**How to Install Castalia**

**General**

Castalia is based on [OMNeT++](http://www.omnetpp.org). The latest version (Castalia 3.3) works with OMNeT versions **4.3** to **4.6** . Note that **the latest versions of OMNeT (5.x) are not compatible** with Castalia.

We recommend a **Linux or Mac OS X** system. Castalia has been tested with Ubuntu 14 and 16 and OS X 10.9. Note that Castalia is designed **as a command line tool**. Even though some people have adapted it to work with the OMNeT graphical IDE, it is not recommended (nor supported) to be used with it. The instructions given henceforth refer to an **Ubuntu** environment, but they should also work on a Mac.

**Installing OMNeT++**

For OMNeT installation troubleshooting and a detailed list of Linux distributions supported refer to OMNeT’s Installation guide ([pdf link](https://omnetpp.org/doc/omnetpp/InstallGuide.pdf)). Note that OMNeT’s Installation guide is for versions 5.x, so some things might be different or not needed for versions 4.x. Following are some basic steps to install OMNeT, that assume you have already installed the gcc compiler and other build tools in your system (in Ubuntu install build-essential and in Mac install either cmd line developer tools or XCode. OMNeT’s installation guide contains more info on this).

Get the source code. This link works as of March 2017: [OMNeT++ 4.6 (source + IDE, tgz)](https://omnetpp.org/component/jdownloads/send/32-release-older-versions/2290-omnet-4-6-source-ide-tgz)

If the link does not work just go to the OMNeT’s website and find how to download the source code for version 4.6 (or earlier down to 4.3) The zipped source code for version 4.6 is a large file (188MB) so it might take some time to download. Place it in your home directory.

Untar and unzip the source file:

$ tar xvfz omnetpp-4.6-src.tgz

A directory named omnetpp-4.6 will be created.

Set environment variables by typing (assuming you are using bash as your shell)

$ export PATH=$PATH:~/omnetpp-4.6/bin

$ export LD\_LIBRARY\_PATH=~/omnetpp-4.6/lib

**Also add the above two export commands at the end of your .bash\_profile file**.

You are now ready to build OMNeT:

$ cd omnetpp-4.6/

$ NO\_TCL=1 ./configure

$ make

The last command will take a few minutes to complete. You are now done building OMNeT. Castalia does not use the Tcl functionality so we opt to build OMNeT without it. The installation process can be easier if Tcl in not required. If you wish, you can try build OMNeT with Tcl.

Make sure that OMNeT++ is in the path . For example you can try:

$ which opp\_makemake

/home/NICTA/aboulis/omnetpp-4.6/bin/opp\_makemake

**Installing Castalia**

Get the [source code from GitHub](https://github.com/boulis/Castalia).

If you’ve downloaded a compressed file (instead of cloning the project) then untar and unzip it:

$ tar –xvzf Castalia-master.tar.gz

A new directory will be created, named Castalia/. In there you can find another directory named Castalia, and the User Manual and this installation guide in various forms. You can copy the inner Castalia directory to some other directory in your home dir, if you do not like having one Castalia dir nested in another. You are ready to build Castalia:

$ cd Castalia/Castalia (or cd Castalia-yourchosenname)

$ ./makemake

Wait for a few seconds till the script ends[[1]](#footnote-1). This automatically generates a Makefile that you can use to build Castalia. Type:

$ make

Wait again for some time until everything is built. Check that the soft link CastaliaBin is created in Castalia/Castalia. You have now successfully built Castalia!

Refer to the User’s Manual to start running simulations.

**Building Castalia with your additional custom source code**

Castalia is built for extension. Users are encouraged to do their custom modifications and add their own modules. If you have created a new module with .cc, .h, .ned, and .msg files, first make sure that the files and directory structure follow the module structure.

For example, place files for a new MAC module named SuperMAC in src/node/communication/mac/superMAC. Notice that we keep the convention of directory names starting with small case letters (this comes from the OMNeT package convention).

If you are using external libraries you have to edit the script makemake by adding some options in the EXTOPTS variable. More specifically you can:

1. Use the -I includedir option to specify the include path for any external header files[[2]](#footnote-2)
2. Use the -Ldir option to specify the directory of the external library
3. Use the -llibrary option to specify the name of the external dependency.

To rebuild Castalia, simply go on the Castalia top directory and type:

$ make clean

$ ./makemake

$ make

1. If the access to the script is refused, make sure you have the right permissions to the file. If not, type chmod u+x makemake and then try again [↑](#footnote-ref-1)
2. Any files outside of the Castalia/src/ directory tree (except the system and OMNeT++ headers which are always included automatically). [↑](#footnote-ref-2)