GATE Installation Instructions

Introduction

You can view the official Gate installation documentation here.
https://opengate.readthedocs.io/en/latest/
It is recommended to use the Ubuntu system for configuration (please do not use the Deepin system, as the author encountered many unresolved issues while configuring with Deepin).
First, you need to install the relevant dependencies.

```
 sudo apt install git dpkg-dev cmake g++ gcc binutils libx11-dev libxpm-dev
libxft-dev libxext-dev gfortran libssl-dev libpcre3-dev libglu1-mesa-dev libglew-
dev libftgl-dev libfftw3-dev libcfitsio-dev libgraphviz-dev libavahi-compat-
libdnssd-dev libldap2-dev python2-dev libxml2-dev libkrb5-dev libgsl-dev qtbase5-
dev qtchooser qt5-qmake qtbase5-dev-tools qt5* libxtst-dev libxrender-dev libxmu-
dev libxmuu-dev libmysqlclient-dev xlibmesa-glu-dev libglew1.5-dev graphviz-dev
libgsl0-dev
```

STEP 1 Download Gate

1. Access the GATE official website:
   http://www.opengatecollaboration.org/

2. Then you can see the latest GATE version and its configuration requirements.

GATE source code

...

GATE is distributed under the terms of the GNU Lesser General Public Licence (LGPL), version 3, 29 June 2007. See LICENSE.md for further details.

SOURCE CODE, EXAMPLES AND TOOLS ARE AVAILABLE ON GITHUB: HTTPS://GITHUB.COM/OPENGATE

Below, you can download the major releases and get informed about the dependencies and major features.

GATE 9.3

RELEASE DATE: 24/05/2023

Required dependencies:

- geant4 11.1 (c++17)
- root ROOT 6.24.06 is recommended (with -DCMAKE_CXX_STANDARD=17)
- gcc up to 11.3
- cmake minimal version 3.3 (with SSL support)
- ITK 5.2.0 (with RTK enabled)

Optional dependencies:

- libtorch 1.10.1

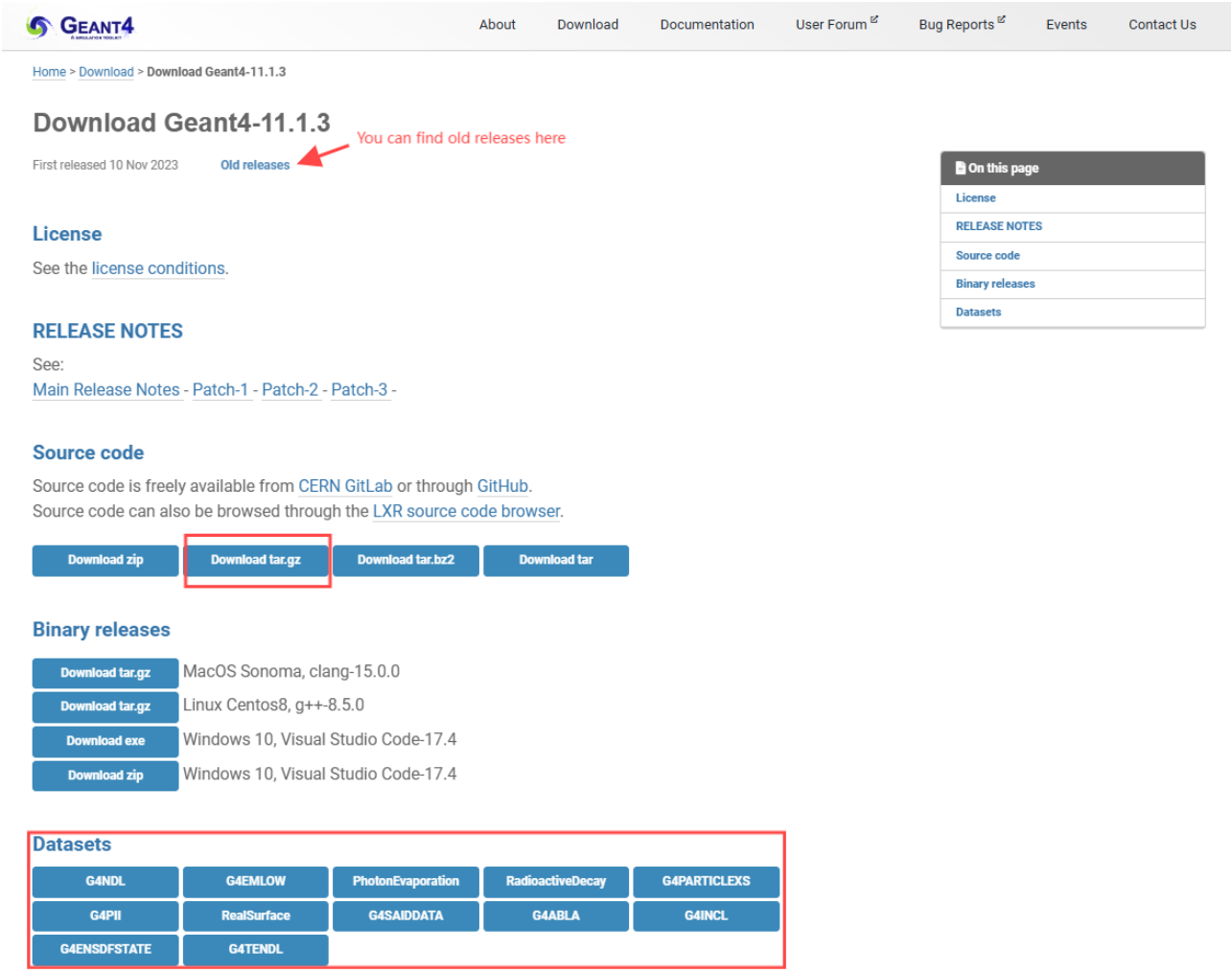
NEW FEATURES ARE DESCRIBED HERE.

Keep in mind the `required dependencies` for the GATE version you choose, and install them strictly according to the requirements; otherwise, errors will occur.

STEP 2 Install geant

1. Download Geant and the data files.
   Download link: https://geant4.web.cern.ch/support/download
   When you click the link, you will be automatically directed to the download page for the latest version.

Note: The latest version may not be the one required for the GATE you are downloading!

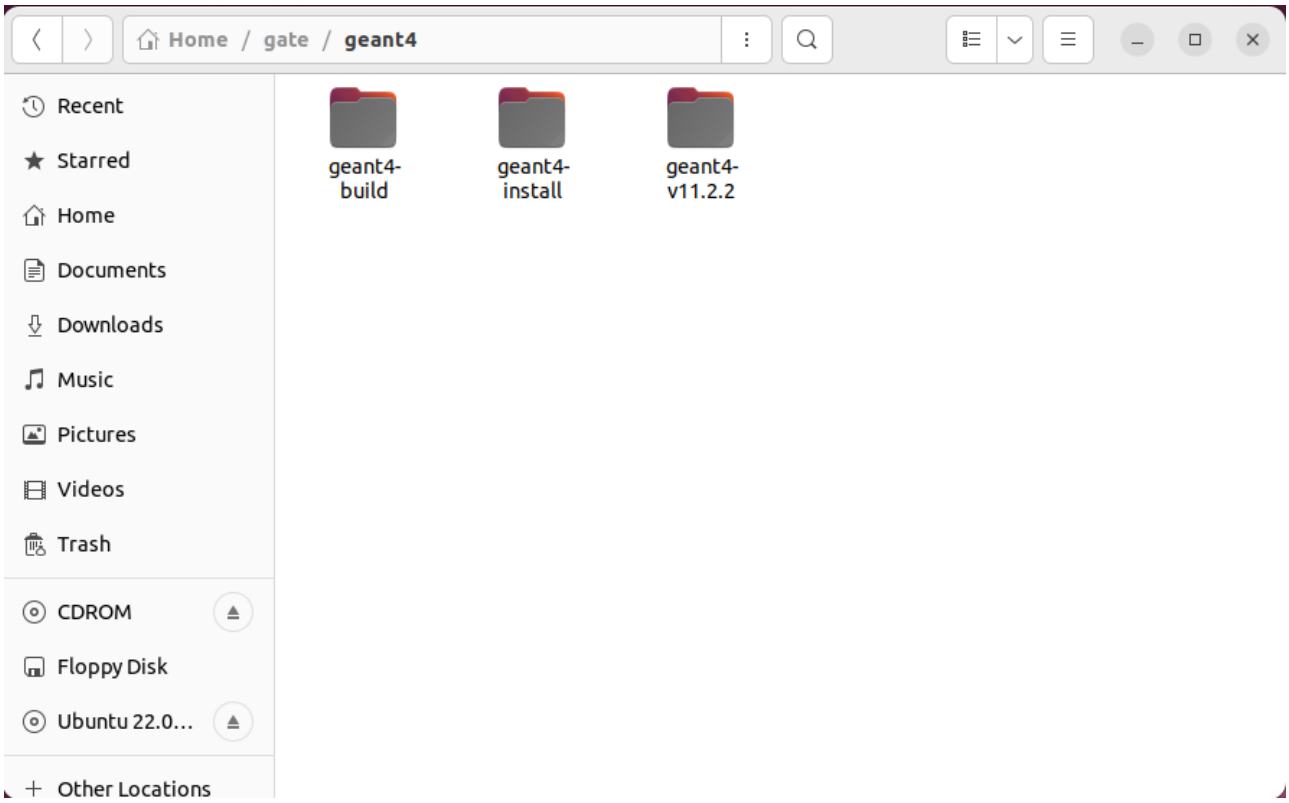


Find the version you need, then download the `source code` and `datasets`.

2. Choose a location with ample space (approximately 15GB will be needed after installation), create a `gate` folder, and create a `geant4` folder within it.

```
mkdir gate
cd gate
mkdir geant4
```

Unzip the downloaded source code in the `geant4` folder. Then create two folders: `geant4-build` and `geant4-install`.



3. Use CMake for compilation, and remember to use `sudo` to grant permissions during the process.

```
cd geant4
mkdir geant4-build
cd geant4-build
```

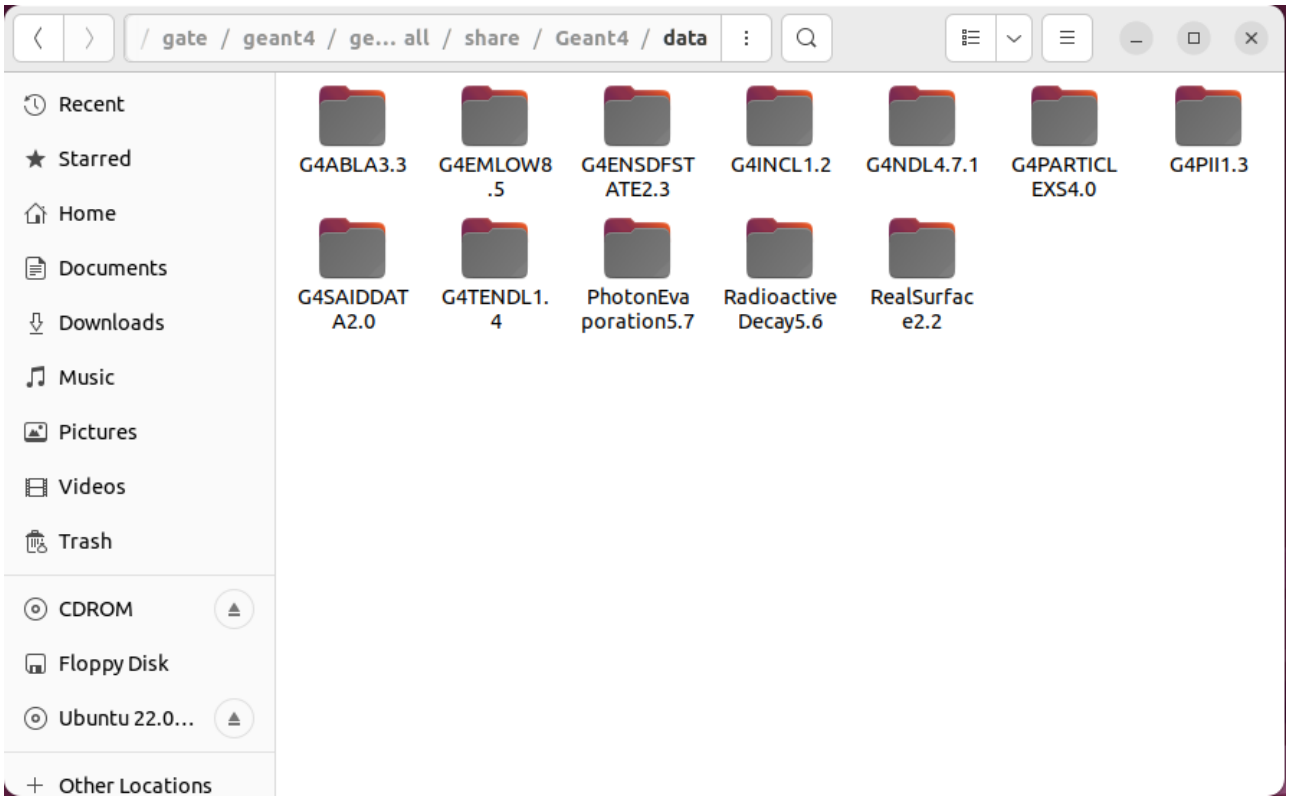
Make sure to replace the installation path.

```
cmake -DCMAKE_INSTALL_PREFIX=/PATH_TO_YOUR_GEANT/geant4-install/ -
DGEANT4_USE_OPENGL_X11=ON -DGEANT4_USE_RAYTRACER_X11=ON -DGEANT4_USE_QT=ON -
DGEANT4_USE_SYSTEM_EXPAT=OFF GEANT4_BUILD_MULTITHREADED=ON
/PATH_TO_YOUR_GEANT/geant4-v11.1.3
```

```
make -j4
make install
```

The command `make -j4` means to use four processors for compilation. If your computer's performance is good enough, you can choose `make -j8` or higher to improve compilation speed.

4. Unzip the datasets to `/PATH_TO_YOUR_GEANT/geant4-install/share/Geant4/data`. (The `data` folder is a new folder that you need to create.)



Note that the extracted datasets might be in a deeper directory, so it's better to place them in a shallower directory.

5. After adding the data, continue by adding environment variables directly in the `.bashrc` file.

```
cd ~/
```

Make sure to replace the installation path.

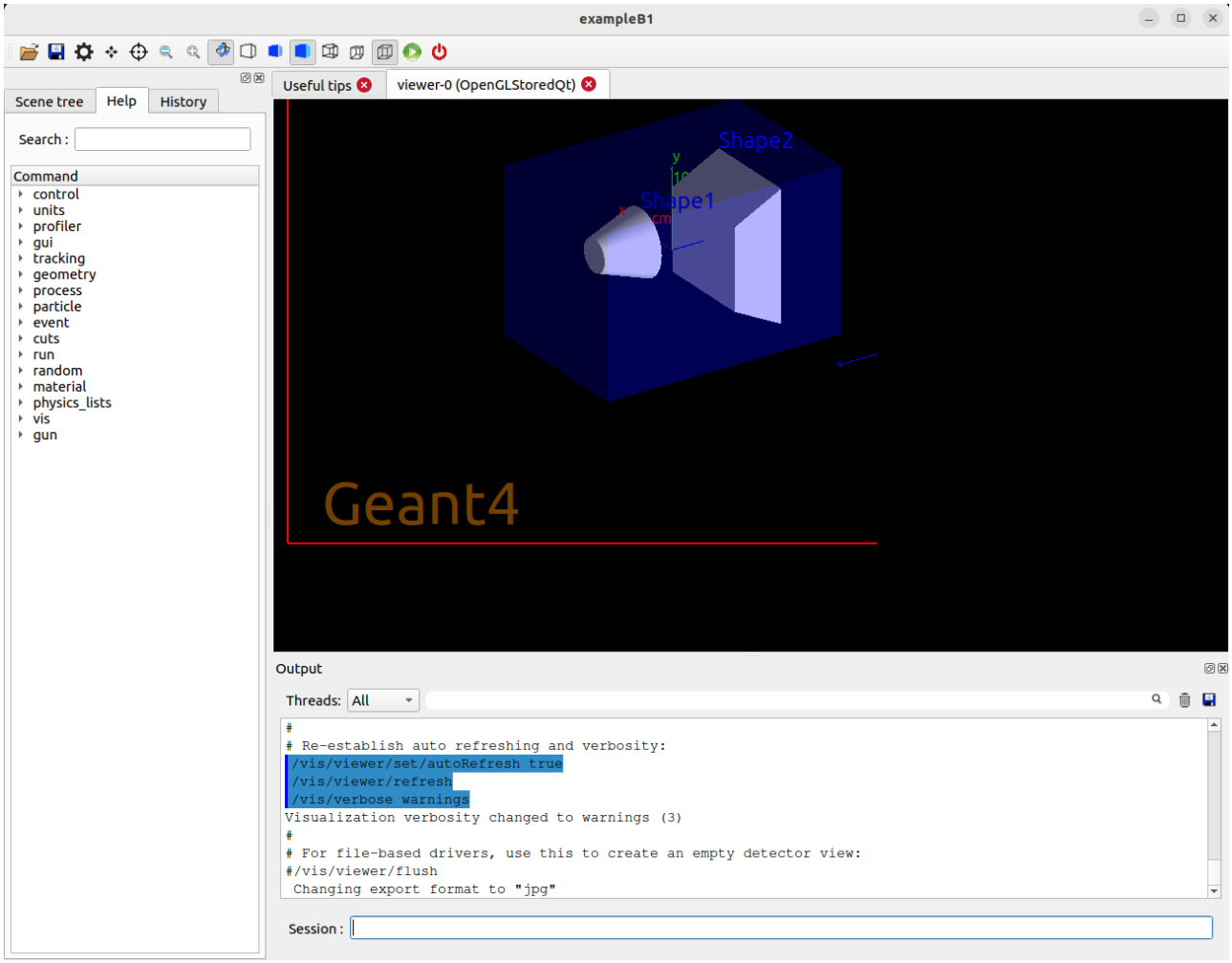
```
echo 'source /PATH_TO_YOUR_GEANT/geant4-
install/share/Geant4/geant4make/geant4make.sh' >> ~/.bashrc
```

```
echo 'source /PATH_TO_YOUR_GEANT/geant4-install/bin/geant4.sh' >> ~/.bashrc
```

6. Test the sample files.
   Enter the sample folder: `/PATH_TO_YOUR_GEANT/geant4-v11.1.3/examples/basic/`

```
mkdir B1-build
cd B1-build
cmake ../B1
make -j4
./exampleB1
```

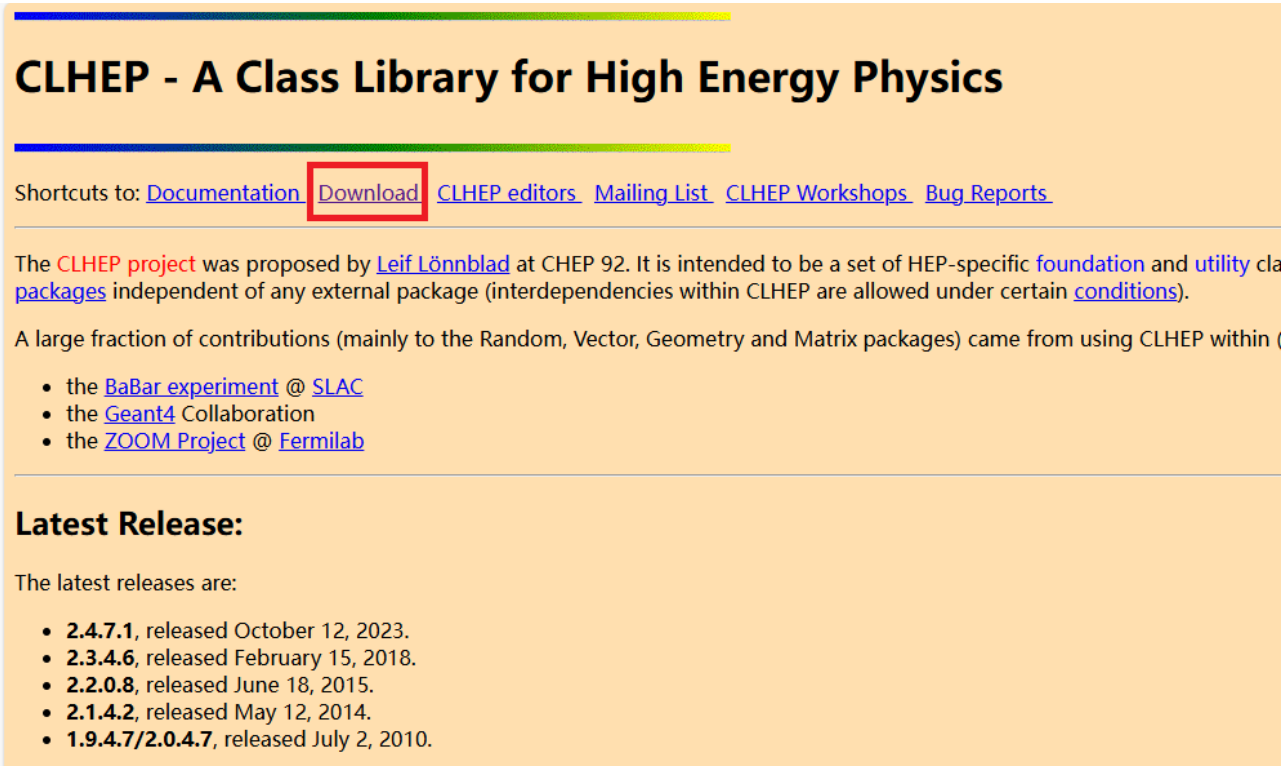
If you see the interface shown in the image, it means you have succeeded.

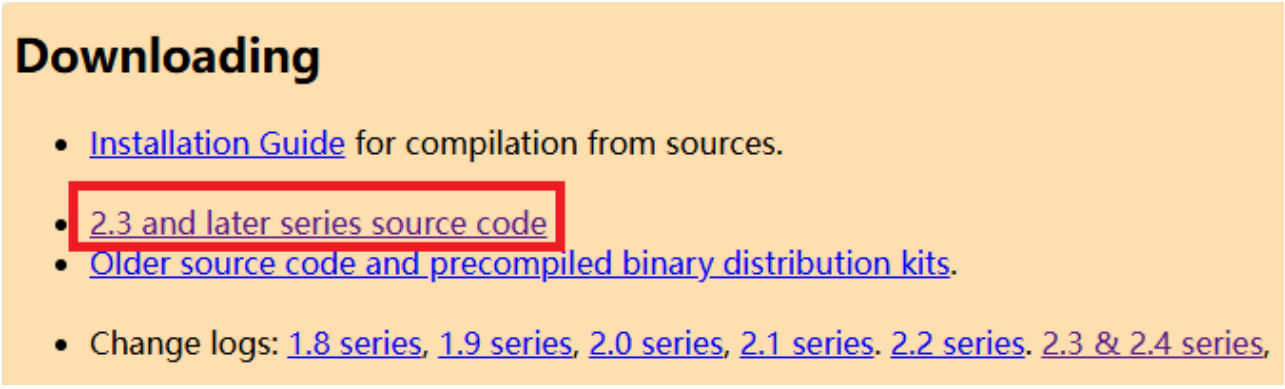


STEP 3 Install clhep

1. Download the lateset version.

   Download link: http://proj-clhep.web.cern.ch/proj-clhep/





Downloading

- Installation Guide for compilation from sources.
- 2.3 and later series source code
- Older source code and precompiled binary distribution kits.
- Change logs: 1.8 series, 1.9 series, 2.0 series, 2.1 series. 2.2 series. 2.3 & 2.4 series,

CLHEP 2.3 and later

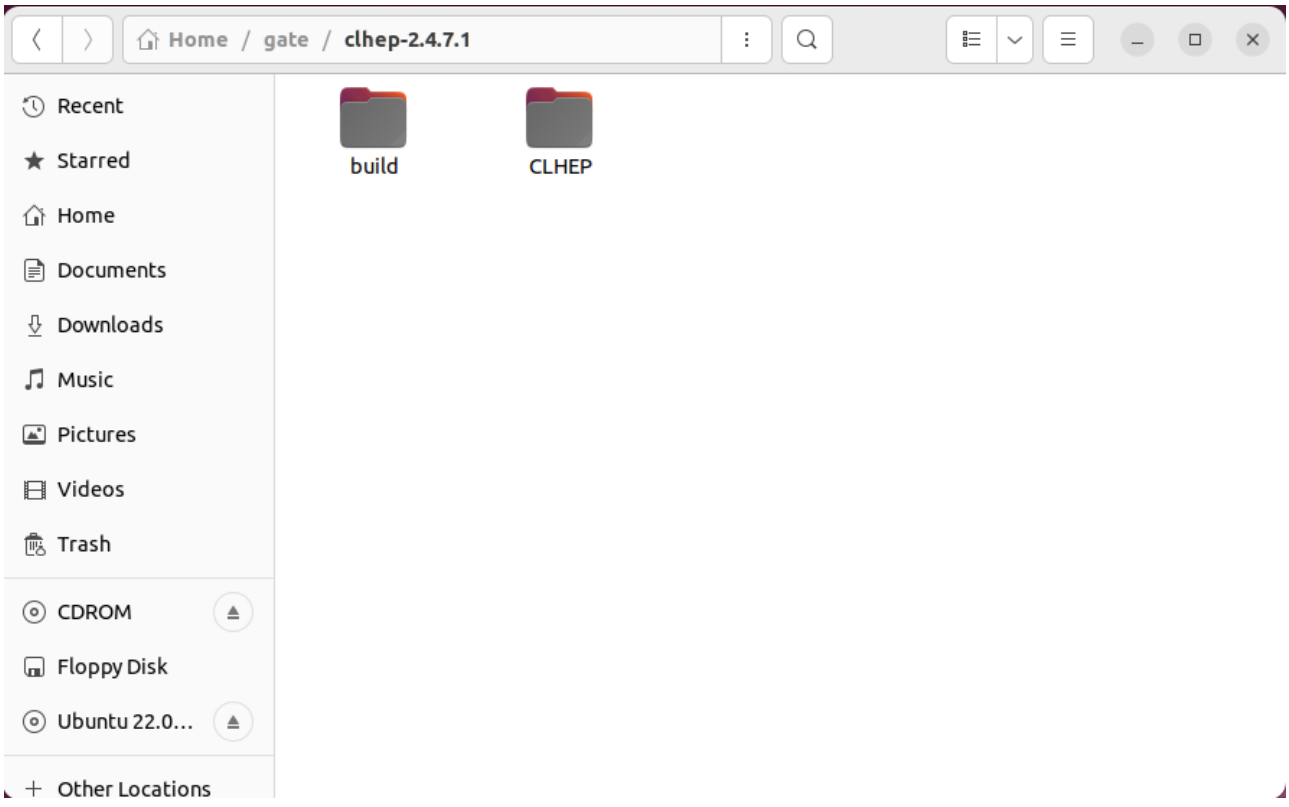
Shortcuts to: Documentation Download CLHEP editors Mailing List CLHEP Workshops Bug Reports

As of CLHEP 2.3.1.0, a modern compiler which supports at least -std=c++11 is REQUIRED.

Note that when using gcc, gcc 4.8 or better is required. When using clang, clang 3.5 or better is required. icc needs to be version 15 or better.

Release	Source	ChangeLog	Notes
2.4.7.1	clhep-2.4.7.1.tgz	ChangeLog for 2.4.7.1	add missing shootArray implementations and fix install
2.4.7.0	clhep-2.4.7.0.tgz	ChangeLog for 2.4.7.0	MIXMAX update for performance
2.4.6.4	clhep-2.4.6.4.tgz	ChangeLog for 2.4.6.4	Add support for -std=c++20.
2.4.6.3	clhep-2.4.6.3.tgz	ChangeLog for 2.4.6.3	Fix XCode 14.1 compilation warnings.
2.4.6.2	clhep-2.4.6.2.tgz	ChangeLog for 2.4.6.2	Update Evaluator units.
2.4.6.0	clhep-2.4.6.0.tgz	ChangeLog for 2.4.6.0	Resolve worrying gcc 12 warnings.
2.4.5.4	clhep-2.4.5.4.tgz	ChangeLog for 2.4.5.4	RandGamma.cc: reproduce results reliably in multi-threading environment
2.4.5.3	clhep-2.4.5.3.tgz	ChangeLog for 2.4.5.3	fix problems with out of source comparison
2.4.5.2	clhep-2.4.5.2.tgz	ChangeLog for 2.4.5.2	cmake improvements and fix compilation warnings
2.4.5.1	clhep-2.4.5.1.tgz	ChangeLog for 2.4.5.1	fix compilation warnings

2. Create a `clhep` folder under the `gate` folder. Then unzip the source code here and create a `build` folder.



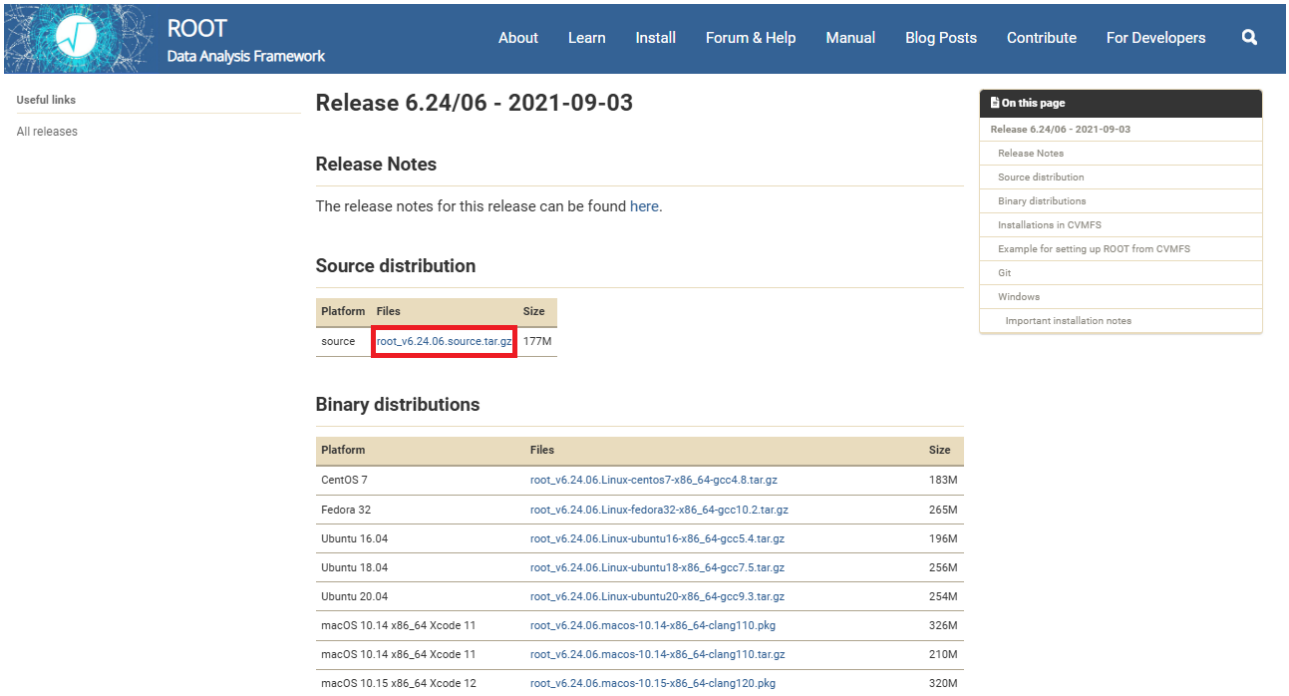
3. Next is the compilation.

```
cd build
cmake ../CLHEP
make
sudo make all install
```

The file paths may differ from yours, so be sure to replace them accordingly.

STEP 4 Install ROOT

1. Download the ROOT installation package.
   Download link: https://root.cern/install/all_releases/
   Choose the version of Gate that is recommended for you.



2. Create a `root` folder under the `gate` folder. Then unzip the source code here and create a `root-install` folder.

```
cmake -DCMAKE_INSTALL_PREFIX=/PATH_TO_YOUR_ROOT/root-install -
DGEANT4_USE_OPENGL_X11=ON -DGEANT4_BUILD_MULTITHREADED=ON
/PATH_TO_YOUR_ROOT/root-6.24.06
```

```
make  -j4
sudo make all install
```

3. Adding environment variables directly in the `.bashrc` file.

```
cd ~/
```

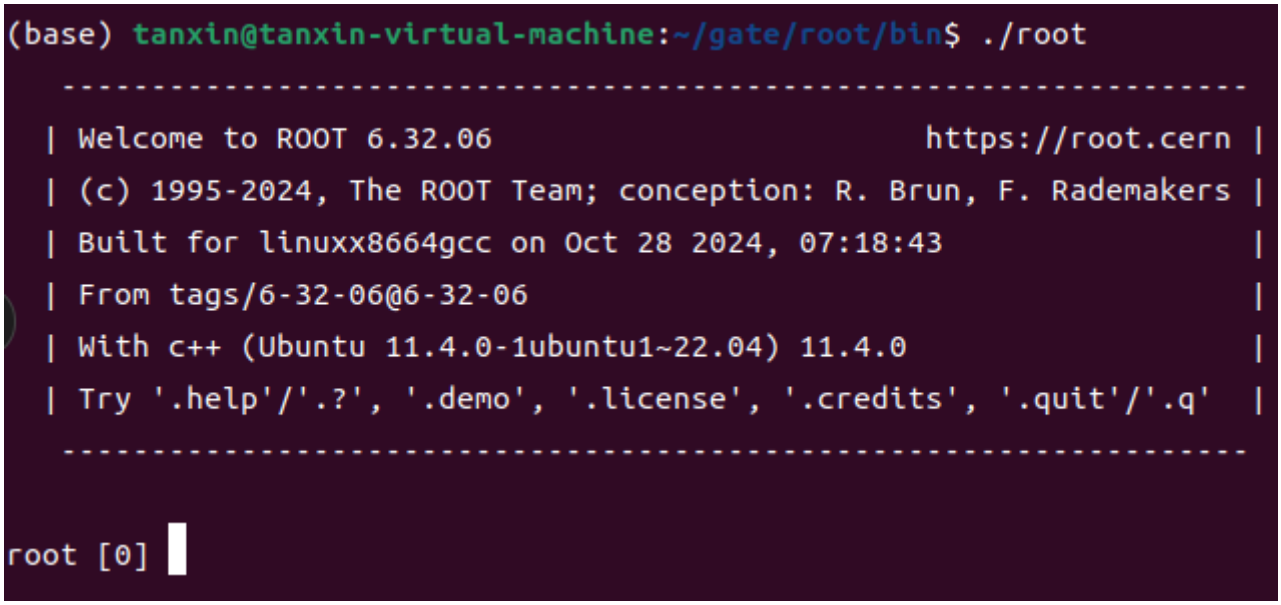
Make sure to replace the installation path.

```
echo 'source /PATH_TO_YOUR_ROOT/bin/thisroot.sh' >> ~/.bashrc
```

4. Test the installation.

```
cd /PATH_TO_YOUR_ROOT/bin
./root
```

If you see the following interface, it means the installation was successful.



STEP 5 Install OpenGL, ITK, and RTK

1. Install OpenGL.

```
sudo apt-get install build-essential
sudo apt-get install libgl1-mesa-dev
sudo apt-get install freeglut3-dev
```

2. Install ITK.
Download link: https://github.com/InsightSoftwareConsortium/ITK
Create a `ITK-Sandbox` folder under the `gate` folder. Then unzip the source code here and create a `ITK-build` folder.

```
cd ITK-build
cmake ../ITK
make -j4
sudo make all install
```

3. Install RTK.
   Download link: https://www.openrtk.org/
   Create a `RTK-Sandbox` folder under the `gate` folder. Then unzip the source code here and create a `RTK-build` folder.

```
cd RTK-build
cmake ../RTK
make -j4
```

   If you encounter an error like this.

```
-- Looking for isnan
-- Looking for isnan - found
CMake Error at CMakeLists.txt:239 (message):
  Modules can only be built against an ITK build tree; they cannot be built
  against an ITK install tree.

-- Configuring incomplete, errors occurred!
See also "/data/home/magician/gate/RTK-Sandbox/RTK-build/CMakeFiles/CMakeOutput.log".
```

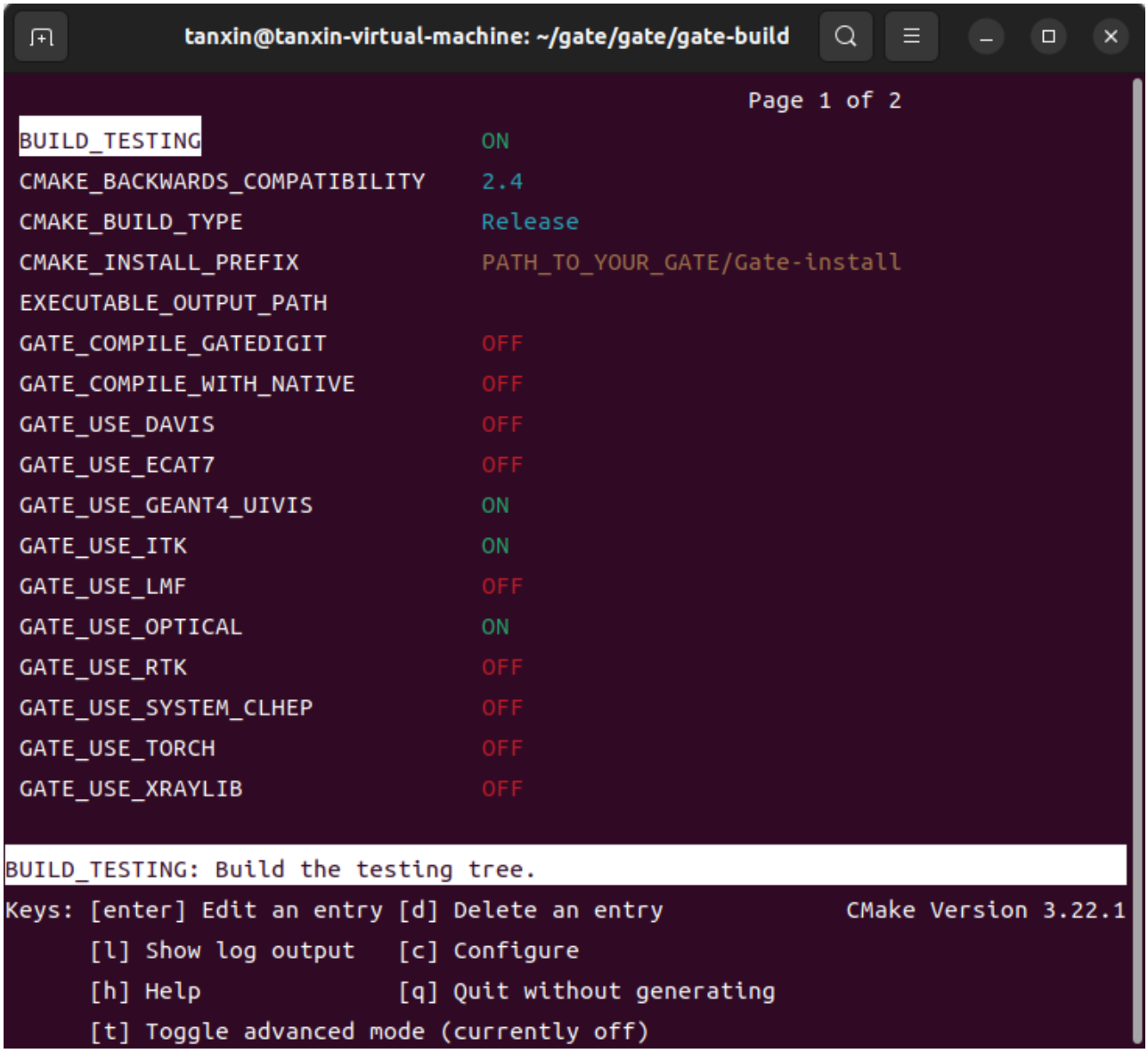
   According to the ITK GitHub issue, RTK has moved to Modules/External, and you need to activate Module_RTK.

STEP 6 Install Gate

1. Create a `Gate` folder under the `gate` folder.
   Unzip the downloaded source code in the `Gate` folder. Then create two folders: `Gate-build` and `Gate-install`.

```
cd Gate-build
cmake ../Gate-9.3
ccmake ../Gate-9.3
```

Open `ccmake` and configure the environment as shown in the image.



Then press the `c`, `e`, and `q` keys in sequence to generate the binary executable files.

2. Start the compilation.

```
make -j4
sudo make install -j4
```

If you encounter the error `GATE should be compiled with a non-multithreaded installation of Geant4`, simply comment out the corresponding line numbers in the CMake file, and it will be fine; you can also use multithreading.

3. Configure the environment variables by creating an `auto.bashrc` text file in the `Gate` folder.

```
source /PATH_TO_YOUR_ROOT/root/bin/thisroot.sh
source /PATH_TO_YOUR_GEANT/geant4-install/bin/geant4.sh
export PATH=$PATH:/PATH_TO_YOUR_GATE/Gate-install
export PATH=$PATH:/PATH_TO_YOUR_ITK/ITK-build/bin
export LD_LIBRARY_PATH=/PATH_TO_YOUR_ITK/ITK-build/bin:$LD_LIBRARY_PAT
```

Put all the necessary environment variables into that file.

4. Open the `Gate-install` folder we created earlier and run the `Gate` binary executable in the terminal. At this point, the Gate environment has been successfully configured.