DEVS-Graph Web-Application User Manual

Nirmal Patel

Introduction

DEVS-Graph is a graphical formalism to represent DEVS models. With the DEVS-Graph web-application users can draw the DEVS model using graphical user-interface, generate the JSON for the model so that it can be re-used in future. The users can also generate cadmium code for DEVS models from the DEVS-Graph. In this document, a general outline of how to use the web-application is given step by step.

Step 1: Open this URL in the browser of your choice:

https://devs-simulators.sce.carleton.ca/DEVS/

Step 2: Let's understand three main components of the application,

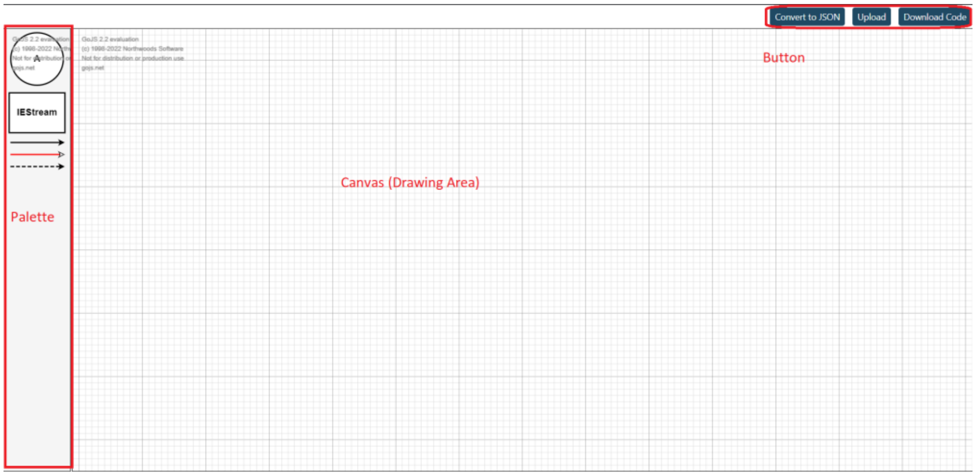
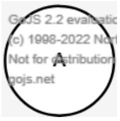
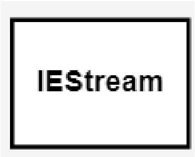


Figure 1: Overview of DEVS-Graph web-application

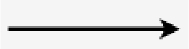
Palette: On left-hand side of the window, the palette is positioned. There are 5 elements in the palette which can be used to draw the DEVS model.

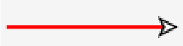


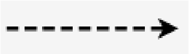
: The circle is used to define the state. The state name can be edited by clicking on the text inside the circle. By default, the circle has label "A". This text can be multi-lined label as well.



: The square is used to indicate external input source to the model. To define the type of the input, click on the text and press enter to type the label.

: The solid line arrow is used for defining external transition in the atomic models. The transition condition can be edited by clicking on the label of the link.

: The red arrow with white arrowhead is used for couplings between atomic and coupled models.

: The dashed line arrow is used for defining internal transition in the atomic models. The transition output can be edited by clicking on the label of the link.

Canvas: The canvas or the drawing area is used for drawing the models. To draw the model, drag and drop different components from the palette window. Some of the short-cuts for the canvas are as follows,

Ctrl + A: Select all the components of the canvas

Ctrl + - : Zoom-out the canvas layout (makes the components look smaller)

Ctrl + +: Zoom-in the canvas layout (makes the components look bigger)

Ctrl + C: Copy the selected components

Ctrl + V: Paste the copied components

Ctrl + G: Group the selected component, puts the selected components in a square box.

Ctrl + Shift + G: Ungroup the selected components.

Buttons: There are three buttons on the right-top part of the webpage. The buttons functions are as follows, Convert to JSON: Converts the drawn model in canvas to a raw JSON, which will be downloaded in the user’s machine. This JSON can be re-used to load the same model again in future.

Load: Upload JSON from the user’s machine to draw the model defined in the JSON file.

Download Code: To generate cadmium code for the DEVS model click on the download code button which will prompt for a name for the project and upon filling the name and clicking on download will download a zip file containing all the source files needed to run the model.

Example:

```
{ "class": "GraphLinksModel",
  "copiesArrays": true,
  "copiesArrayObjects": true,
  "linkFromPortIdProperty": "fromPort",
  "linkToPortIdProperty": "toPort",
  "modelData": {"position":"2112.238576250846 852.238728752538"},
  "nodeDataArray": [
{"text":"*idle\nTa=inf","figure":"Ellipse","size":"94 94","fill":"white","key":-
1,"loc":"2474.000000000001 1092.5058499991082","group":-3},
{"text":"bar\nTa=5","figure":"Ellipse","size":"98 97","fill":"white","key":-
2,"loc":"2474.000000000001 1312.5058499991082","group":-3},
{"text":"Lobby","isGroup":true,"color":"blue","I":[{"id":"I1","text":"in1\nint","
color":"red"}],"O":[{"id":"O1","text":"out1\nint","color":"red"},{"id":"O2","text
":"out2\nint","color":"red"}],"key":-3,"group":-13},
{"text":"*idle\nTa=inf","figure":"Ellipse","size":"93 92","fill":"white","key":-
4,"loc":"3216.500000000001 1085.8025975772794","group":-6},
{"text":"ready\nTa=4","figure":"Ellipse","size":"85 85","fill":"white","key":-
5,"loc":"3462.500000000001 1082.3025975772794","group":-6},
{"text":"Bar","isGroup":true,"color":"blue","I":[{"id":"I1","text":"in4\nint","co
lor":"red"}],"O":[{"id":"O1","text":"out3\nint","color":"red"}],"key":-
6,"group":-11},
{"text":"*idle\nTa=inf","figure":"Ellipse","size":"89 87","fill":"white","key":-
7,"loc":"3189.5 1379.417097577279","group":-9},
{"text":"ready\nTa=15","figure":"Ellipse","size":"91 90","fill":"white","key":-
8,"loc":"3470.5 1380.917097577279","group":-9},
{"text":"Table","isGroup":true,"color":"blue","I":[{"id":"I1","text":"in5\nint","
color":"red"}],"O":[{"id":"O1","text":"out4\nint","color":"red"}],"key":-
9,"group":-11},
{"text":"table\nTa=7","figure":"Ellipse","size":"91 87","fill":"white","key":-
10,"loc":"2668.000000000001 1166.5058499991087","group":-3},
{"text":"Restaurant","isGroup":true,"color":"blue","I":[{"id":"I1","text":"in2\nint","color":"red"},{"id":"I2","text":"in3\nint","color":"red"}],"O":[{"id":"O1","text":"out5\nint","color":"red"}],"key":-11,"group":-13},
{"text":"IEStream\ntype=int","figure":"Rectangle","size":"90 65","fill":"white","key":-12,"loc":"2215 1191.55","group":-13},
{"text":"top_model","isGroup":true,"color":"blue","I":[],"O":[{"id":"O1","text":"output\nint","color":"red"}],"key":-13}
],
  "linkDataArray": [
{"points":[2474,1139.5058499991085,2474,1149.5058499991085,2474,1201.7558499991085,2474,1201.7558499991085,2474,1254.0058499991085,2474,1264.0058499991085],"color":"black","from":-1,"to":-2,"fromPort":"B","toPort":"T","label":"in1?1"},
{"points":[2523,1312.5058499991085,2533,1312.5058499991085,2533,1092.5058499991085,2532,1092.5058499991085,2531,1092.5058499991085,2521,1092.5058499991085],"dash_array":[6,3],"from":-2,"to":-1,"fromPort":"R","toPort":"R","label":"out1!1"},
{"points":[3263,1085.802597577279,3273,1085.802597577279,3341.5,1085.802597577279,3341.5,1082.302597577279,3410,1082.302597577279,3420,1082.302597577279],"color":"black","from":-4,"to":-5,"fromPort":"R","toPort":"L","label":"in4?1"},
{"points":[3462.5,1039.802597577279,3462.5,1029.802597577279,3216.5,1029.802597577279,3216.5,1029.802597577279,3216.5,1029.802597577279,3216.5,1039.802597577279],"dash_array":[6,3],"from":-5,"to":-4,"fromPort":"T","toPort":"T","label":"out3!2"},
{"points":[3234,1379.417097577279,3244,1379.417097577279,3329.5,1379.417097577279,3329.5,1380.917097577279,3415,1380.917097577279,3425,1380.917097577279],"color":"black","from":-7,"to":-8,"fromPort":"R","toPort":"L","label":"in5?1"},
{"points":[3470.5,1335.917097577279,3470.5,1325.917097577279,3189.5,1325.917097577279,3189.5,1325.917097577279,3189.5,1325.917097577279,3189.5,1335.917097577279],"dash_array":[6,3],"from":-8,"to":-7,"fromPort":"T","toPort":"T","label":"out4!3"},
{"points":[2474,1045.5058499991085,2474,1035.5058499991085,2668,1035.5058499991085,2668,1074.2558499991085,2668,1113.0058499991085,2668,1123.0058499991085],"color":"black","from":-1,"to":-10,"fromPort":"T","toPort":"T","label":"in1?2"},
{"points":[2668,1210.0058499991082,2668,1220.0058499991082,2596.4874954223633,1220.0058499991082,2596.4874954223633,1092.5058499991082,2531,1092.5058499991082,2521,1092.5058499991082],"dash_array":[6,3],"from":-10,"to":-1,"fromPort":"B","toPort":"R","label":"out2!1"},
{"points":[2777.261423749155,1155.0270277827572,2787.261423749155,1155.0270277827572,2888.7787348417323,1155.0270277827572,2888.7787348417323,1197.844600180241,2990.2960459343094,1197.844600180241,3000.2960459343094,1197.844600180241],"color":"red","arrow_color":"red","fill_arrow":"red","from":-3,"to":-11,"fromPort":"O1","toPort":"I1","label":""},
{"points":[2777.261423749155,1206.777924999554,2787.261423749155,1206.777924999554,2888.7787348417323,1206.777924999554,2888.7787348417323,1259.595497397038,2990.2960459343094,1259.595497397038,3000.2960459343094,1259.595497397038],"color":"re
d","arrow_color":"red","fill_arrow":"red","from":-3,"to":-
11,"fromPort":"O2","toPort":"I2","label":""},
{"points":[3004.2960459343094,1193.844600180241,3004.2960459343094,1203.844600180
241,3113.238576250847,1203.844600180241,3113.238576250847,1149.0036994844404,3113
.238576250847,1094.1627987886397,3113.238576250847,1084.1627987886397],"color":"r
ed","arrow_color":"red","fill_arrow":"red","from":-11,"to":-
6,"fromPort":"I1","toPort":"I1","label":""},
{"points":[3004.2960459343094,1263.595497397038,3004.2960459343094,1253.595497397
038,3088.238576250846,1253.595497397038,3088.238576250846,1307.4145230928389,3088
.238576250846,1361.2335487886396,3088.238576250846,1371.2335487886396],"color":"r
ed","arrow_color":"red","fill_arrow":"red","from":-11,"to":-
9,"fromPort":"I2","toPort":"I1","label":""},
{"points":[3564.761423749155,1067.2873501802412,3564.761423749155,1077.2873501802
412,3564.761423749155,1217.844600180241,3613.6145782506846,1217.844600180241,3662
.4677327522145,1217.844600180241,3662.4677327522145,1207.844600180241],"color":"r
ed","arrow_color":"red","fill_arrow":"red","from":-6,"to":-
11,"fromPort":"O1","toPort":"O1","label":""},
{"points":[3575.761423749154,1354.358100180241,3575.761423749154,1344.35810018024
1,3575.761423749154,1205.844600180241,3619.114578250684,1205.844600180241,3662.46
77327522145,1205.844600180241,3662.4677327522145,1215.844600180241],"color":"red"
,"arrow_color":"red","fill_arrow":"red","from":-9,"to":-
11,"fromPort":"O1","toPort":"O1","label":""},
{"points":[2260.000000000001,1191.5499999999993,2270.000000000001,1191.5499999999
993,2312.1192881254237,1191.5499999999993,2312.1192881254237,1197.7779249995544,2
354.238576250846,1197.7779249995544,2364.238576250846,1197.7779249995544],"color"
:"red","arrow_color":"red","fill_arrow":"red","from":-12,"to":-
3,"fromPort":"R","toPort":"I1","label":""},
{"points":[3666.4677327522145,1211.844600180241,3676.4677327522145,1211.844600180
241,3755.1740417552746,1211.844600180241,3755.1740417552746,1212.594600180241,375
5.1740417552746,1213.344600180241,3745.1740417552746,1213.344600180241],"color":"
red","arrow_color":"red","fill_arrow":"red","from":-11,"to":-
13,"fromPort":"O1","toPort":"O1","label":""}
]}
```

Step 1: Open any text editor and copy-paste the content of the above file and save it as model.json

Step 2: Open URL https://devs-simulators.sce.carleton.ca/DEVS/ in the web-browser.

Step 3: Click on load button and upload the file created in step 1 (model.json).

Step 4: Click on load and the drawn model will be loaded in the canvas.

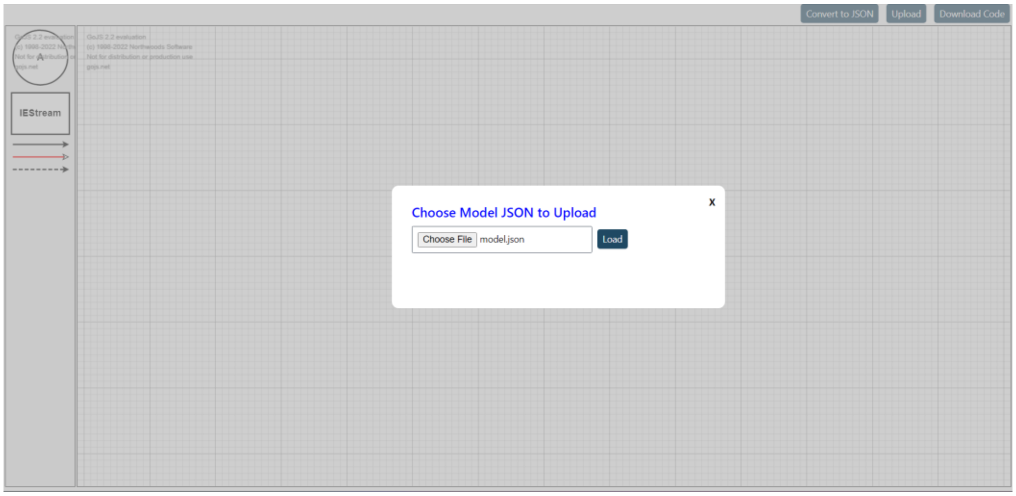


Figure 2: Upload file model.json to the web-application

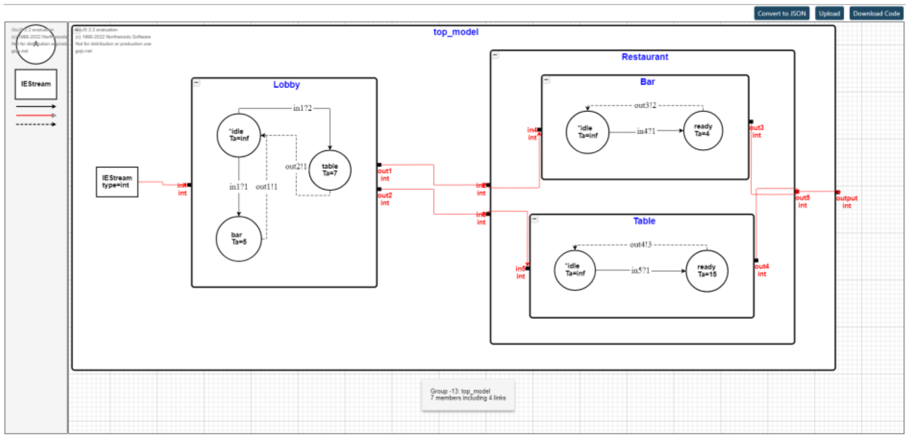


Figure 3: Canvas after loading model.json

Step 5: Click on "Download Code" button to download the source code, it will open a pop-up to specify the project name. We can put any name of our choice, in this example put name as 'hotel'.

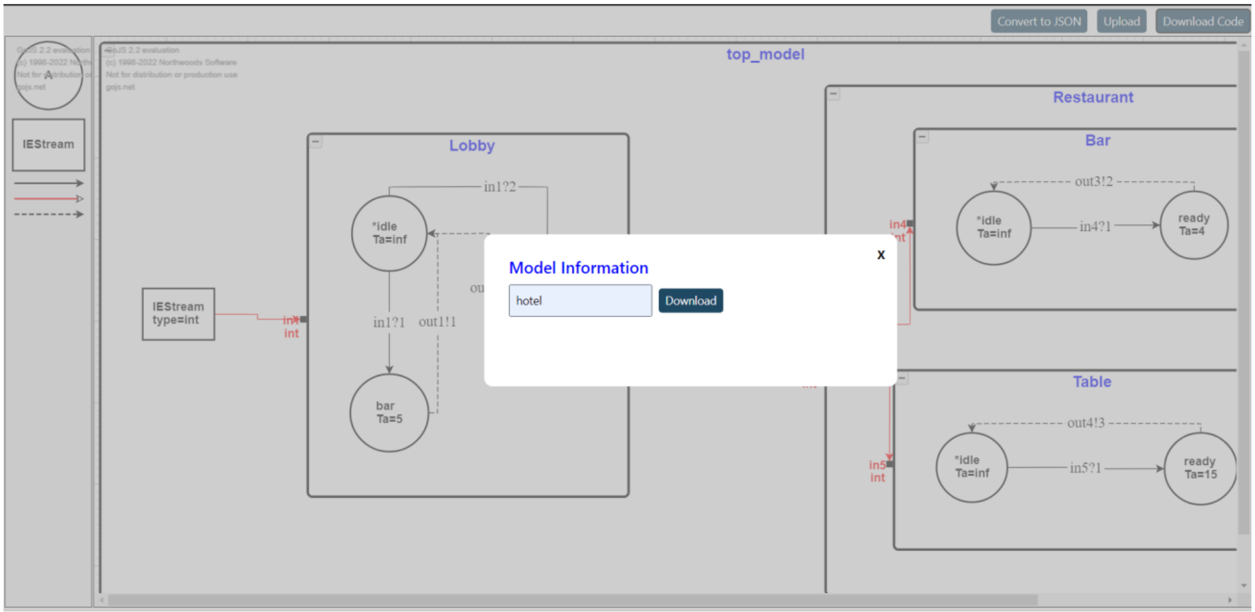
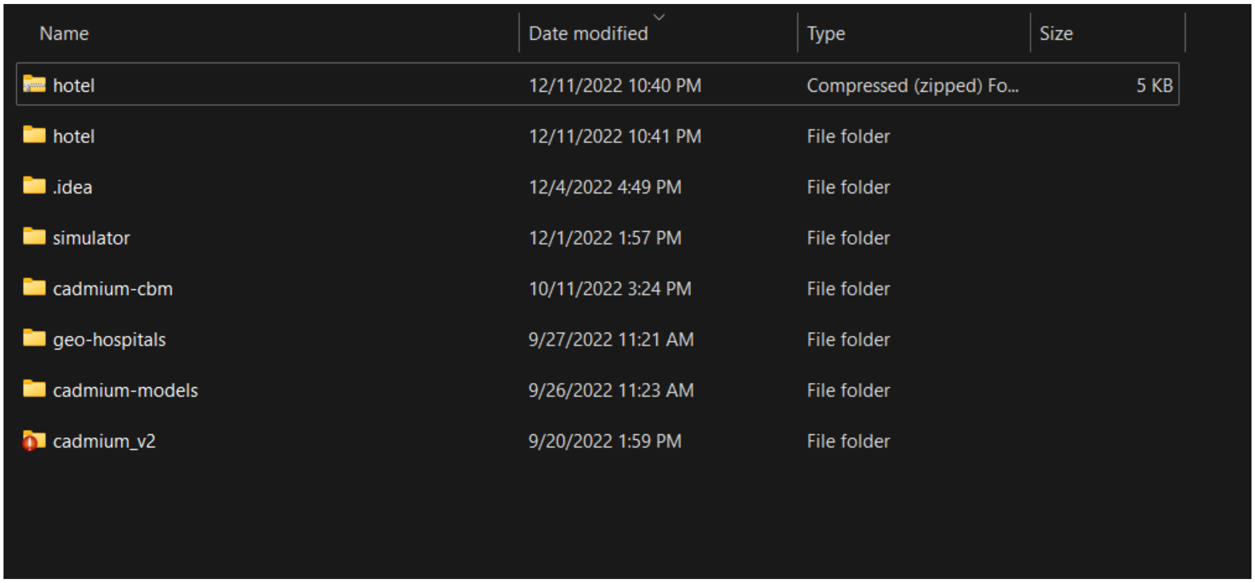


Figure 4: Download model named as hotel.

Step 6: After clicking on download, verify that a zip file named hotel.zip will be downloaded in the downloads folder. Extract this zip file.

Step 7: Place the extracted file in the folder in which the cadmium_v2 source code is located.



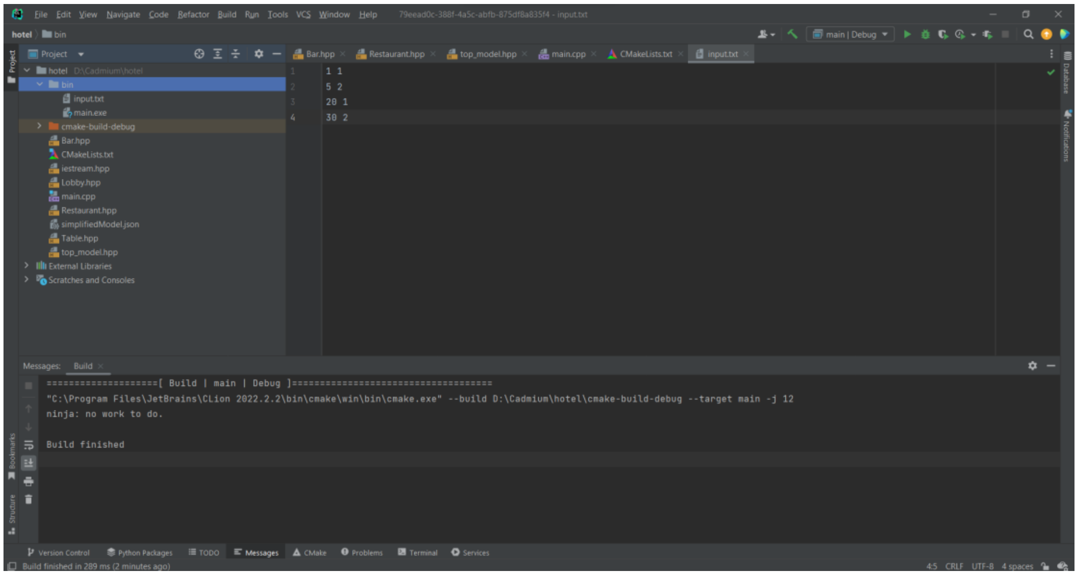
Name	Date modified	Type	Size
hotel	12/11/2022 10:40 PM	Compressed (zipped) Fo...	5 KB
hotel	12/11/2022 10:41 PM	File folder	
.idea	12/4/2022 4:49 PM	File folder	
simulator	12/1/2022 1:57 PM	File folder	
cadmium-cbm	10/11/2022 3:24 PM	File folder	
geo-hospitals	9/27/2022 11:21 AM	File folder	
cadmium-models	9/26/2022 11:23 AM	File folder	
cadmium_v2	9/20/2022 1:59 PM	File folder	

Figure 5: Place the hotel folder in the cadmium_v2's root directory.

Step 8: Open any IDE of your choice for c++ source code and build the project hotel using the CMakeLists.txt file provided. For this example, IDE Clion is used.

Step 9: Build the project, after successful build, in the hotel/bin folder executable file will be created. In case of windows the executable will be "main.exe".

Step 10: Create a sample input.txt file to run this executable for a simple input message sequence.



```
1 1
5 2
20 1
30 2
```

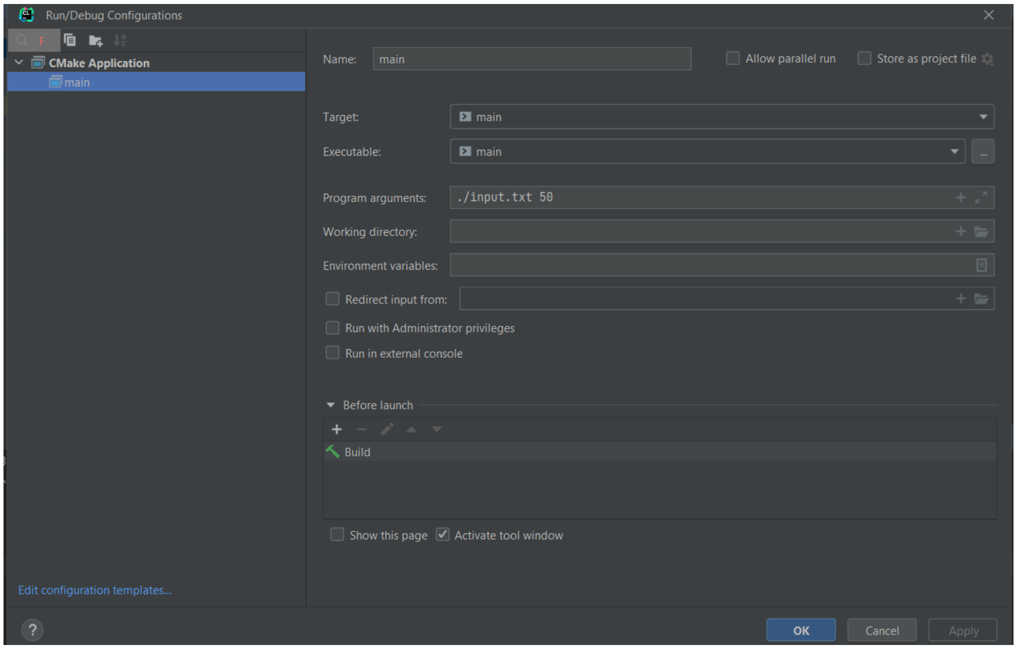
```
====================[ Build | main | Debug ]===================================
"C:\Program Files\JetBrains\CLion 2022.2.2\bin\cmake\win\bin\cmake.exe" --build D:\Cadmium\hotel\cmake-build-debug --target main -j 12
ninja: no work to do.

Build finished
```

Build finished in 289 ms (2 minutes ago)

Figure 6: Sample input.txt file created in the bin folder of the project

Step 10: Edit the run configuration to add run-time arguments for the simulator or optionally run this code using command line command “main.exe ./input.txt 50”.



Name: main

□ Allow parallel run □ Store as project file

Target: main

Executable: main

Program arguments: `./input.txt 50`

Working directory:

Environment variables:

□ Redirect input from:

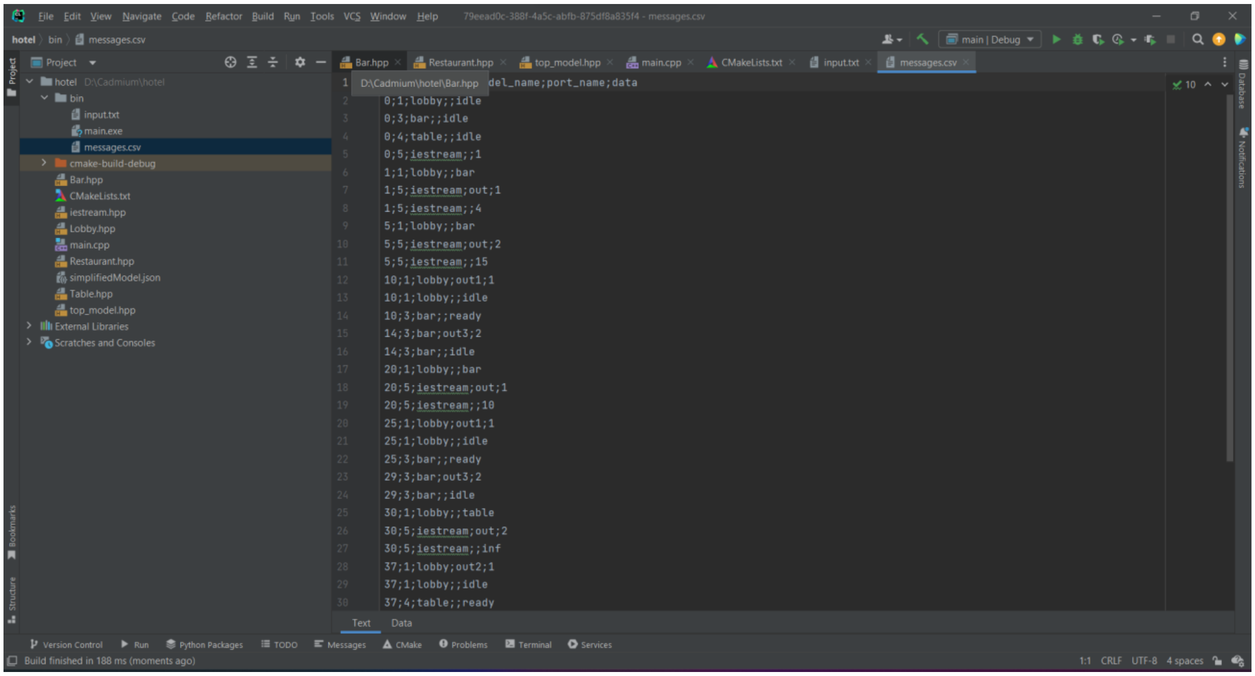
□ Run with Administrator privileges

□ Run in external console

□ Show this page ☑ Activate tool window

Figure 7: Add Program arguments ./input.txt 50

Step 11: Run the current configuration of the program and after successful run an output file named messages.csv will be created in the hotel/bin folder.



```
del_name;port_name;data
0;1;lobby;;idle
0;3;bar;;idle
0;4;table;;idle
0;5;iestream;;1
1;1;lobby;;bar
1;5;iestream;out;1
1;5;iestream;;4
5;1;lobby;;bar
5;5;iestream;out;2
5;5;iestream;;15
10;1;lobby;out1;1
10;1;lobby;;idle
10;3;bar;;ready
14;3;bar;out3;2
14;3;bar;;idle
20;1;lobby;;bar
20;5;iestream;out;1
20;5;iestream;;10
25;1;lobby;out1;1
25;1;lobby;;idle
25;3;bar;;ready
29;3;bar;out3;2
29;3;bar;;idle
30;1;lobby;;table
30;5;iestream;out;2
30;5;iestream;;inf
37;1;lobby;out2;1
37;1;lobby;;idle
37;4;table;;ready
```

Figure 8: Sample output file for this example with the messages in csv format.