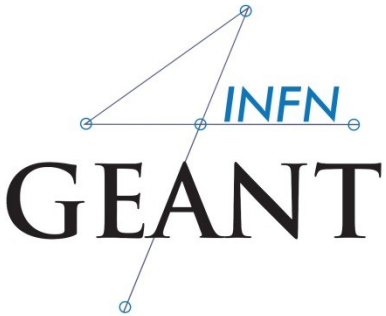


Installation GEANT4

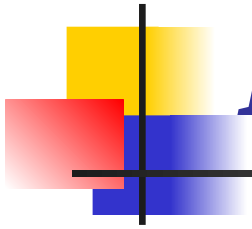
Alexei Sytov

INFN – Ferrara Division

sytov@fe.infn.it

Material by P. Cirrone, J. Pipek

XXII Seminar on Software for Nuclear, Subnuclear and Applied Physics
10th June 2025



Installation process

1. Check that you meet all the requirements
2. Download **Geant4** source code
3. **Configure** the build using **CMake**
4. **Make** & install
5. Configure your environment to use Geant4

https://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/index.html

1) Supported platforms and requirements

Virtual Machine:
CentOS 7 with gcc 10.2.1

- ***Operating system***
  - *"recent" Linux (e.g. CentOS 7), best support*
  - *macOS 10.10+*
  - *Windows 10 (limited support, not recommended)*
- ***Compilers***
  - *C++11 compliance*
  - *such as GCC 8+, clang 8+, Visual C++ 2019+*
- ***CMake** (configuration generation tool) 3.16+*
- *System libraries (as development packages):*
  - *expat, xerces-c*

These may or may not be necessary. Just keep this in mind when compilation fails.

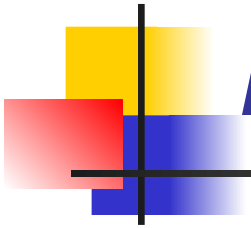
Pre-requirement: CMake installation

*The VM has **CMake** installed*

- *Geant4 build is configured by **CMake** (version >3.16)*
- *Depending on the OS installation, **CMake** may not be installed by default. In that case you have to install it:*
  - ***Linux**: it is recommended to use the CMake provided by the package management system of your distribution.*
    *If version **3.16+** is not available:*
    1. *download the latest version (`http://www.cmake.org/`)*
    2. *unzip the tar-ball*
    3. ***./bootstrap**, **make**, **make install***

  Ubuntu: sudo apt install cmake
  - ***macOS**: install it using the Darwin64 dmg installerpackage*
  - ***Windows**: install it using the Win64/32 exe installerpackage*

Pre-requirements: optional libraries

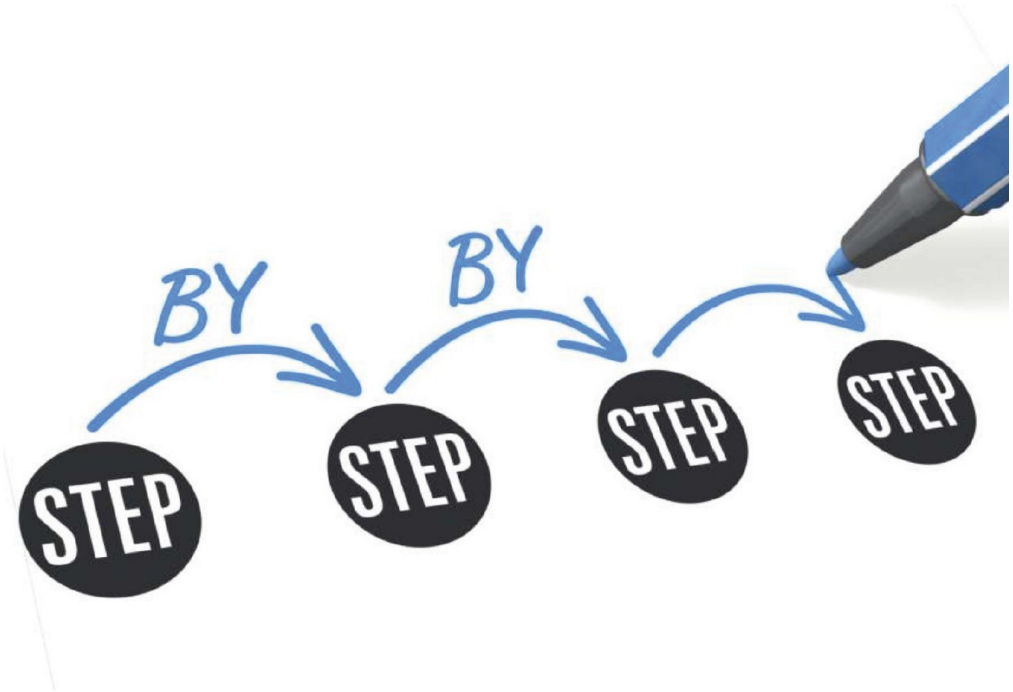


- ***X11*** *for simple graphical user interface and ray-tracing*
- ***OpenGL*** *for visualization*
- ***Qt4*** *or* ***Qt5*** *for graphical user interface*
- ***ROOT*** *for data analysis* *(even inside Geant4)*

Less frequently used libraries/tools:

Motif, OpenInventor, DAWN, RayTracer X11, HepRApp, WIRED JAS Plug-in, AIDA, VRML browser, (external) CLHEP, Wt...

Installation steps

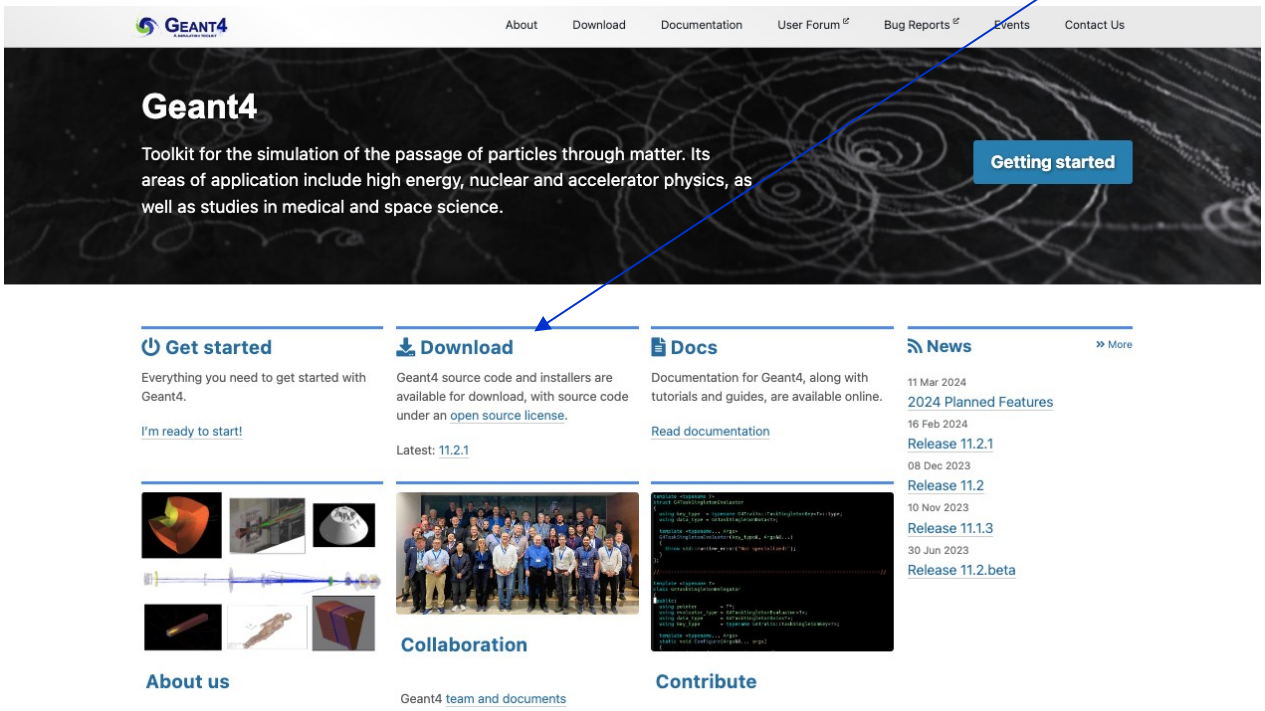


2) Download GEANT4 A SIMULATION TOOLKIT

- *Go to the Geant4 webpage:*

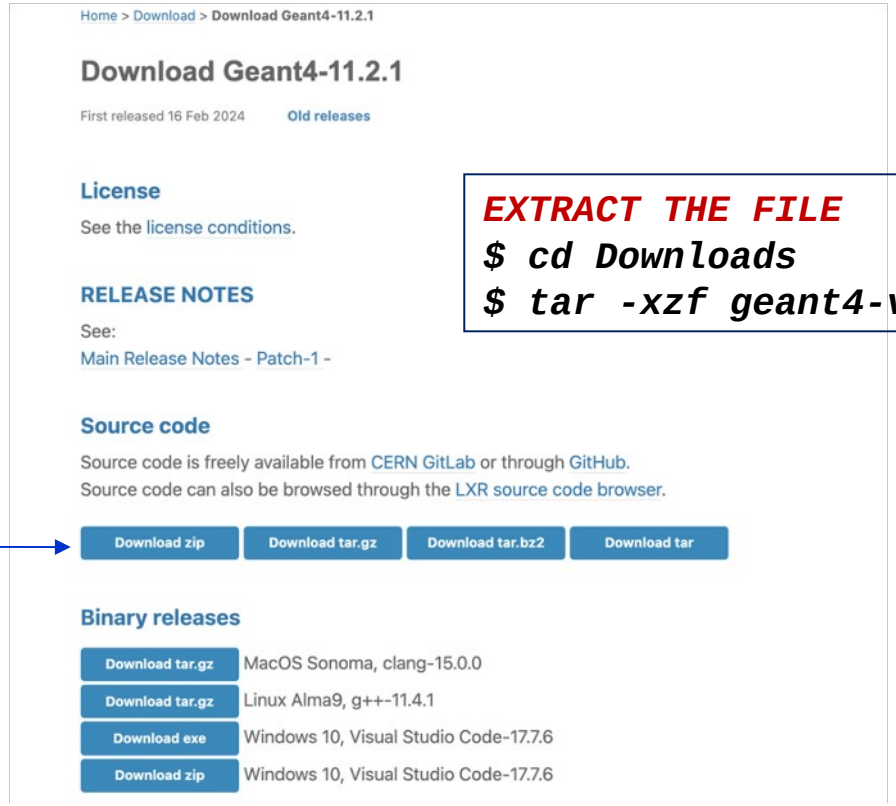
  http://geant4.org

Click here



… download GEANT4 A SIMULATION TOOLKIT

http://www.geant4.org/geant4/support/download



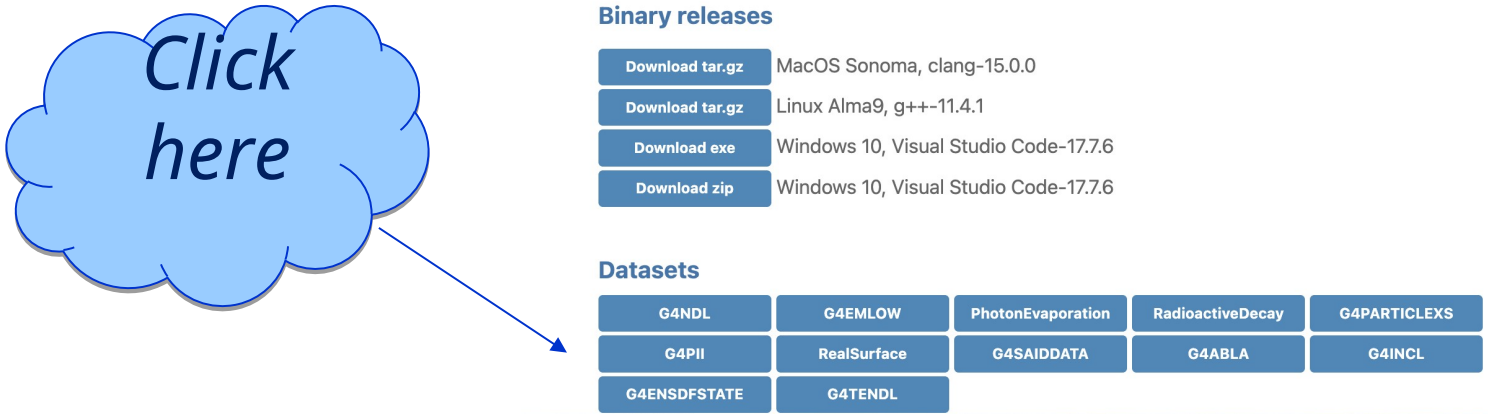
Click here

EXTRACT THE FILE

```
$ cd Downloads
$ tar -xzf geant4-v11.3.2.tar.gz
```

Download data (optional)

Option 1: *download manually* *(slow connections)*



Option 2: *use* ***CMake*** *to download data automatically* *(preferred)*

Pre-3) Directories for installation

***Source directory**: where you unpack the source*

`/usr/local/geant4/geant4-v11.3.2` VM

***Build directory**: where you run **CMake** and build Geant4 ("working directory")*

`/usr/local/geant4/geant4-v11.3.2-build` VM

***Installation directory**: where you install Geant4 to and which the applications compile against*

`/usr/local/geant4/geant4-v11.3.2-install` VM

*Only the **installation dir** is necessary to compile & run user apps.*

3) Configuration with **CMake**

- *Extract the package into* ***source directory***

  `tar -xzf geant4-v11.3.2`

- *Create the* ***build directory***

  `mkdir geant4-build`

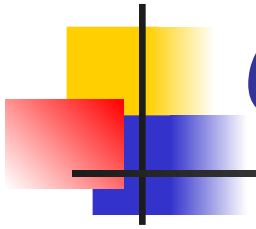
  Choose name to your liking

- ***Run*** *CMake in the* ***build directory***

  `cd geant4-build`

  `cmake [options...] ../geant4-v11.3.2`

CMake configuration options



Install directory

Important options:

- *-DCMAKE_INSTALL_PREFIX=...installation_path...*
- *-DGEANT4_INSTALL_DATA=ON/OFF*
- *-DGEANT4_BUILD_MULTITHREADED=ON/OFF*

Further options:

- *-DGEANT4_USE_OPENGL_X11=ON/OFF*
- *-DGEANT4_USE_QT=ON/OFF*
- *-DCMAKE_BUILD_TYPE=Release/Debug/RelWithDebInfo*
- *...*

https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/html/installguide.html#geant4-build-options

Running CMake

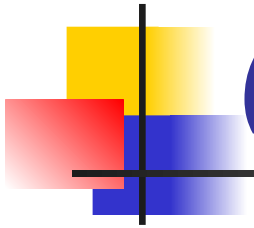
- CMake configures the build and generates Unix ***Makefiles*** to perform the actual build:

```
cmake -DGEANT4_INSTALL_DATA=ON -DGEANT4_BUILD_MULTITHREADED=OFF
-DCMAKE_INSTALL_PREFIX=/usr/local/geant4/geant4-v11.3.2-install
/usr/local/geant4/geant4-v11.3.2
```

```
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -
works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
...(~50 lines)...
-- Configuring done
-- Generating done
-- Build files have been written to:
/usr/local/geant4/geant4-v11.3.2-build
```

If you see that, you are successful !!!

*If you see **errors** at this point, carefully check the messages output by CMake*



(Random) installation notes

- *Windows: See the installation guide (and good luck!)*

http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/InstallationGuide/html/installguide.html#on-windows-platforms

- *Binary packages: Installation without compiling Geant4 is possible (but **not recommended**)*
- *Data packages: If you haven't used CMake to download them, unpack the downloaded files in the **share/Geant4-v11.3.2/data/** sub-directory of your installation*

4) Compile

- Run *make* (and get a cup of coffee)

`make -j2`

Tip: *If you have a* ***multi-core machine****, you can run the compilation in parallel using multiple jobs. Just add the* *–jN* *parameter, where N is the number of cores*

```
Scanning dependencies of target G4ENSDFSTATE
Scanning dependencies of target G4NDL
[  0%] Creating directories for 'G4ENSDFSTATE'
[  0%] Creating directories for 'G4NDL'
[  0%] Performing download step (download, verify and
extract) for 'G4NDL'
...(4029 lines, ~1 hour of execution)
[100%] Built target G4visXXX
[100%] Building CXX object
source/visualization/gMocren/CMakeFiles/G4GMocren.dir/src
/G4GMocrenIO.cc.o
[100%] Building CXX object
source/visualization/gMocren/CMakeFiles/G4GMocren.dir/src
/G4GMocrenMessenger.cc.o
[100%] Linking CXX shared library
../../../BuildProducts/lib64/libG4GMocren.so
[100%] Built target G4GMocren
```

If you see that, you are successful !!!

… and install

- Run *make install* (this takes much less time)

```
make install
```

```
[  0%] Built target G4ENSDFSTATE
[  0%] Built target G4NDL
[  0%] Built target PhotonEvaporation
[  0%] Built target RadioactiveDecay
[  0%] Built target G4ABLA
...(42830 lines, ~2 minute of execution)
-- Installing:
/usr/local/geant4/geant4-v11.3.2-install/include/Geant4/G4VModelCommand.hh
-- Installing:
/usr/local/geant4/geant4-v11.3.2-install/include/Geant4/G4VModelFactory.hh
-- Installing:
/usr/local/geant4/geant4-v11.3.2-install/include/Geant4/G4VTrajectoryModel.hh
-- Installing:
/usr/local/geant4/geant4-v11.3.2-install/include/Geant4/G4VisTrajContext.hh
-- Installing:
/usr/local/geant4/geant4-v11.3.2-install/include/Geant4/G4VisTrajContext.icc
```

My script I use to install Geant4 on any machine

```
cp Downloads/geant4-v11.3.2.tar.gz /usr/local/geant4/
cd /usr/local/geant4/
tar -xzf geant4-v11.3.2.tar.gz

mkdir geant4-v11.3.2-build
mkdir geant4-v11.3.2-install
cd geant4-v11.3.2-build

cmake -DCMAKE_INSTALL_PREFIX=/usr/local/geant4/geant4-
v11.3.2-install -DGEANT4_INSTALL_DATA=ON
-DGEANT4_USE_OPENGL_X11=ON -DGEANT4_USE_QT=ON
-DGEANT4_BUILD_MULTITHREADED=ON
-DCMAKE_BUILD_TYPE=Release /usr/local/geant4/geant4-v11.3.2

make -j2
make install
cd ..
```

5) Set-up your environment

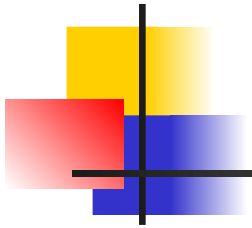
- *Geant4 needs properly set environment variables:*

```
G4ABLADATA=
G4ENSDFSTATEDATA=
G4LEDATA=
…
```

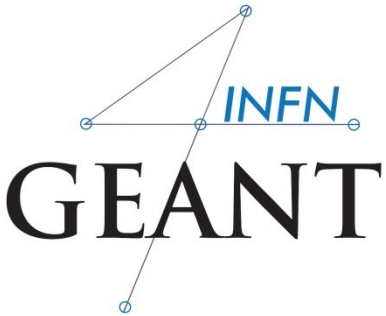
- *To set them up properly in your shell, run the script in Geant4 installation directory:*

```
source /usr/local/geant4/geant4-v11.3.2-install/bin/geant4.sh
```

- *You can put this line your ~/.bashrc file (or similar for other shells)*



Your GEANT4 A SIMULATION TOOLKIT *is ready now*



Building a GEANT4 application

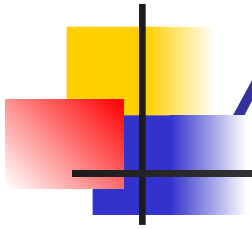
Alexei Sytov

INFN – Ferrara Division

sytov@fe.infn.it

Material by P. Cirrone, J. Pipek

XXII Seminar on Software for Nuclear, Subnuclear and Applied Physics
10th June 2025



Application build checklist

1. Properly organize your code into directories
2. Prepare a **`CMakeLists.txt`** file
3. Create a build directory and run CMake
4. Compile (make) the application
5. Run the application

1) Structure of an application

Official basic/B1 example:

```
2,4K  4 Dic 14:48 CMakeLists.txt
475B  4 Dic 14:48 GNUmakefile
2,8K  4 Dic 14:48 History
7,5K  4 Dic 14:48 README
4,0K  4 Dic 14:48 exampleB1.cc
226B  4 Dic 14:48 exampleB1.in
 35K  4 Dic 14:48 exampleB1.out
272B  4 Dic 14:49 include
338B  4 Dic 14:48 init_vis.mac
553B  4 Dic 14:48 run1.mac
448B  4 Dic 14:48 run2.mac
272B  4 Dic 14:49 src
3,8K  4 Dic 14:48 vis.mac
```

Macro file containing the commands

The text file **`CMakeLists.txt`** *is the CMake script containing commands which describe how to build the exampleB1 application*

contains **`main()`** *for the application*

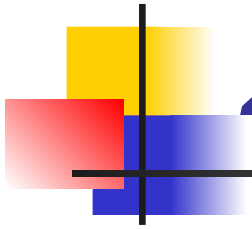
Header files

```
2,2K  4 Dic 14:48 B1ActionInitialization.hh
2,4K  4 Dic 14:48 B1DetectorConstruction.hh
2,4K  4 Dic 14:48 B1EventAction.hh
2,7K  4 Dic 14:48 B1PrimaryGeneratorAction.hh
2,5K  4 Dic 14:48 B1RunAction.hh
2,4K  4 Dic 14:48 B1SteppingAction.hh
```

Note: *Recommended, not enforced!*

Source files

```
2,9K  4 Dic 14:48 B1ActionInitialization.cc
7,7K  4 Dic 14:48 B1DetectorConstruction.cc
2,6K  4 Dic 14:48 B1EventAction.cc
4,3K  4 Dic 14:48 B1PrimaryGeneratorAction.cc
5,8K  4 Dic 14:48 B1RunAction.cc
3,2K  4 Dic 14:48 B1SteppingAction.cc
```



2) CMake

CMake *is a build configuration tool*

it takes configuration file (`CMakeLists.txt`)

it finds all dependencies (in our case, Geant4)

- *there might be others, e.g. ROOT, MySql, …*

creates ***`Makefile`*** *to run the compilation itself*

You have to write this `CMakeLists.txt` file

take inspiration in examples directories

be sure to set the name of your application correctly

specify all auxiliary files you need

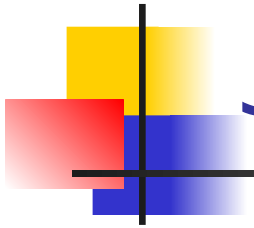
Note: *It is possible but* ***discouraged*** *to base build on GNU* ***make*** *instead of* ***CMake***.

CMakeList.txt – an example

File structure

1) Cmake minimum version and **project name**
2) Find and configure G4
3) Configure the project to use G4 and B1 headers
4) List the **sources**
5) Define and link the **executable**
6) Set and copy any macro files to the build directory

```
cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
project(B1)
option(WITH_GEANT4_UIVIS "Build example with Geant4 UI
and Vis drivers" ON)
if(WITH_GEANT4_UIVIS)
  find_package(Geant4 REQUIRED ui_all vis_all)
else()
  find_package(Geant4 REQUIRED)
endif()

include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include)

file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)

add_executable(exampleB1 exampleB1.cc ${sources} $
{headers})

target_link_libraries(exampleB1 ${Geant4_LIBRARIES})

set(EXAMPLEB1_SCRIPTS
  exampleB1.in
  exampleB1.out
  init_vis.mac
  run1.mac
  run2.mac
  vis.mac  )

foreach(_script ${EXAMPLEB1_SCRIPTS})
  configure_file(
    ${PROJECT_SOURCE_DIR}/${_script}
    ${PROJECT_BINARY_DIR}/${_script}
    COPYONLY)
```



3) Build directory & CMake

*If modifying the Geant4 examples, copy them to your **$HOME** first:*

```
cp -r /usr/local/geant4/geant4-v11.3.2/examples/basic/B1 ~
```

*Create a **build directory***, where the compiled application will be put:*

```
mkdir -p ~/B1-build
cd ~/B1-build
```

***Note:** It is possible (though not recommended) to compile **inside** source directory.*

Run CMake

*In the **build** directory you just created, run **CMake***

Path to Geant4

```
cmake -DGeant4_DIR=/usr/local/geant4/geant4-v11.3.2-install/lib64/cmake/Geant4 ~/B1/
```

Path to source

```
-- The C compiler identification is GNU 4.8.5
-- The CXX compiler identification is GNU 4.8.5
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /path/to/build/directory
```

4) Compilation

In the build directory, run make

```
make
```

(and don't get a cup of coffee)

You have only a couple of files, it should be ready in a minute

*An **executable** with the name of your application is created (e.g. exampleB1) in build directory*

***Macros** and other auxiliary files are copied into build directory*

```
Scanning dependencies of target exampleB1
[ 12%] Building CXX object CMakeFiles/exampleB1.dir/exampleB1.cc.o
[ 25%] Building CXX object CMakeFiles/exampleB1.dir/src/B1RunAction.cc.o
[ 37%] Building CXX object CMakeFiles/exampleB1.dir/src/B1SteppingAction.cc.o
[ 50%] Building CXX object
CMakeFiles/exampleB1.dir/src/B1DetectorConstruction.cc.o
[ 62%] Building CXX object
CMakeFiles/exampleB1.dir/src/B1PrimaryGeneratorAction.cc.o
[ 75%] Building CXX object CMakeFiles/exampleB1.dir/src/B1EventAction.cc.o
[ 87%] Building CXX object
CMakeFiles/exampleB1.dir/src/B1ActionInitialization.cc.o
[100%] Linking CXX executable exampleB1
[100%] Built target exampleB1
```

5) Running the application - GUI

Just type the name of your application, including the `./` *identifier of current directory (e.g.* `./exampleB1`*)*

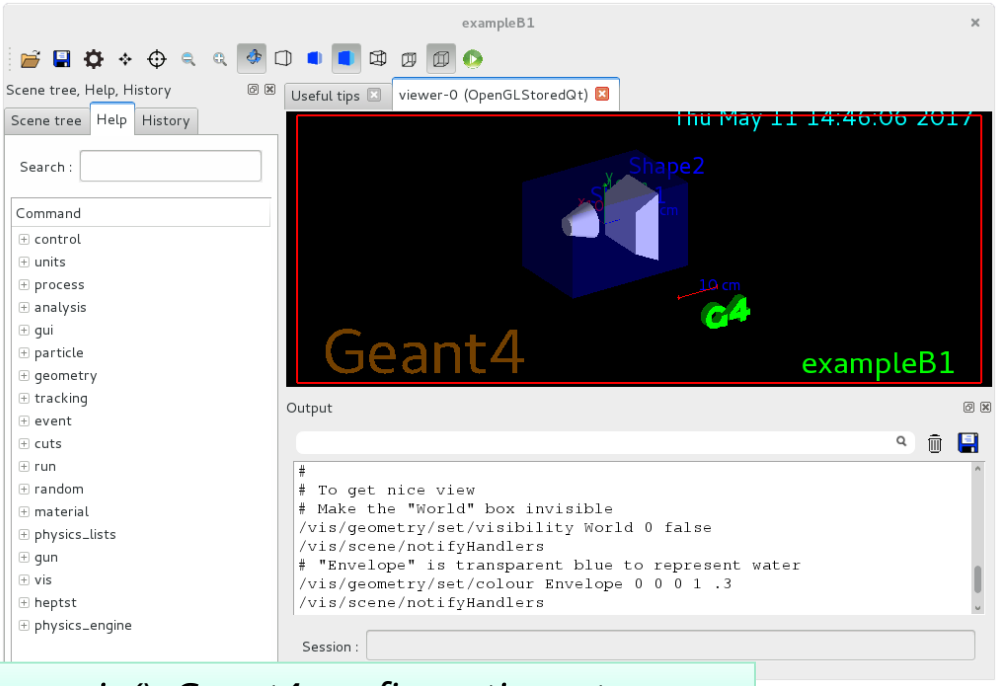
By default, ***graphical user interface (GUI)*** *is started**

```
./exampleB1
```

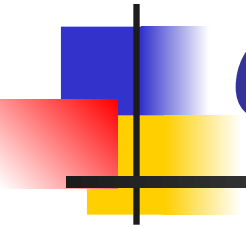
```
Available UI session types: [ Qt, GAG, tcsh, csh ]
```

Run without GUI but with macro:

```
./exampleB1 run1.mac
```



***Note:** Depends on your application main(), Geant4 configuration, etc.*



Conclusions

Building an application is easy ☺