How to Install Castalia

General

Castalia is based on OMNeT++. Castalia 3.0 works with OMNeT versions **4.0** or **4.1**. Older versions of Castalia require older versions of OMNeT. If you wish to install an older version of Castalia please refer to the old Installation Guide found in the Castalia website.

We recommend a **Linux** system. OMNeT++ can also be installed in Windows or in Cygwin under Windows. Castalia however is *not* supported in Windows. The instructions given henceforth refer to a **Linux (or Cygwin)** environment.

Installing OMNeT++

For OMNeT installation troubleshooting and a detailed list of Linux distributions supported refer to OMNeT's Install Guide (pdf) . If a suitable version of OMNeT is already installed in your machine then you just need to set the environment variables described below, pointing to the directories OMNeT is installed. Here we assume a fresh installation.

Get the source code. At the moment of writing the latest version is: **OMNeT++ 4.1 (source + IDE, tgz)**

An easy way to get the tgz file into your system is go in your home dir and type:

```
$ wget http://www.omnetpp.org/omnetpp/doc_download/2217-omnet-41-source--ide-tgz
```

This is a large file (148 MB) so it might take some time to download.

Untar and unzip the source file:

```
$ tar xvfz omnetpp-4.1-src.tgz
```

A directory named `omnetpp-4.1` will be created.

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

```
$ export PATH=$PATH:~/omnetpp-4.1/bin
$ export LD_LIBRARY_PATH=~/omnetpp-4.1/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-4.1/
$ 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-4.1/bin/opp_makemake
```

Installing Castalia

Get the source code from http://castalia.npc.nicta.com.au/ .

Assume the file you downloaded is named Castalia-3.0.tar.gz

Untar and unzip the source code:

```
$ tar -xvzf Castalia-3.0.tar.gz
```

A new directory will be created, named `Castalia-3.0/`.You are ready to build Castalia:

```
$ cd Castalia-3.0/
$ ./makemake
```

Wait for a few seconds till the script ends[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-3.0/`. You have now successfully built Castalia!

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

[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

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]
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
```

[2] Any files outside of the Castalia/src/ directory tree (except the system and OMNeT++ headers which are always included automatically).