

Installing CompuCell on Linux systems

The list of prerequisites for compiling CompuCell3d include:

1. xercesc library and development files
2. python + development files
3. qt version 4 or higher
4. cmake version 2.4 or higher
5. swig version 1.3.27 or higher
6. vtk version 5 or higher

We are using ubuntu/debian systems and all the above packages are available through package manager. The only exception might be vtk which sometimes needs to be installed from source and we include detailed instructions how to do that. If for some reason qt4 is unavailable through your package manager you may follow the instructions below.

We came across several distributions that do not provide all above packages and we therefore recommend that in case of any dependency problems you follow the procedure below (especially for those packages that seem to be missing or misconfigured).

Installing cmake from source

We assume that there is no cmake installed on your system.

From www.cmake.org website download cmake-2.4.*.tar.gz package for platform Unix Linefeed Source. We assume that there is no cmake installed on your system

the current link as of writing this document is
(<http://www.cmake.org/files/v2.4/cmake-2.4.6.tar.gz>)

Unpack this tarball:

```
tar -zxvf cmake-2.4.6.tar.gz
```

and go to the unpacked directory:
cd cmake-2.4.6

To begin compilation process use the following command syntax:

```
./bootstrap -prefix=<installation_directory>
```

e.g.

```
./bootstrap -prefix=/usr/local/cmake
```

After bootstrapping is finished you will be asked to type make:

```
make
```

and once compilation is done you would type

```
make install
```

Now cmake is installed on your system.

Add path to your PATH system variable by appending the following line to your .bashrc file

```
export PATH=/usr/local/cmake/bin:$PATH
```

REMARK: If you installed cmake from source you will probably have to install VTK

from source as well to avoid possible incompatibilities of VTK with cmake. Compiling VTK using same version of cmake you will use to compile CompuCell3D should solve potential problems.

Installing Qt 4.1 (you may skip this section if you managed to install qt4 through package manager)

This part may take some time as Qt is quite big library.

Download open source X11 version of Qt 4.1 package from <http://www.trolltech.com/download/>
The exact link is the following: <http://www.trolltech.com/download/qt/mac.html>
Once you have qt file on your hard drive (I downloaded qt-mac-opensource-src-4.1.1.tar.gz) type

```
tar -zxvf qt-x11-opensource-src-4.1.1.tar.gz
cd qt-x11-opensource-src-4.1.1
./configure -prefix=/usr/local/qt
make
sudo make install
```

Do not forget to add to your .bashrc these lines
export PATH=/usr/local/qt/bin:\$PATH
export QMAKESPEC=linux-g++

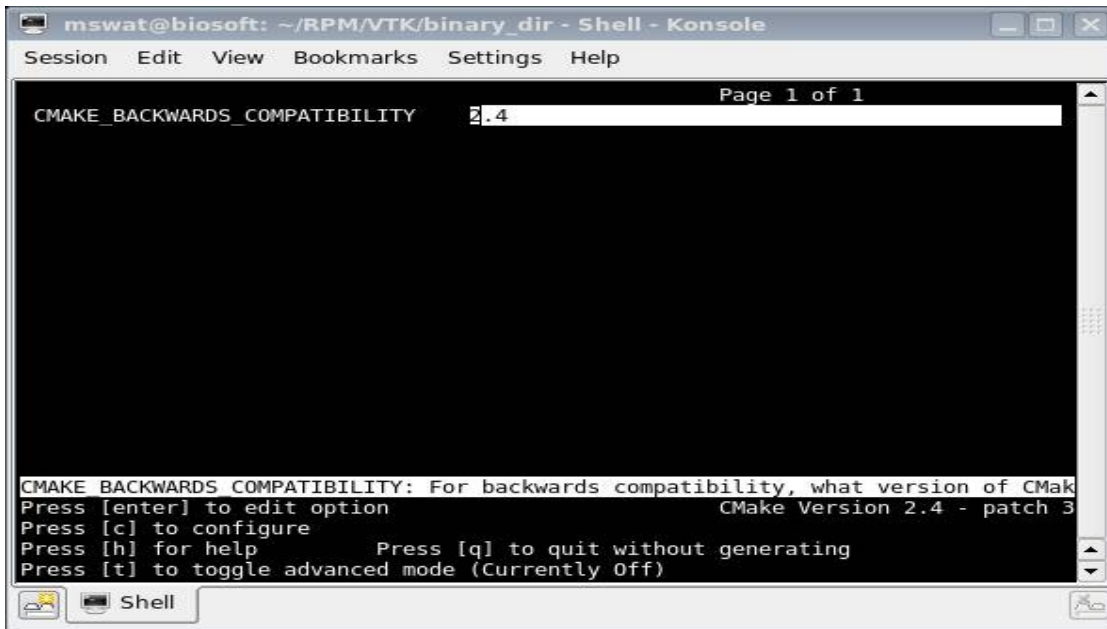
Installing VTK

Finally we will install vtk. You will need to get sources fro VTK 5.0 or higher from www.vtk.org

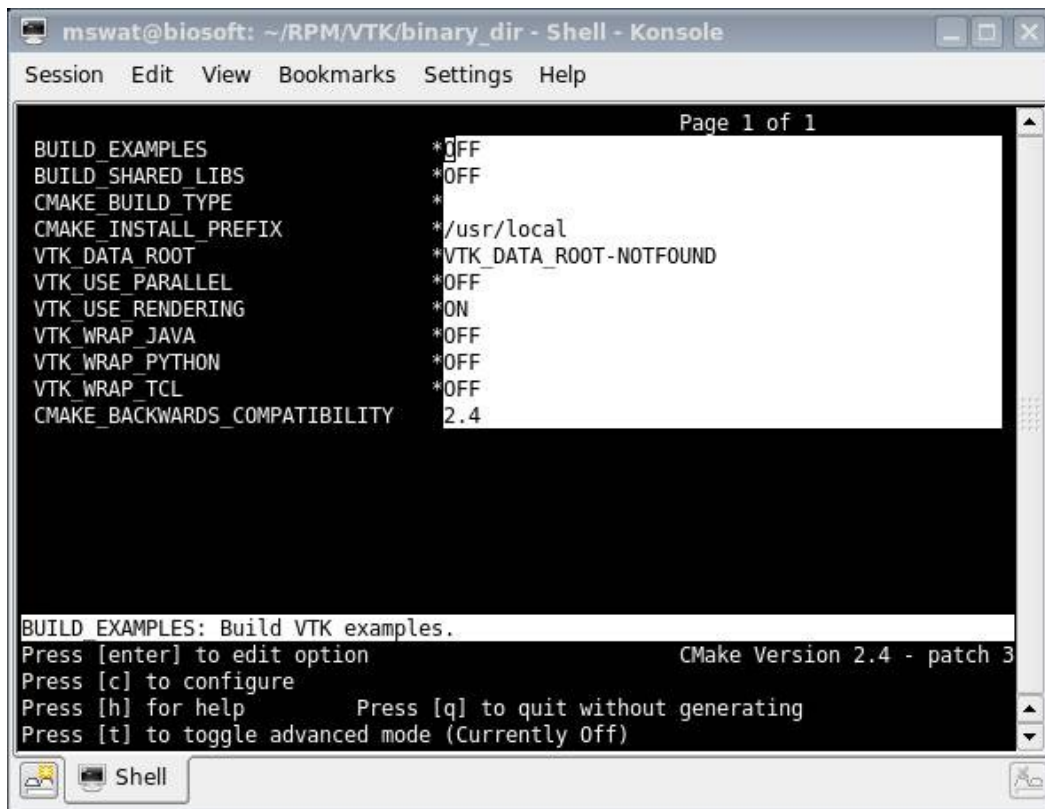
After unpacking the tarball go to the directory which contains the sources and do the following:
mkdir binary_dir
cd binary_dir
ccmake ..

Notice: you need to type ccmake not cmake in the last line

This way you ensure that when you build vtk using cmake the source directory will not be contaminatd with object, executable and auxiliary files. For more information on that please consult cmake manual.
After issuing ccmake .. you will get the following screen:



Hit 'c' and initial configuration will be performed which will give you the next screen:



Change CMAKE_INSTALL_PREFIX to /usr/local/vtk, VTK_PYTHON_WRAP to ON and BUILD_SHARED_LIBS to ON (to change from OFF to on and vice versa, you just hit Enter key). Once you are done with that, hit 'c' to get the following screen (in fact to get generate option display in the ccmake interface, you may need to hit 'c' more than once):

```

mswat@biosoft: ~/RPM/VTK/binary_dir - Shell - Konsole
Session Edit View Bookmarks Settings Help
Page 1 of 2
BUILD_EXAMPLES OFF
BUILD_SHARED_LIBS ON
CMAKE_BACKWARDS_COMPATIBILITY 2.4
CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX /usr/local/vtk
PYTHON_INCLUDE_PATH /usr/include/python2.4
PYTHON_LIBRARY /usr/lib/python2.4/config/libpython2.4.so
TCL_INCLUDE_PATH /usr/include/tcl8.4
TCL_LIBRARY /usr/lib/libtcl8.4.so
TK_INCLUDE_PATH /usr/include/tcl8.4
TK_LIBRARY /usr/lib/libtk8.4.so
VTK_DATA_ROOT VTK_DATA_ROOT-NOTFOUND
VTK_USE_PARALLEL OFF
VTK_USE_RENDERING ON
VTK_USE_RPATH OFF
VTK_WRAP_JAVA OFF
VTK_WRAP_PYTHON ON

BUILD EXAMPLES: Build VTK examples.
Press [enter] to edit option CMake Version 2.4 - patch 3
Press [c] to configure Press [g] to generate and exit
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
Shell

```

Hit 'g' to generate makefiles and you ccmake will exit automatically to binary_dir
 Now type make
 and once compilation is finished type
 sudo make install to install vtk

The only thing remaining is to change .bashrc and .profile by setting VTK_DIR variable

Assuming that you installed vtk to /usr/local/vtk library you would add the following line to .bashrc and .profile

```
export VTK_DIR=/usr/local/vtk/lib/vtk-5.0.
```

The idea here is to set VTK_DIR to the directory containing UseVTK.cmake and VTKConfig.cmake.

swig

Although this package is standard in most distributins we still would recommend that you get latest versin from www.swig.org and install it from source into /usr/local/swig (this path will make the task of finding swig by cmake much easier):

To install swig please follow these steps:

```
tar -zxvf swig-1.3.31.tar.gz
```

```
cd swig-1.3.31
```

```
make
```

make install and make sure you add the following line to the .bashrc and .bash_profile files:

```
export PATH=/usr/local/swig/bin:$PATH
```

CompuCell3D installation

This short instruction will guide you through CompuCell3D installation process on Linux

Notice the actual name of the CompuCell package might be different.

- 1.Download CompuCell installation package to directory called, say CompuCell3D
- 2.Go to CompuCell3D (type : `cd ~/CompuCell3D` notice that in Mac OSX/Unix/Linux “~” denotes your home directory)
- 3.Unpack CompuCell3D*.tar.gz package by typing: `gunzip CompuCell3D.tar.gz`
- 4.Untar (i.e. Extract files from tar archive) by typing: `tar -xvf CompuCell3D.tar`
- 5.Notice that new directory called CompuCell3D was created. Now, go to this directory(`cd CompuCell3D`)
- 6.Before you go any further , please make sure that you have installed required libraries and packages that are needed to compile CompuCell on your Mac box.
- 7 type `./install_full -i <installation_directory_path>`

REMARK:

On Linux systems CompuCellPlayer may slow down the simulation. To avoid too much of a slowdown decrease screen refresh rate in CompuCell Player:

Go to Configure->Screenshot frequency and increase Screen Update Frequency
We are working to resolve this problem in more elegant way

IMPORTANT: The automated installer will work with most “standard” installation of the above-mentioned packages. IF your systems has packages installed in non-standard places you might actually have to do a little bit more work and use cmake front end to set few paths there in order to install CompuCell3D. See below, for instructions how to do this.

To install CompuCell3D using cmake front-end - ccmake first run `./install_full -i <installation_directory_path>` script. The installation will be interrupted and an error message will be displayed. Let us guide you through a concrete installation process of CompuCell3D on one of the Indiana University clusters. The packages on this cluster are installed in custom directories so that cmake could not locate several of them. OK let's start.

We typed:

```
./install_full -i ~/CompuCell3D_Qt4_3.1.0_install
```

and got the following errors:

```
-- Looking for connect
-- Looking for connect - found
-- Looking for remove
-- Looking for remove - found
-- Looking for shmat
-- Looking for shmat - found
-- Looking for IceConnectionNumber in ICE
-- Looking for IceConnectionNumber in ICE - found
-- Looking for Q_WS_X11
-- Looking for Q_WS_X11 - found
-- Looking for Q_WS_MAC
-- Looking for Q_WS_MAC - not found.
-- Looking for Q_WS_WIN
-- Looking for Q_WS_WIN - not found.
-- Found Qt-Version 4.3.0
THIS IS VTK_INCLUDE_DIRS /N/soft/linux-sles9-ppc64/VTK-5.0.2-32/include/vtk-
5.0;/N/soft/linux-sles9-ppc64/mesa-
6.4.2/include;/usr/include;/usr/include;/usr/include;/usr/include;/usr/include;/usr/
include;/usr/include;/usr/include;/usr/include;/N/soft/linux-sles9-ppc64/mesa-
6.4.2/include;/opt/osshpc/mpich-mx/32/include;/N/soft/linux-sles9-ppc64/python-
2.4.3-32/include/python2.4
```

```

THIS IS VTK_LIBRARY_DIRS /N/soft/linux-sles9-ppc64/VTK-5.0.2-32/lib
DESIRED_QT_VERSION=4
-- Will not build QVTK plugin because VTK isn't build as shared libraries
THIS IS EXTRALIB FOR LINUX util
CMake Error: Swig was not found on the system. Please specify the location of Swig.
CMake Error: This project requires some variables to be set,
and cmake can not find them.
Please set the following variables:
XERCESC_INCLUDE_DIR
XERCESC_LIBRARY

-- Configuring done
ERRORS DURING CONFIGURATION

```

Clearly swig and xercesc were not found. This is in part due to the way the macros that search for the two tools are written. To make long story short those macros are somewhat simplistic and will be upgraded in the future. In any case, let's see how we deal with this situation.

First thing that you need to do is to go to
<installation_directory_path>/cmake_binary_dir.

In our case I need to type

```
cd ~/CompuCell3D_Qt4_3.1.0_install/ cmake_binary_dir
```

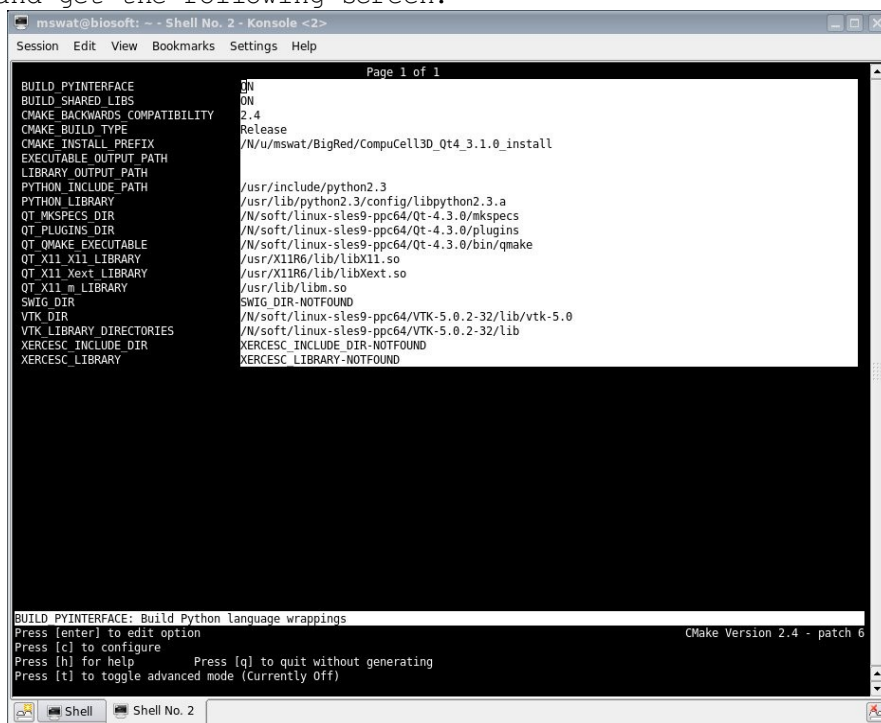
The `cmake_binary_dir` is a directory where install script and cmake store all the makefiles and from this directory we will be compiling CompuCell3D.

While in `cmake_binary_dir` open up cmake front-end. The syntax is `ccmake <source_directory>`. Now, `<source_directory>` in our case is the directory that contains, source tree, cmake files and install_full script.

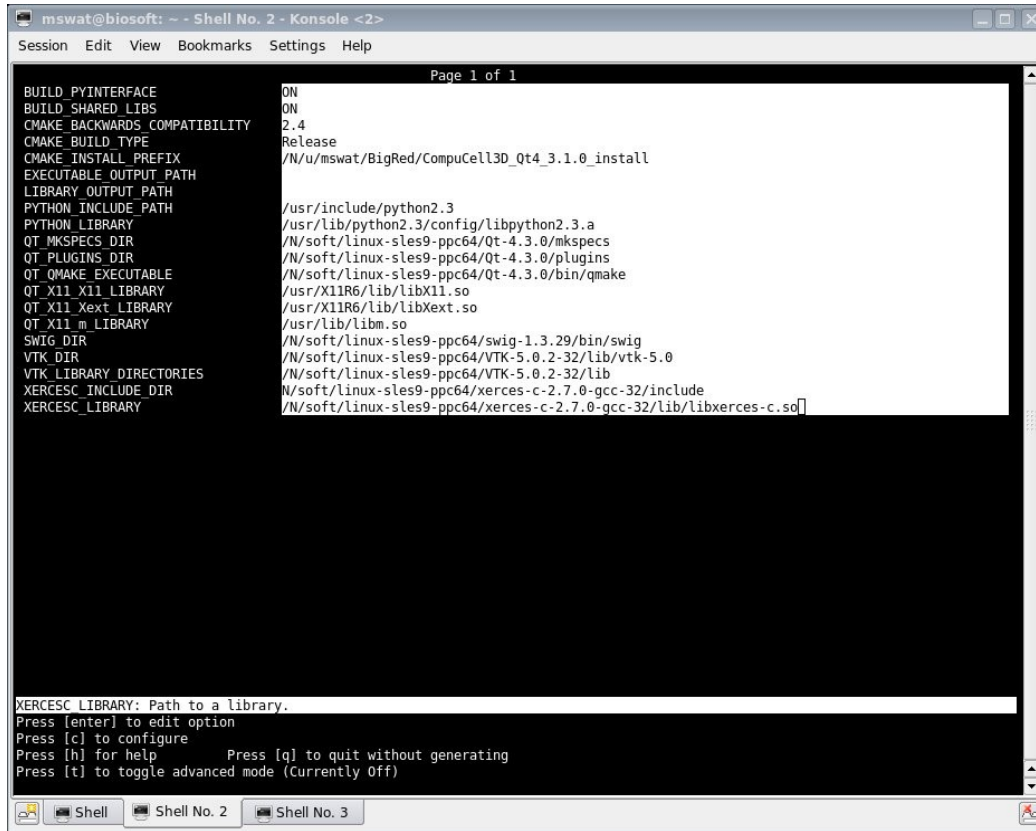
So we type:

```
ccmake ~/CompuCell3D_Qt4_3.1.0
```

and get the following screen:



As you can see from the window above cmake had difficulties finding SWIG_DIR , XERCESC_INCLUDE_DIR, XERCESC_LIBRARY. All we need to do now is to put in those values:



```
mswat@biosoft: ~ - Shell No. 2 - Konsole <2>
Session Edit View Bookmarks Settings Help
Page 1 of 1
BUILD_PYINTERFACE ON
BUILD_SHARED_LIBS ON
CMAKE_BACKWARDS_COMPATIBILITY 2.4
CMAKE_BUILD_TYPE Release
CMAKE_INSTALL_PREFIX /N/u/mswat/BigRed/CompuCell3D_Qt4_3.1.0_install
EXECUTABLE_OUTPUT_PATH
LIBRARY_OUTPUT_PATH
PYTHON_INCLUDE_PATH /usr/include/python2.3
PYTHON_LIBRARY /usr/lib/python2.3/config/libpython2.3.a
QT_MKSPECS_DIR /N/soft/linux-sles9-ppc64/Qt-4.3.0/mkspecs
QT_PLUGINS_DIR /N/soft/linux-sles9-ppc64/Qt-4.3.0/plugins
QT_QMAKE_EXECUTABLE /N/soft/linux-sles9-ppc64/Qt-4.3.0/bin/qmake
QT_X11_X11_LIBRARY /usr/X11R6/lib/libX11.so
QT_X11_Xext_LIBRARY /usr/X11R6/lib/libXext.so
QT_X11_m_LIBRARY /usr/lib/libm.so
SWIG_DIR /N/soft/linux-sles9-ppc64/swig-1.3.29/bin/swig
VTK_DIR /N/soft/linux-sles9-ppc64/VTK-5.0.2-32/lib/vtk-5.0
VTK_LIBRARY_DIRECTORIES /N/soft/linux-sles9-ppc64/VTK-5.0.2-32/lib
XERCESC_INCLUDE_DIR /N/soft/linux-sles9-ppc64/xerces-c-2.7.0-gcc-32/include
XERCESC_LIBRARY /N/soft/linux-sles9-ppc64/xerces-c-2.7.0-gcc-32/lib/libxerces-c.so

XERCESC_LIBRARY: Path to a library.
Press [enter] to edit option
Press [c] to configure
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

In your case the directories that we input above will be different. Once you filled up all the missing entries on the previous screen have been input we need to configure cmake build system of CompuCell3D and generate makefiles. To configure , simply hit "c" - see the bottom of the above screen for additional key assignments. Occasionally you may have to hit "c" several times until [g] option appears at the bottom of the screen:

```
mswat@biosoft: ~ - Shell No. 2 - Konsole <2>
Session Edit View Bookmarks Settings Help
Page 1 of 1
BUILD_PYINTERFACE ON
BUILD_SHARED_LIBS ON
CMAKE_BACKWARDS_COMPATIBILITY 2.4
CMAKE_BUILD_TYPE Release
CMAKE_INSTALL_PREFIX /N/u/mswat/BigRed/CompuCell3D_Qt4_3.1.0_install
EXECUTABLE_OUTPUT_PATH
LIBRARY_OUTPUT_PATH
PYTHON_INCLUDE_PATH /usr/include/python2.3
PYTHON_LIBRARY /usr/lib/python2.3/config/libpython2.3.a
QT_MKSPECS_DIR /N/soft/linux-sles9-ppc64/Qt-4.3.0/mkspecs
QT_PLUGINS_DIR /N/soft/linux-sles9-ppc64/Qt-4.3.0/plugins
QT_QMAKE_EXECUTABLE /N/soft/linux-sles9-ppc64/Qt-4.3.0/bin/qmake
QT_X11_X11_LIBRARY /usr/X11R6/lib/libX11.so
QT_X11_Xext_LIBRARY /usr/X11R6/lib/libXext.so
QT_X11_m_LIBRARY /usr/lib/libm.so
SWIG_DIR /N/soft/linux-sles9-ppc64/swig-1.3.29/bin/swig
SWIG_EXECUTABLE /N/soft/linux-sles9-ppc64/swig-1.3.29/bin/swig
VTK_DIR /N/soft/linux-sles9-ppc64/VTK-5.0.2-32/lib/vtk-5.0
VTK_LIBRARY_DIRECTORIES /N/soft/linux-sles9-ppc64/VTK-5.0.2-32/lib
XERCESC_INCLUDE_DIR /N/soft/linux-sles9-ppc64/xerces-c-2.7.0-gcc-32/include
XERCESC_LIBRARY /N/soft/linux-sles9-ppc64/xerces-c-2.7.0-gcc-32/lib/libxerces-c.so

BUILD_PYINTERFACE: Build Python language wrappings
Press [enter] to edit option
Press [c] to configure Press [g] to generate and exit
Press [h] for help Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
CMake Version 2.4 - patch 6
```

Now, all we need to do is to hit “g” to generate makefiles. After this step the cmake front-end will automatically shut down and take you to a place from which you would type “make”

```
mswat@biosoft: ~ - Shell No. 2 - Konsole <2>
Session Edit View Bookmarks Settings Help
mswat@BigRed:~> cd CompuCell3D_Qt4_3.1.0/
mswat@BigRed:~/CompuCell3D_Qt4_3.1.0> ./install full -i ~/CompuCell3D_Qt4_3.1.0_
install_path: /N/u/mswat/BigRed/CompuCell3D_Qt4_3.1.0_install
Successfully created installation directory: /N/u/mswat/BigRed/CompuCell3D_Qt4_3
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Check size of void* - done
-- Check size of void** - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Found ZLIB: /usr/lib/libz.so
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- Looking for regex.h
-- Looking for regex.h - found
-- Looking for unistd.h
-- Looking for unistd.h - found
-- Looking for XOpenDisplay in /usr/X11R6/lib/libX11.so;/usr/X11R6/lib/libXext.s
-- Looking for XOpenDisplay in /usr/X11R6/lib/libX11.so;/usr/X11R6/lib/libXext.s
-- Looking for gethostbyname

mswat@BigRed:~/CompuCell3D_Qt4_3.1.0_install/cmake_binary_dir> cmake ~/CompuCell3D_Qt4_3.1.0

mswat@BigRed:~/CompuCell3D_Qt4_3.1.0_install/cmake_binary_dir> make[]
```


Now the actual compilation has began and after everything compiles all you would type to install CompuCell3D would be:

```
make install
```

As you have seen installing CompuCell3D using cmake front end is a bit more complex than typing `./install_full -i <installation_path>` but on the other hand not that complicated. In any case , on some machines this is the only way to install CompuCell3D so it is worth learning it. In general, as cmake becomes more and more popular it is useful to get familiar with this way of compiling software.