

[01/20/2024]

FLoRa installation guide

Attending

Marcos Alexandre Moreira Seraphim

Rayane Araujo lima

Antonio Oliveira - JR

Announcements

This document is crafted to assist telecommunications and related academic professionals in seamlessly installing, configuring, and extracting results from the Flora simulation environment using the Omnet++ tool. Its purpose is to streamline the process, making it more accessible and efficient for users engaged in telecommunications studies and research.

- 1- step, download the tool we use

<https://flora.aalto.fi/>

<https://omnetpp.org/>

- 2- With the tool already downloaded, unzip the file, I recommend that you leave the folder for this program directly in C:

- 2.1 installation file



omnetpp-6.0-windows-x86_64.zip

- 2.2 initialize this file



mingwenv.cmd

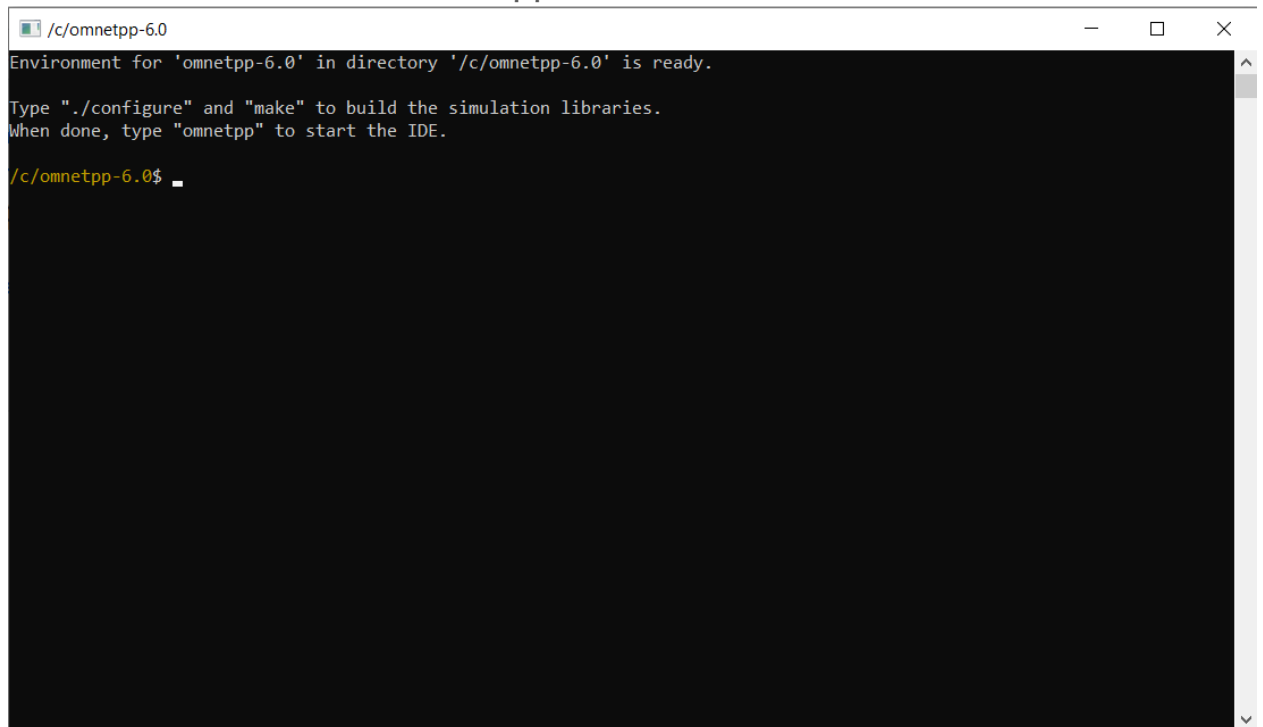
- 3- This process will take a few minutes, machines with good processor and memory performance usually do this in 5 minutes

```
C:\Windows\system32\cmd.exe
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\exceptions.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\help.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\hooks.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\models.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\packages.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\sessions.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\status_codes.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\structures.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\requests\utils.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\resolvelib\_init_.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\resolvelib\compat\collections_abc.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\resolvelib\providers.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\resolvelib\reporters.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\resolvelib\resolvers.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\resolvelib\structs.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\six.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\_init_.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\asyncio.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\utils.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\after.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\before.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\before_sleep.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\compat.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\nap.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\retry.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\stop.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\tornadoweb.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\tenacity\wait.py
Extracting mingw64\lib\python3.8\site-packages\pip\_vendor\toml\_init_.py
```

- 4- At the end of step 3, if a window does not automatically open, simply search for this program omNeT++ 6.0 Shell



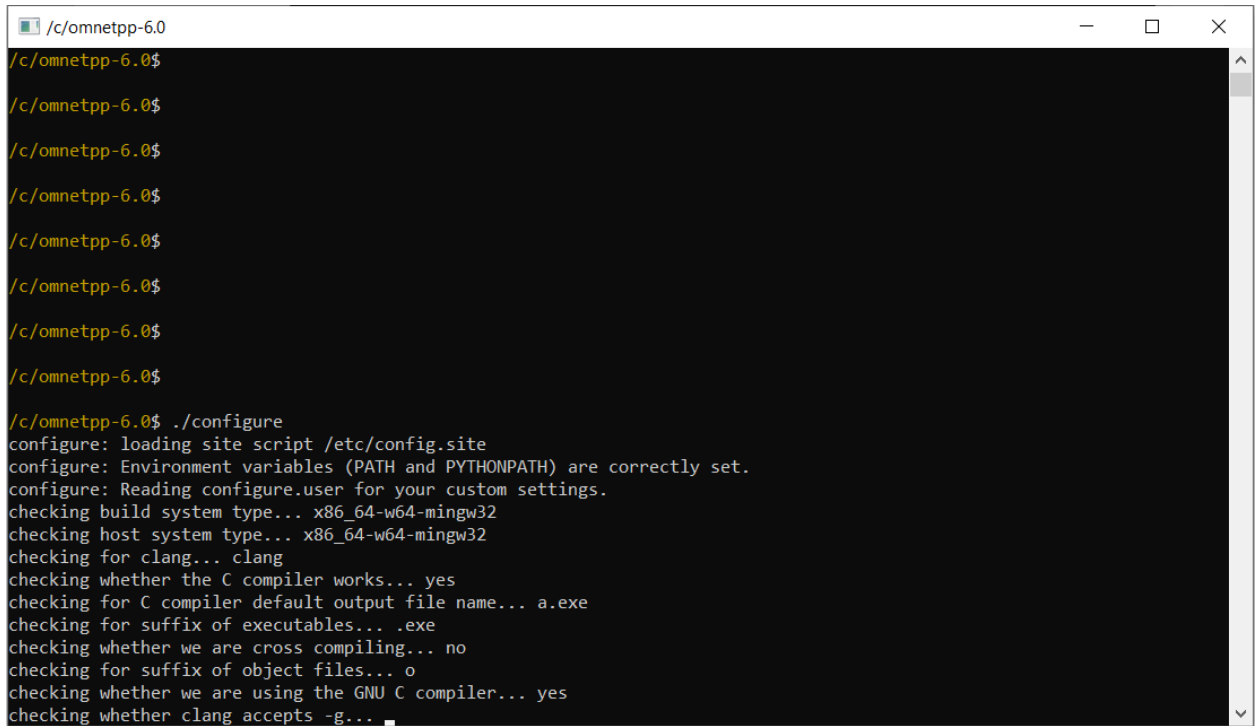
This is the window that should appear



```
/c/omnetpp-6.0
Environment for 'omnetpp-6.0' in directory '/c/omnetpp-6.0' is ready.
Type "./configure" and "make" to build the simulation libraries.
When done, type "omnetpp" to start the IDE.
/c/omnetpp-6.0$
```

- 5- When this window opens, type the following command `./configure` as in the image below.

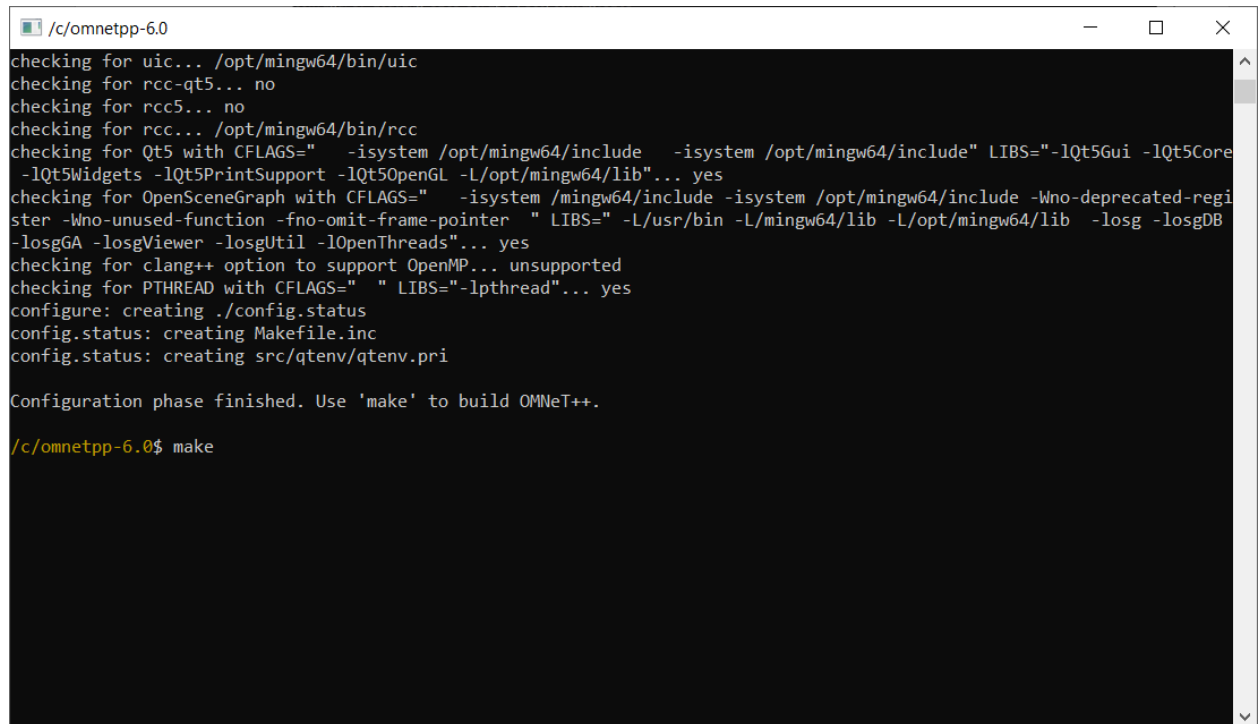
Note this process takes a little time, don't worry, approximately 10 minutes.



```
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$  
/c/omnetpp-6.0$ ./configure  
configure: loading site script /etc/config.site  
configure: Environment variables (PATH and PYTHONPATH) are correctly set.  
configure: Reading configure.user for your custom settings.  
checking build system type... x86_64-w64-mingw32  
checking host system type... x86_64-w64-mingw32  
checking for clang... clang  
checking whether the C compiler works... yes  
checking for C compiler default output file name... a.exe  
checking for suffix of executables... .exe  
checking whether we are cross compiling... no  
checking for suffix of object files... o  
checking whether we are using the GNU C compiler... yes  
checking whether clang accepts -g... _
```

6- After this process type **make** as in the image below.

Note this process takes a little time, don't worry, approximately 10 minutes.



```
/c/omnetpp-6.0
checking for uic... /opt/mingw64/bin/uic
checking for rcc-qt5... no
checking for rcc5... no
checking for rcc... /opt/mingw64/bin/rcc
checking for Qt5 with CFLAGS=" -isystem /opt/mingw64/include -isystem /opt/mingw64/include" LIBS="-lQt5Gui -lQt5Core
-lQt5Widgets -lQt5PrintSupport -lQt5OpenGL -L/opt/mingw64/lib"... yes
checking for OpenSceneGraph with CFLAGS=" -isystem /mingw64/include -isystem /opt/mingw64/include -Wno-deprecated-regi
ster -Wno-unused-function -fno-omit-frame-pointer " LIBS="-L/usr/bin -L/mingw64/lib -L/opt/mingw64/lib -log -logDB
-logGA -logViewer -logUtil -lOpenThreads"... yes
checking for clang++ option to support OpenMP... unsupported
checking for PTHREAD with CFLAGS=" " LIBS="-lpthread"... yes
configure: creating ./config.status
config.status: creating Makefile.inc
config.status: creating src/qt5env/qt5env.pri

Configuration phase finished. Use 'make' to build OMNeT++.

/c/omnetpp-6.0$ make
```

7- When the process ends to call the application just type **omnetpp**

```
/c/omnetpp-6.0
MobileNode.cc
OsgEarthScene.cc
RambleNode.cc
WaypointTrackerNode.cc
Creating executable: out/clang-debug//osg-earth_dbg.exe
==== Compiling osg-indoor ====
fallback.cc
OsgScene.cc
Person.cc
Creating executable: out/clang-debug//osg-indoor_dbg.exe
==== Compiling osg-satellites ====
ChannelController.cc
Clock.cc
fallback.cc
GroundStation.cc
OsgEarthScene.cc
Satellite.cc
Creating executable: out/clang-debug//osg-satellites_dbg.exe
==== Compiling wiredphy ====
RxAtEnd.cc
RxAtStart.cc
Sink.cc
Source.cc
Tx.cc
ProgressInfo_m.cc
Creating executable: out/clang-debug//wiredphy_dbg.exe

Now you can type 'omnetpp' to start the IDE.

/c/omnetpp-6.0$ omnetpp
```

8- wait to open the application

```
/c/omnetpp-6.0
WaypointTrackerNode.cc
Creating executable: out/clang-debug//osg-earth_dbg.exe
==== Compiling osg-indoor ====
fallback.cc
OsgScene.cc
Person.cc
Creating executable: out/clang-debug//osg-indoor_dbg.exe
==== Compiling osg-satellites ====
ChannelController.cc
Clock.cc
fallback.cc
GroundStation.cc
OsgEarthScene.cc
Satellite.cc
Creating executable: out/clang-debug//osg-satellites_dbg.exe
==== Compiling wiredphy ====
RxAtEnd.cc
RxAtStart.cc
Sink.cc
Source.cc
Tx.cc
ProgressInfo_m.cc
Creating executable: out/clang-debug//wiredphy_dbg.exe

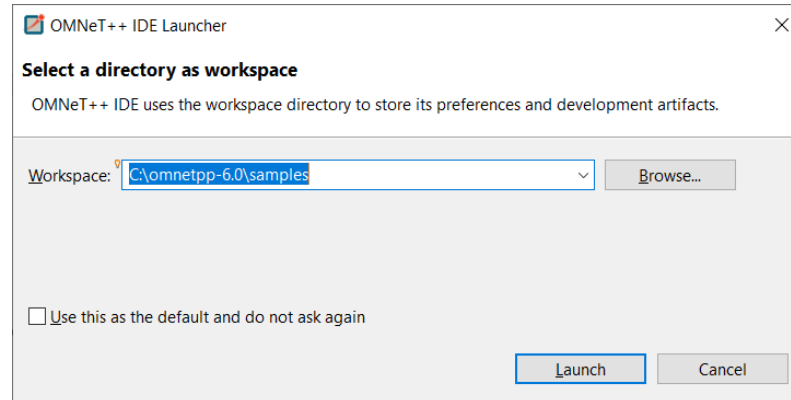
Now you can type 'omnetpp' to start the IDE.

/c/omnetpp-6.0$ omnetpp
Starting the OMNeT++ IDE...

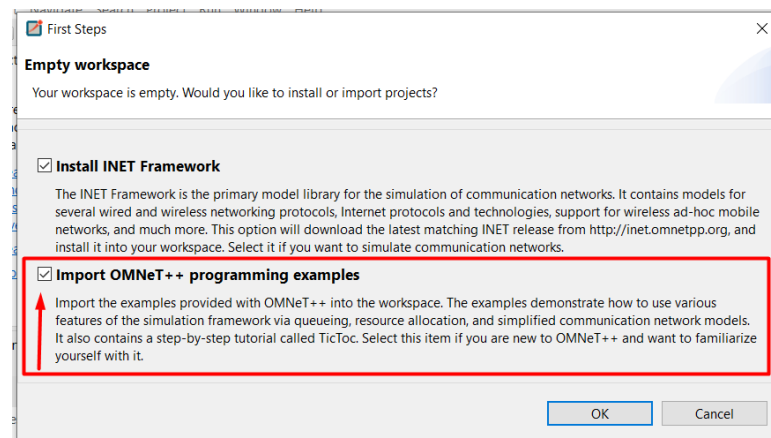
/c/omnetpp-6.0$
```

The logo for OMNeT++ 6, featuring the text "OMNeT++" in blue and "6" in large orange, with "THE OPEN SIMULATOR" in smaller blue text below. The logo is set against a background of interlocking puzzle pieces in white and orange. At the bottom, it says "Academic Edition - not for commercial use".

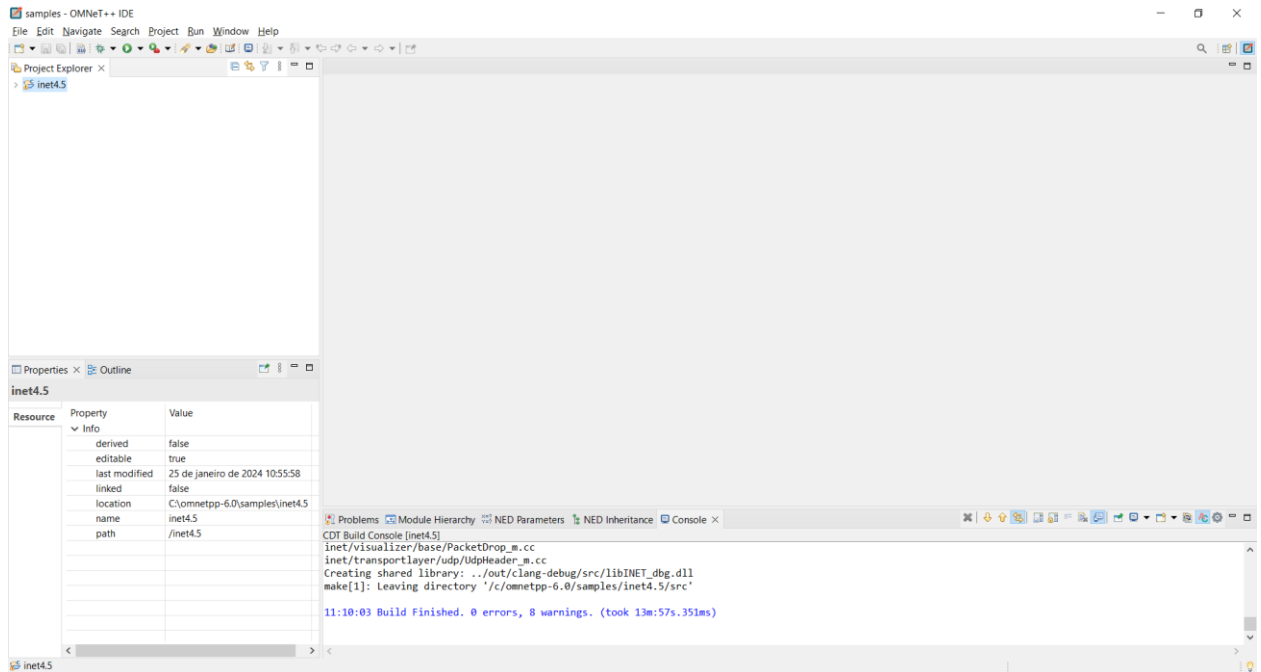
9- In this step, just follow the environment already created "samples", just click on **Launch**



10- After following the previous step, this window will automatically appear to start, my recommendation is that you initially install only **INET FRAMEWORK**, the other option that is marked load other models, uncheck this option, this will make it easier to launch into the flora environment



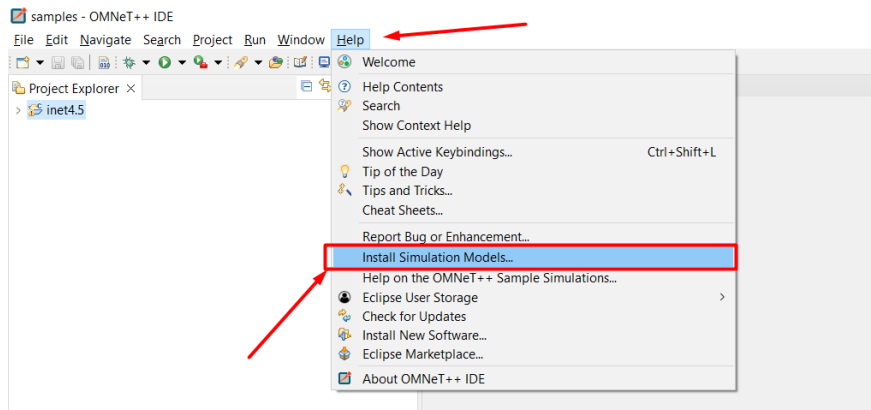
11- After finishing this step, your work area in omnet will look like this




12- Installing the Flora Environment

12.1 - In the upper commands, go to HELP

12.2- Next comes INSTALL SIMULATION MODELS



12.3- install these two templates **FLORA** and **INET FRAMEWORK 4.4.1**

 Install Simulation Models ×

Install Simulation Models
Please select a model and edit the options below

Model

Name	Version	Description
INET Framework 4	4.5.2	Protocols and applications for Internet - recommended (for OMNeT++ 6.x)
Simu5G	1.2.2	A Simulator for 5G Networks
SimuLTE	1.2.0	A Simulator for LTE Networks
FLoRa	1.1.0	A Framework for LoRa (Long Range) networks
INET Framework 4	4.4.1	Protocols and applications for Internet (for OMNeT++ 6.x)
INET Framework 4	4.3.9	Protocols and applications for Internet (for OMNeT++ 6.x)
INET Framework 4	4.2.10	Protocols and applications for Internet (for OMNeT++ 5.x and 6.x)
INET Framework 3	3.8.3	Protocols and applications for Internet - legacy (use INET 4 for new projects)

Other simulation models are available for download in the community [catalog](#).

Description

Options

Project name:

☒ Use default location

Location:

13- At the end of the processes, this will be your work area

