A NetSim Metrics.xml file is shown below:



Figure 4: NetSim output Metrics.xml file

For Example, if the application throughput is to be logged for each simulation run then the script file can be updated as shown below:



**Figure 5:** The application throughput is to be logged for each simulation modified in Script.txt

1. **ConfigWriter.exe**: This executable takes one or more command line arguments as input and generated Configuration.netsim file by replacing the arguments in place of the variable parameters specified in the input.xml file.

If there are two variable parameters specified in the input.xml file ({0} and {1}) then two arguments need to be passed while calling ConfigWriter.exe.

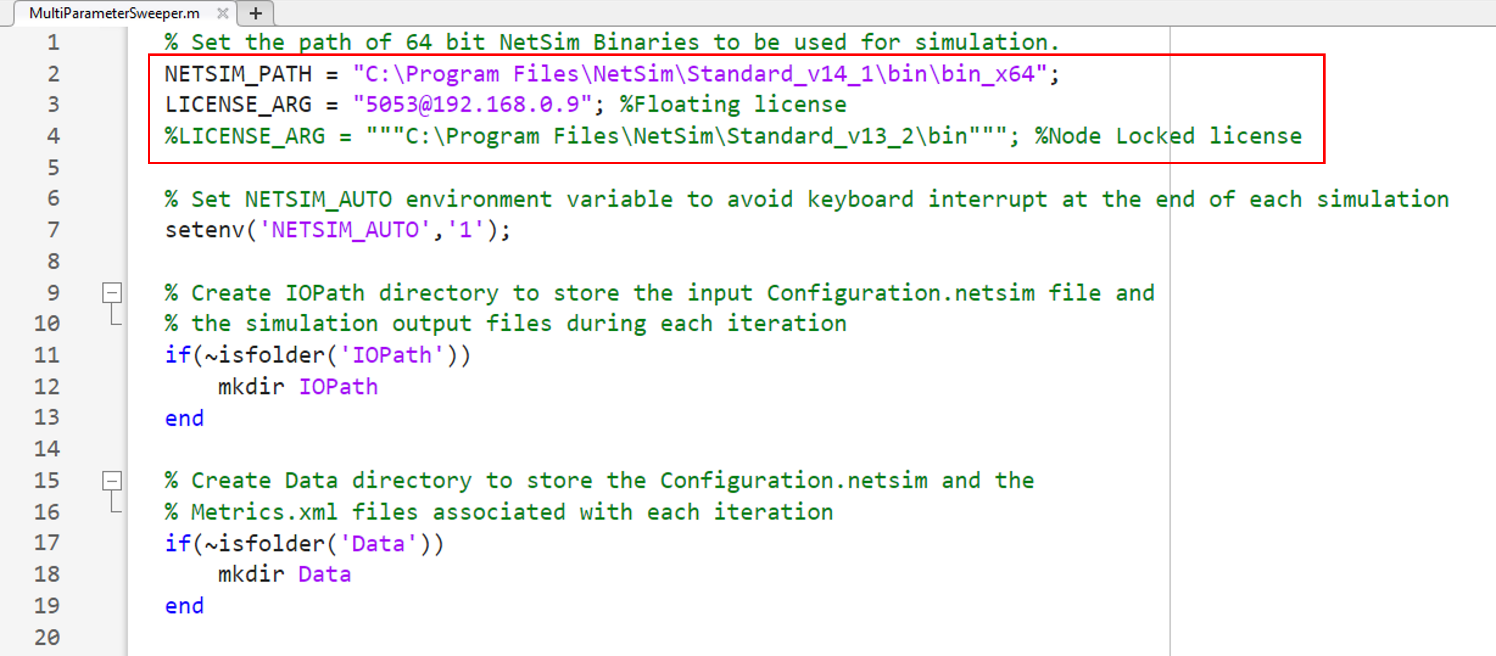
1. **MetricsCSV.exe:** This executable is used to convert the Metrics.xml file present in the output folder into a comma separated file, MetricsPrint.csv.
2. **MetricsReader.exe**: This executable is responsible for reading the output parameter from the Metrics.xml file generated after each simulation and logging it to the results file.

It uses the Script.txt file to determine which parameter to read from the Metrics file.

If multiple parameters are to be read and logged, then the MetricsReader.exe can be called multiple times with Script.txt file having information about the parameter to be read each time.

1. **Supporting DLL’s**: Some the supporting files such as ConfigWriter.dll, MetricsReader.dll, NetSimMetricsReader.dll, NetSimXmlReader.dll, etc. which are present in the project folder are used by other executable such as ConfigWriter.exe and MetricsReader.exe for various purposes during a multi-parameter sweep.
2. **MultiParameterSweeper.m** uses MATLAB programming language which is less complex and offers more flexibility as the number of input and output parameters increases.

* Users can also write the script to run the multi-parameter sweep process in a preferred programming language as per the convenience.
* The script can be configured to run multiple simulation iterations based on the number of parameters to be varied and the range of values of each parameter.
* NETSIM\_PATH variable can be set to the path of NetSim 64-bit binaries (bin\_x64) in the install directory or workspace which is to be used to run Simulations.
* LICENSE\_ARG variable can be set to License server port and IP details in case of floating on premise licenses or the path of license file in case of node locked or cloud licenses.



**Figure 6:** User need to set NetSim Path based on 64-bit

For example,

NETSIM\_PATH = "C:\Users\HP\OneDrive\Documents\NetSim\Workspaces\Default\_Workspace\_v14\_1\bin\_x64";

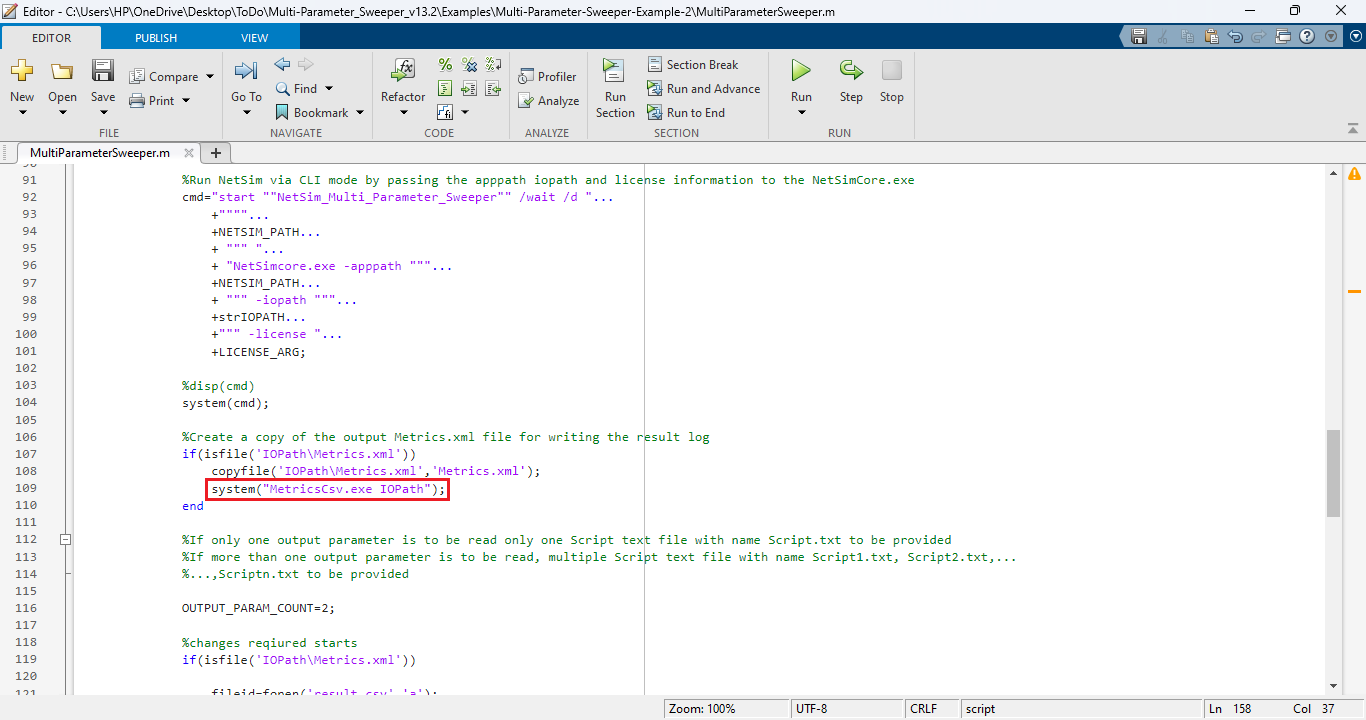
**Floating on-premise license (<port no>@<server ip address>):**

LICENSE\_ARG = "5053@192.168.0.9";

**Node Locked or Cloud licenses (path of license file):**

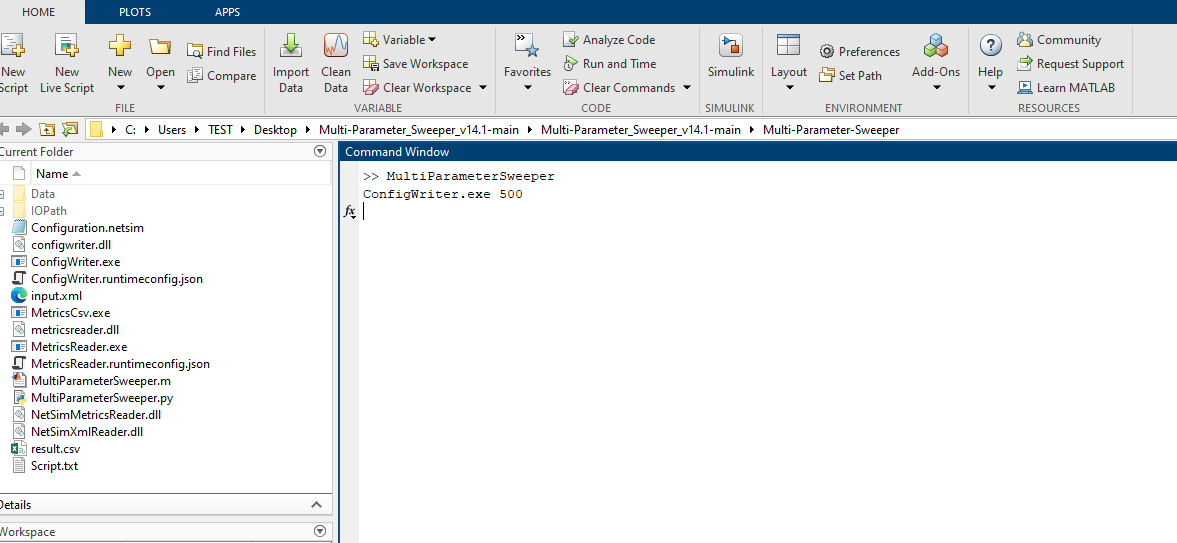
LICENSE\_ARG = """C:\Program Files\NetSim\Standard\_v14\_1\bin""";

* The MetricsCSV.exe will convert the Metrics.xml file inside the output folder into a csv file.



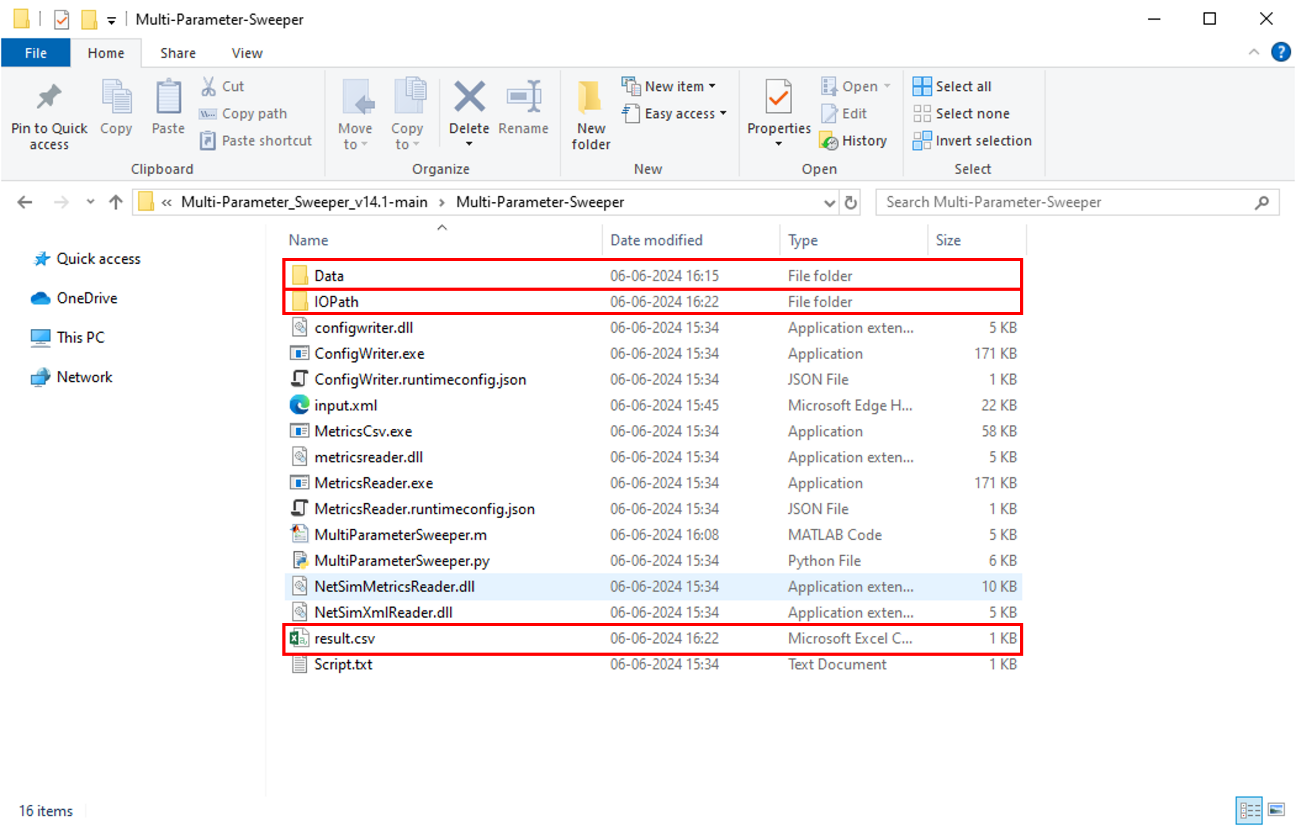
**Figure 7:** MetricsPrint.csv file will be created in the IOPath and then copied into the respective output folder

1. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:



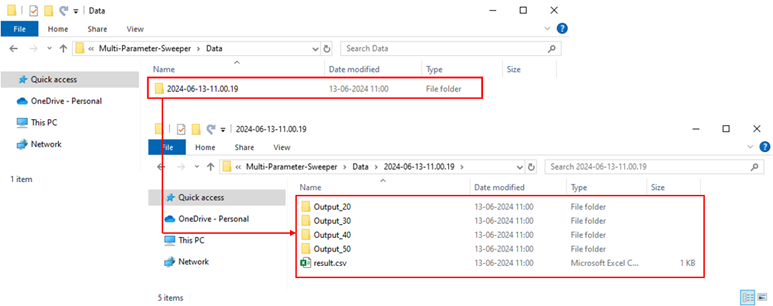
**Figure 8:** Running MATLAB script using cmd prompt

1. This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of UE.
2. At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:



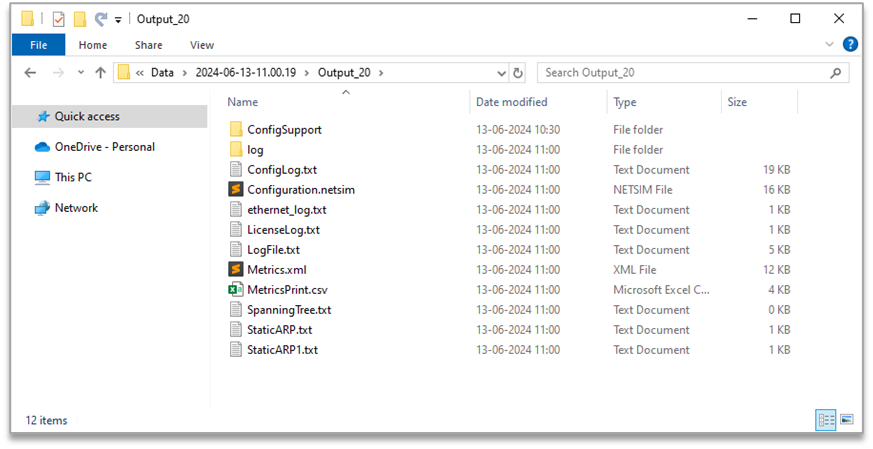
**Figure 9:** After Simulation Multi-Parameter-Sweeping folder contains output files like result.scv, Data etc

* **Data**: Contains multiple folders created based on date and time of simulation inside which multiple output folders corresponding to each simulation run, with its name including the value of the parameters in that iteration gets created.



**Figure 10:** Based on values in the iteration, Output folder gets created in the Data directory inside a folder named as per date and time of simulation along with a copy of result.csv file.

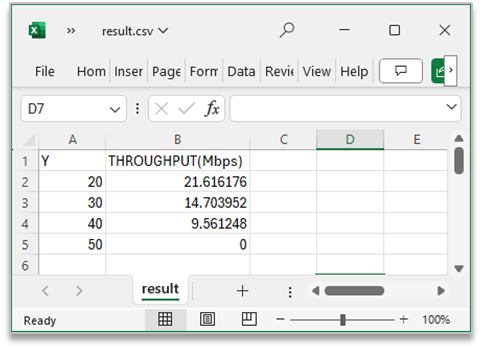
* Each folder contains the all the output files associated with the simulation run.



**Figure 11:** Each folder contains all the output files and the Metrics.xml file converted to MetricsPrint.csv file

***Note:*** *User should keep a back-up of the data folder to avoid data loss.*

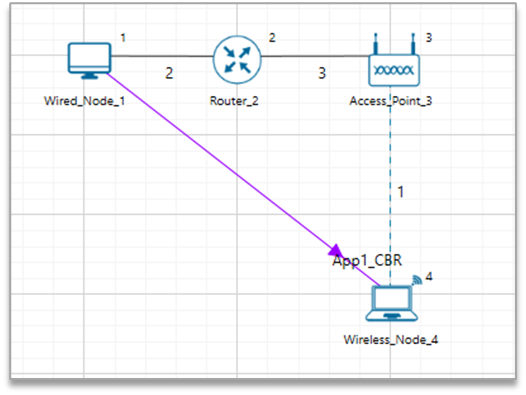
* **IO Path**: Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
* **Result.csv**: This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run. The result.csv file will also be copied into the output folder after each simulation.



**Figure 12**: Iterated value and Throughput obtained listed in result.csv

**Example 1: Modifying a single input parameter and logging a single output parameter**

Consider the following network Internetwork (WiFi) network scenario in NetSim, comprising of a Wired Node, Router, Access Point and a Wireless Node.



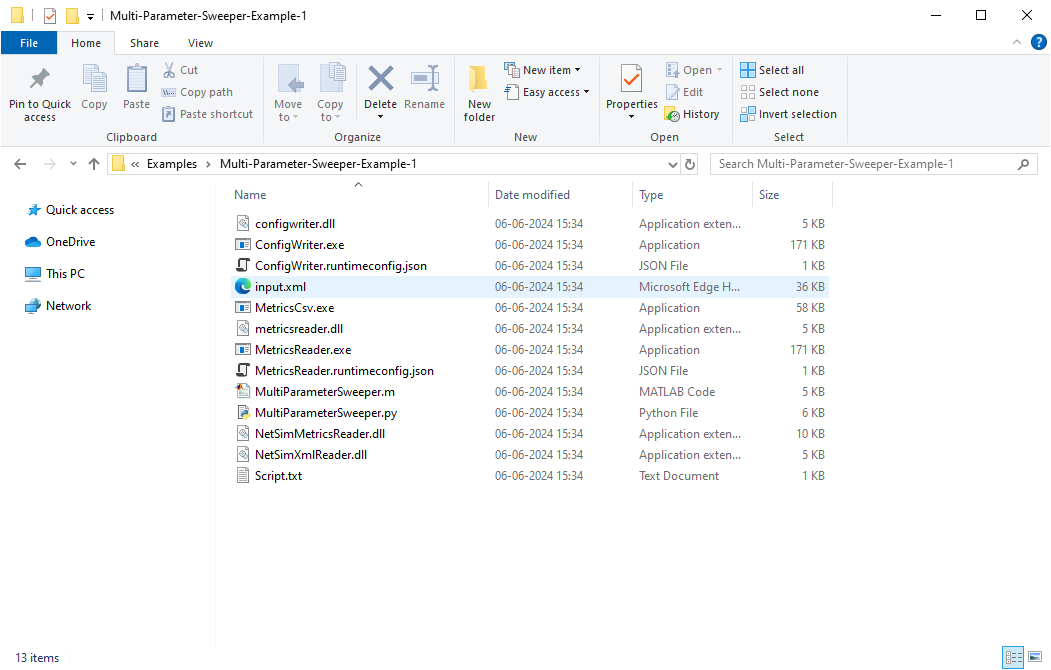
**Figure 13:** Network Topology

The network configuration has the initial distance between the Access Point and Wireless Node as 20 meters with the Access Point located at (15,10) and UE located at (15,20).

Multi-Parameter Sweeper is configured to run simulations for different distance between the gNB and UE by varying the UE Y coordinate value from 20 to 50 in steps of 10 meters.

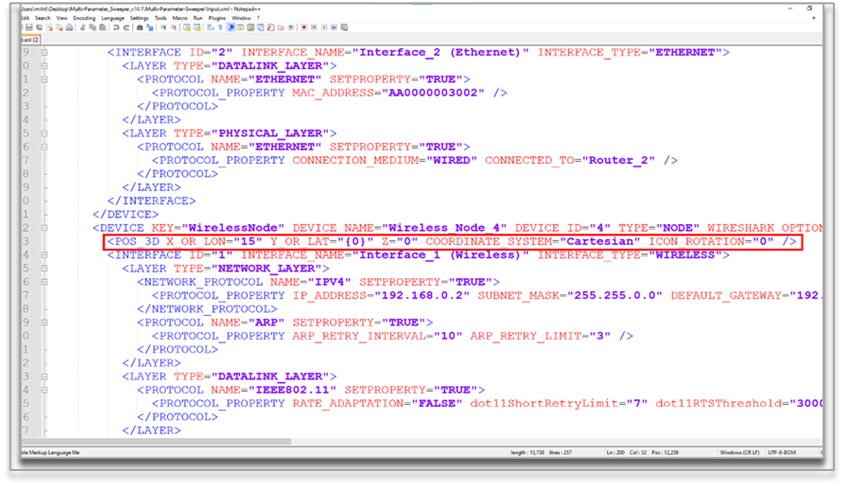
The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.

Refer to the Example 1 directory which is part of the project folder (Multi-Parameter-Sweeper\_v14.1-main\Examples\Multi-Parameter-Sweeper-Example-1)



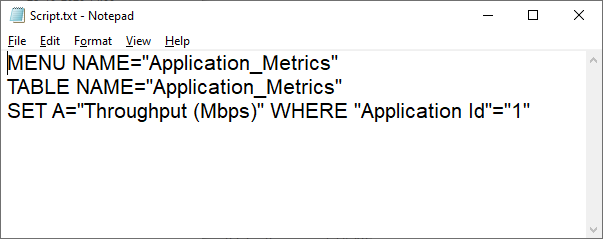
**Figure 14:** Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

1. The value of the Y coordinate of UE that is to be modified during each simulation run is updated (“{0}”) in the configuration file as shown below:



**Figure 15:** Modified Y coordinate of UE in input.xml

1. The Script.txt file is updated with the details of the output parameter to be read from the Metrics.xml file and added to the result csv log file. In this case the Application throughput is to be logged for each simulation run.



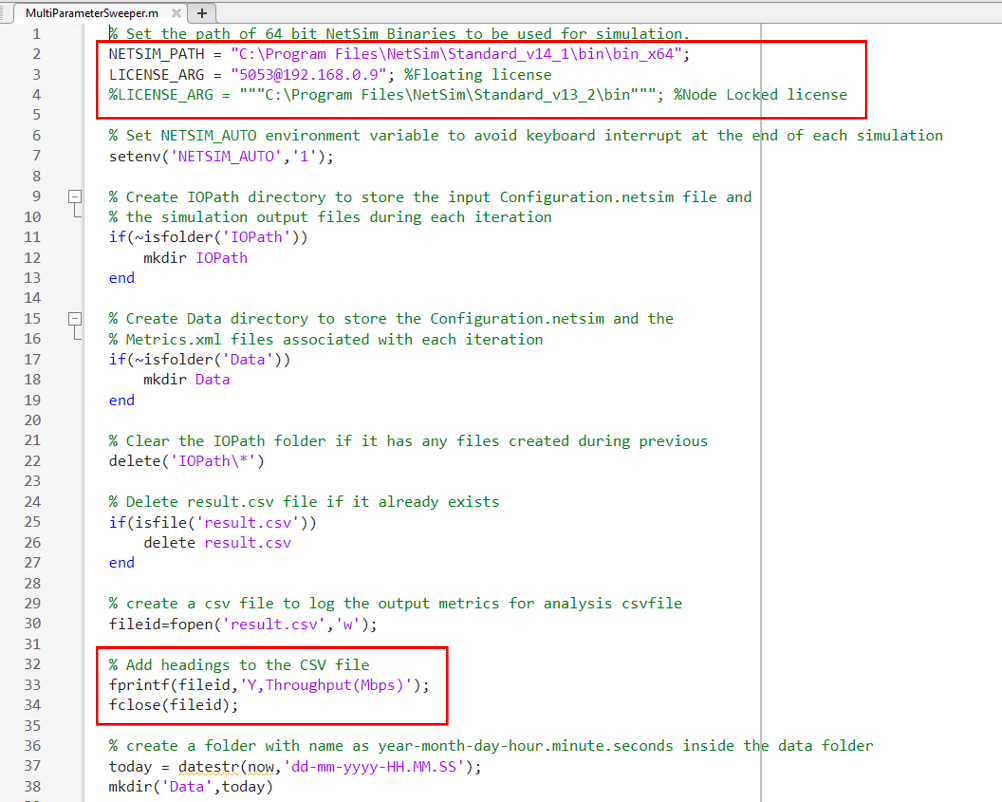
**Figure 16:** The application throughput is to be logged for each simulation modified in Script.txt

1. MultiParameterSweeper.m is updated to pass the Y coordinate value during each iteration to generate Configuration file run simulation and update the result csv log.

The MultiParameterSweeper.bat batch script modified for running simulations for different values of Y coordinates starting from 20 up to 50 in steps of 10 is shown below:

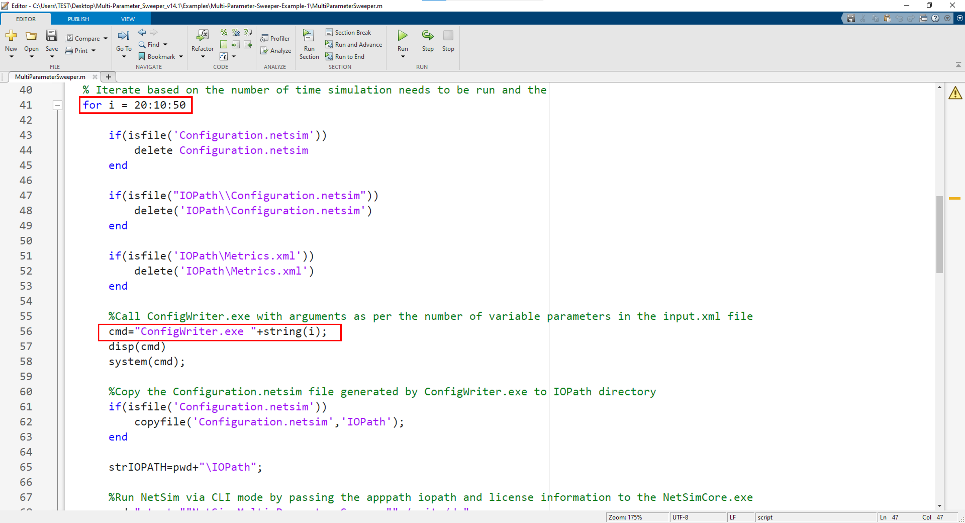
* A result.csv file is created and added with headings Y and Throughput (Mbps).
* A MetricsPrint.csv is created inside every output folder.
* For loop is set to iteratively run simulations for values starting from 20 to 50 in steps of 10.
* The value of the parameter Y in the current iteration is written to the result log file for analysis.
* The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.
* NetSim simulation is run via CLI mode by passing the apppath, iopath and license server information
* Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.

The MultiParameterSweeper.m matlab script modified for running simulations for different values of Y coordinates starting from 20 up to 50 in steps of 10 is shown below:



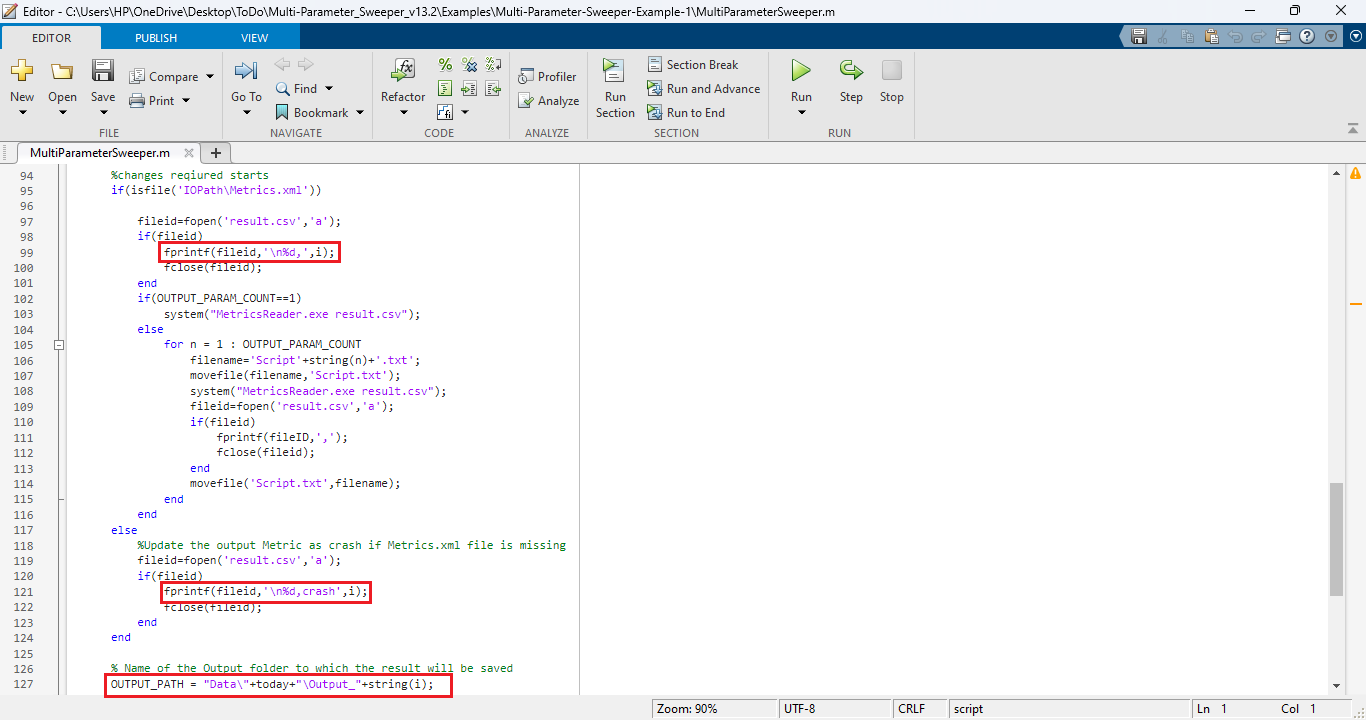
**Figure 17:** To Create result.csv file, added with headings Y and Throughput (Mbps) and NetSim installation Path and License information

* NETSIM\_PATH variable is set to the path of NetSim 64-bit binaries in the install directory or workspace in the system.
* LICENSE\_ARG variable is set to the license server details
* A result.csv file is created and added with headings Y and Throughput (Mbps).



**Figure 18:** Varying Distance and set license server information

* For loop is set to iteratively run simulations for values starting from 20 to 50 in steps of 10.
* The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.



**Figure 19:** Modify parameters in MultiParameterSweeper.m

* The value of the parameter Y in the current iteration is written to the result log file for analysis.
* Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.

1. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:

Graphical user interface, application, Word

Description automatically generated

**Figure 20:** Running MATLAB script using cmd prompt

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of UE.

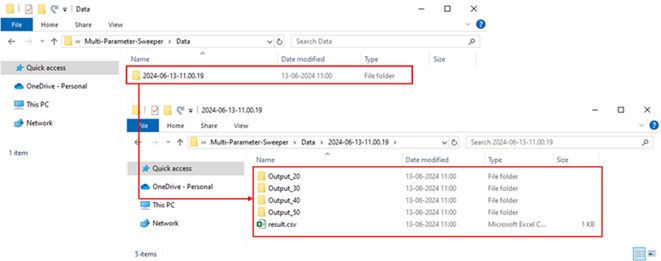
At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

Graphical user interface, application

Description automatically generated

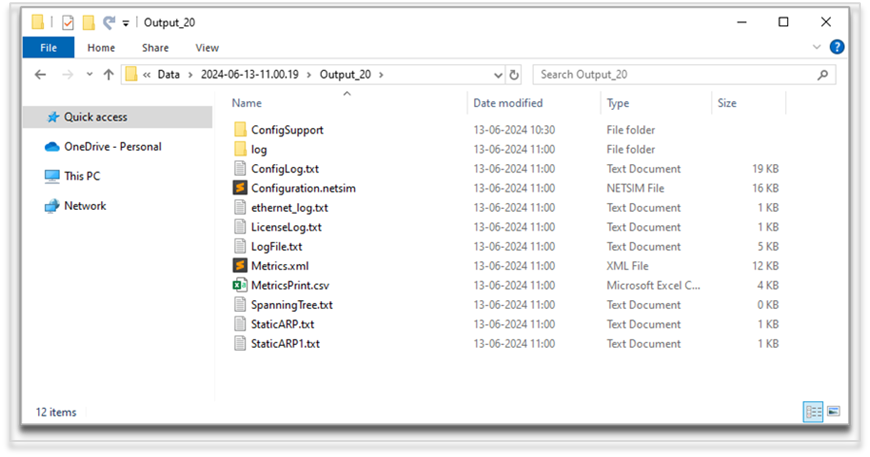
**Figure 21:** After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

* **Data**: Contains multiple folders corresponding to each simulation run, with its name including the value of the parameters in that iteration. The output folders will be created inside folder with name in the format Year-Month-Day-Hours-Minutes-Seconds.



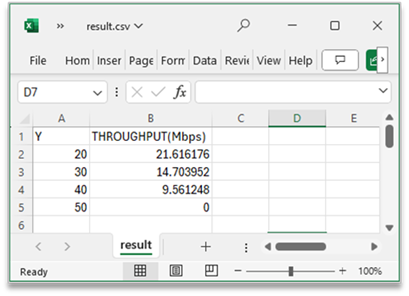
**Figure 22:** Based on distance Configuration.netsim files created in output folder

* Each folder contains the all the output files associated with the simulation run.



**Figure 23:** Each folder contains the all the output files

* **IOPath**: Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
* **Result.csv**: This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.



**Figure 24**: Distance Vs. Throughput obtained in result.csv

**Varying multiple network parameters**

In order to vary multiple network parameters during the multi-parameter sweep process each parameter in the input.xml file can be modified as {0},{1},{2},{3},..{n} respectively.

**Logging multiple output parameters**

Each output parameter that is to be logged should be part of the Script.txt file. However, the Script.txt file should contain only the details of one output parameter during the call to MetricsReader.exe.

To log multiple parameters, multiple script files can be used. If n output parameters are to be logged, then there can be script1.txt, script2.txt, script.txt in the sweeper folder.

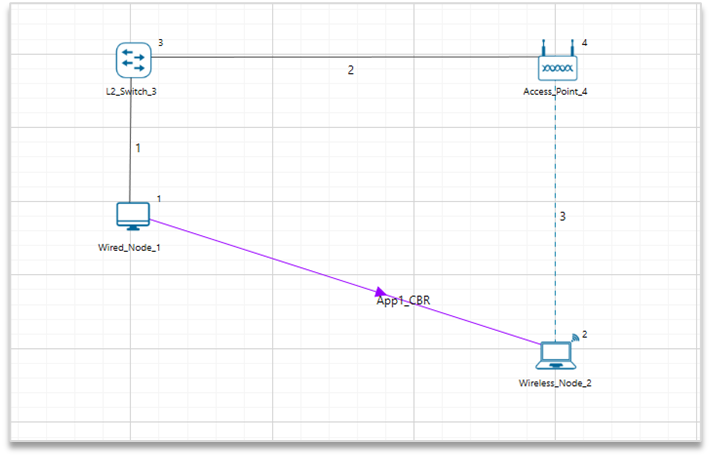
For Example, there can be two Script files as shown below:

Graphical user interface, application

Description automatically generated  
**Figure 25**: Multiple output parameters

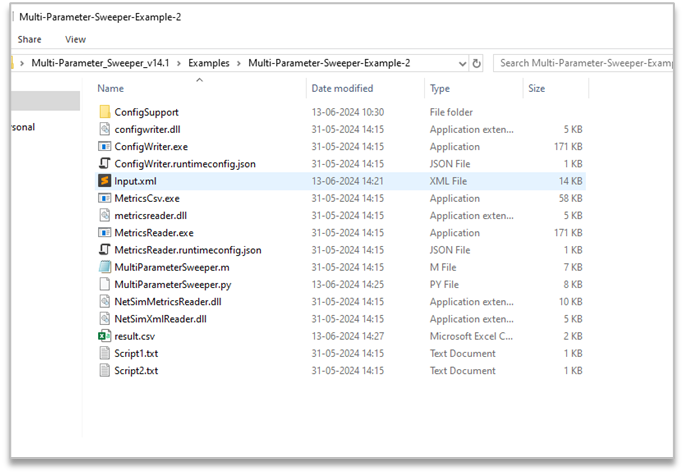
**Example 2: Modifying multiple input parameters and logging multiple output parameter**

1. Consider the following network WiFi network scenario in NetSim, comprising of a Wired Node, Router, Access Point and a Wireless Node.



**Figure 26:** Network Scenario

1. Consider a Grid Length of 50m X 50 m and distance between AP and Wireless node is 20m.
2. Set DCF as the medium access layer protocol under datalink layer properties of access\_ point and wireless node
3. WLAN Standard is set to 802.11n and No. of Tx and Rx Antennas is set to 1 in both access point and wireless node.
4. Channel Characteristics -> Path Loss only, Path Loss Model -> Log Distance and Path loss Exponent as 3. (Wireless Link Properties).
5. Click on Set Traffic tab and set CBR application with 50Mbps generation rate. (Set Inter Arrival Time: 233.6 (micro sec)).
6. Set transport protocol to UDP.
7. The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.
8. Refer to the Example 2 directory which is part of the project folder (Multi-Parameter-Sweeper\_v14.1\Examples\Multi-Parameter-Sweeper-Example-2).

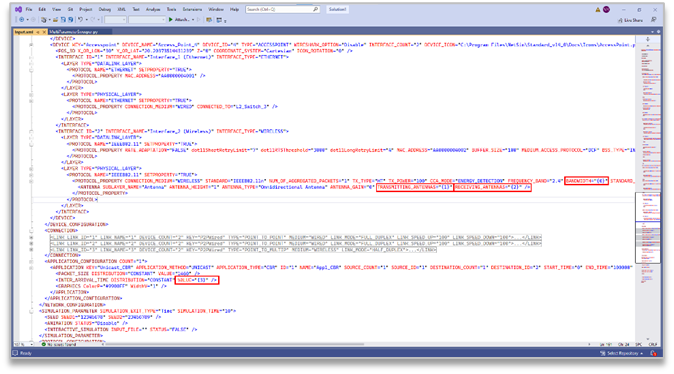


**Figure 27:** Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

1. In the Input.xml file the value of the input variables are modified as shown in the table below:

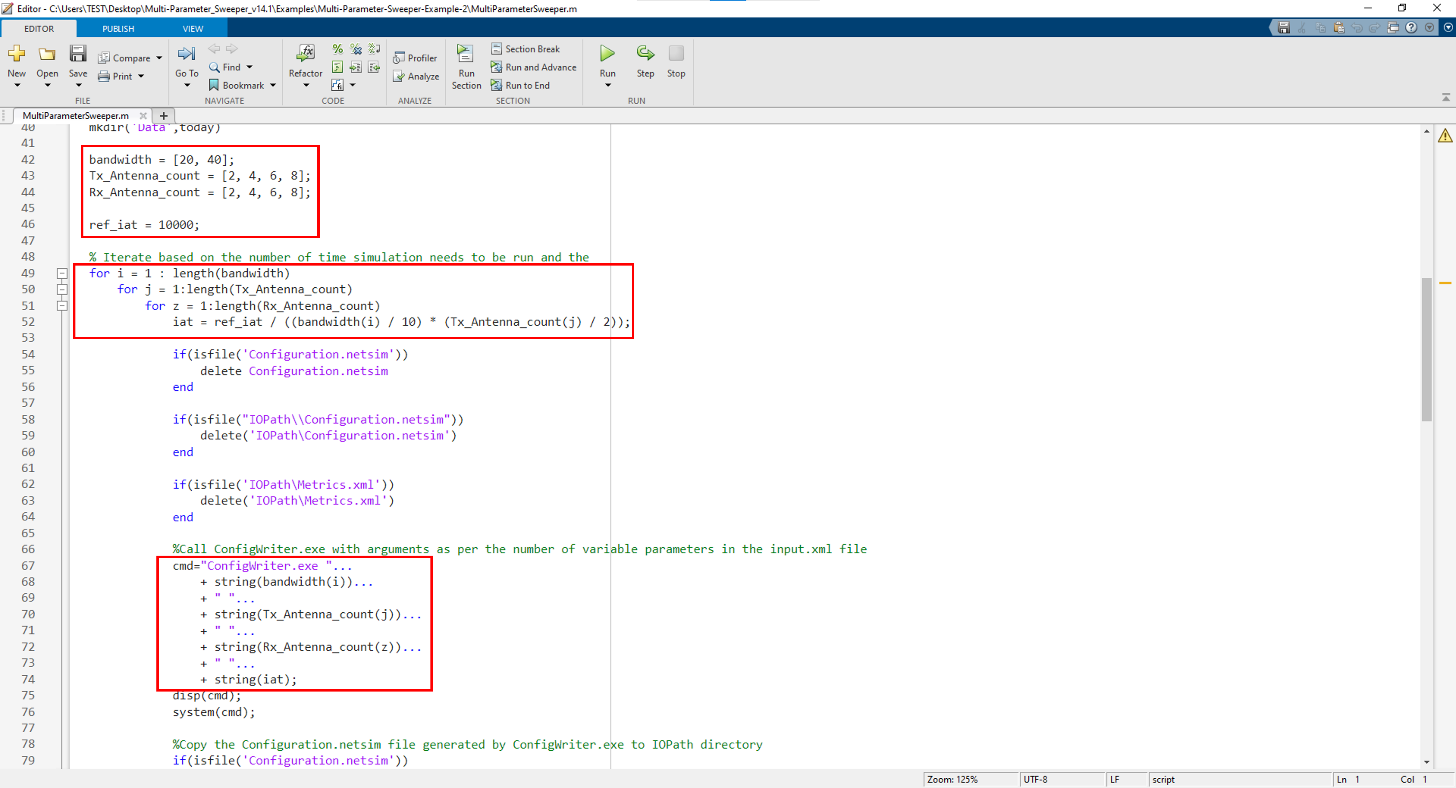
|  |  |
| --- | --- |
| Input Variables |  |
| Channel Bandwidth (MHz) | {0} |
| Transmitting Antennas | {1} |
| Receiving Antennas | {2} |
| Inter Arrival Time (Microseconds) | {3} |

**Table 3**: Variables are modified to the input.xml file



**Figure 28:** The above table 3 Variables are modified in input.xml file

1. The MATLAB script MultiParameterSweeper.m is modified to run simulation for all possible combinations of Bandwidth and Tx Antenna Count and Rx Antenna Count with the respective IAT that is calculated.



**Figure 29:** Modified MultiParameterSweeper.m based on input parameter

1. Multiple parameters are read from the Metrics.xml file and logged in the results.csv file along with the input parameters such as BANDWIDTH, TRANSMITTING\_ANTENNAS, RECEIVING\_ANTENNAS, INTER\_ARRIVAL\_TIME (micro sec).

|  |
| --- |
| Output Parameters |
| Throughput (Mbps) |
| Data Packets transmitted |

**Table 4:** User need to modify these two parameters in Script files

1. Two script text files namely Script1.txt and Script2.txt are created with information to read each of the parameters from the Metrics.xml file. The variable OUTPUT\_PARAM\_COUNT is set to 2 as per the number of Script files.

**Script1.txt**

Graphical user interface, text

Description automatically generated

**Figure 30**: The application throughput is to be logged for each simulation modified in Script1.txt

**Scritp2.txt**

Graphical user interface, text, application

Description automatically generated

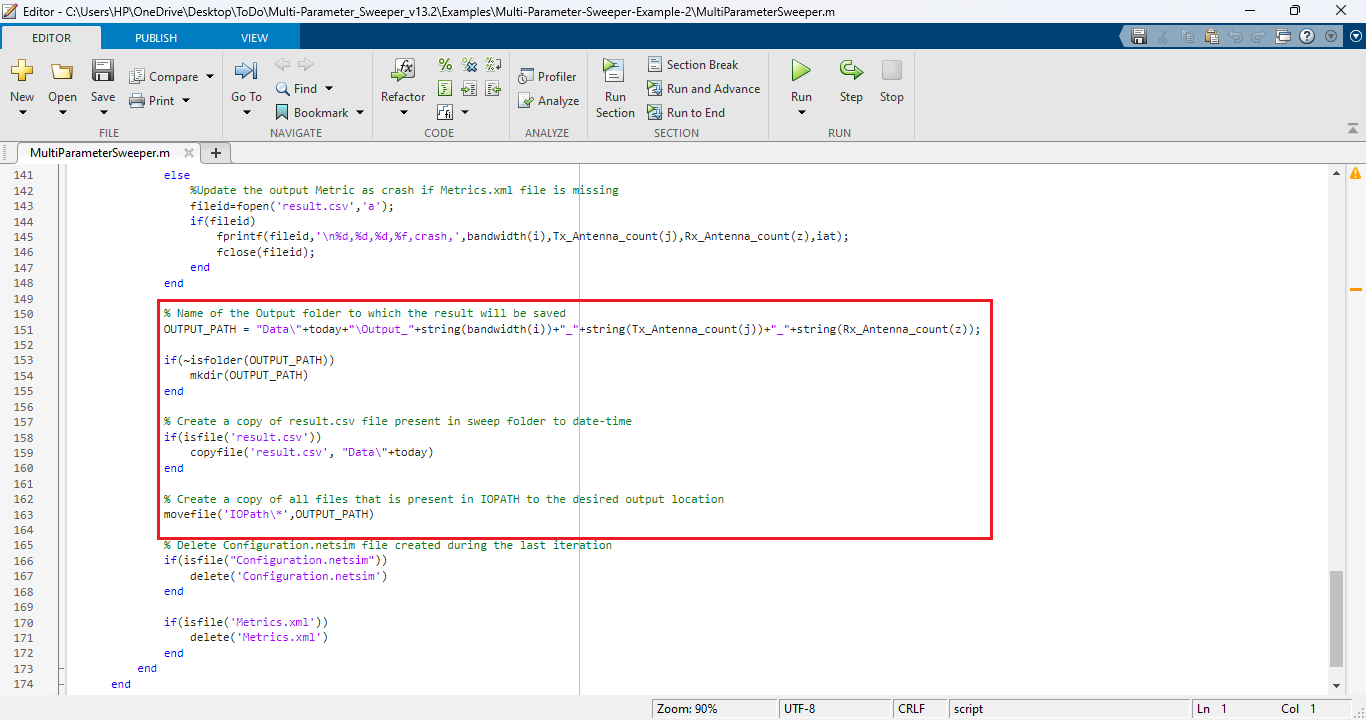
**Figure 31:** The Data Packets transmitted is to be logged for each simulation modified in Script2.txt

1. In the MATLAB script MultiParameterSweeper.m, MetricsReader is called to log each parameter specified in the script text files separating the entries with a comma (“,”). If simulation crashes, without generating the output Metrics.xml, then “crash” message is written to the log for each output parameter. The input parameters that were varied during each simulation run are also logged in the results.csv file.



**Figure 32:** Modify MATLAB script MultiParameterSweeper.m file

1. The simulation Configuration file and all the output files associated with each simulation run is saved to folders with name including the bandwidth and DL MIMO count values that were used during each simulation run.



**Figure 33:** Modify MATLAB script MultiParameterSweeper.m file

1. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:

**MATLAB Script:**

Graphical user interface, application, Word

Description automatically generated

**Figure 34:** Running MATLAB Script using cmd Prompt

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different combinations of input parameters.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

Graphical user interface, application

Description automatically generated

**Figure 35:** After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

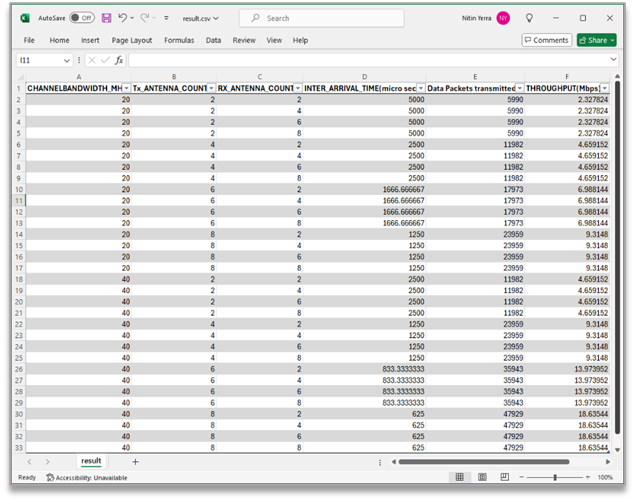
1. **Data**: The Data directory contains multiple output folders with the output files associated with each simulation run.

Graphical user interface, application

Description automatically generated

**Figure 36**: Based on input parameter Configuration.netsim and other files created in output folder

1. **IOPath**: Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
2. **Result.csv**: This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.

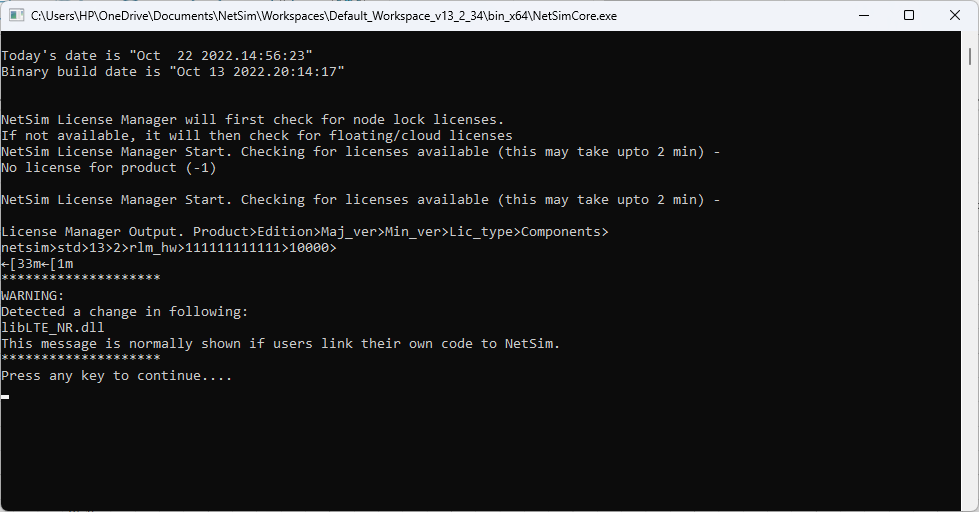


**Figure 37:** Based on script result stored in result.csv file

**Advanced use cases:**

1. Running the sweeper with binaries of a modified workspace.

When the source codes of a workspace are modified, a warning message will be displayed in the simulation console waiting for user interrupt, as shown below:



To suppress this message and user interrupt, so that the sweeper runs without requiring any intervention, follow the steps given below:

* + Open the current workspace location by going to Your Work-> Workspaces-> Open Workspace Location.
  + Go the bin\_x64 directory and rename the Checksum.txt file to say, Checksum\_backup.txt as shown below:

Graphical user interface, text, email

Description automatically generated

Now upon running any further simulations, warning messages will not be printed in the simulation console and no user intervention will be required.

1. Sweeping Configuration files that has other associated files

There can be cases where a Configuration file requires other supporting files such as:

* Mobility – Mobility text file.
* Static route text files
* ACL input
* SUMO configuration files in case of VANET, etc

In such cases, the associated files can also be placed in the sweeper folder which contains the input.xml file and code can be slightly modified to copy all associated files when copying the Configuration file to the IOPath.