

XIV Seminar on Software for Nuclear, Subnuclear and Applied Physics

Alghero (ITALY)

04-09 June 2017

Geant4 Installation

Geant4 tutorial



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

① Supported platforms & requirements

- **Operating system**

- “recent” Linux (e.g. CentOS 7), best support
- macOS 10.10+
- Windows 7+ (limited support, not recommended)

Virtual Machine:
CentOS 7 with gcc 4.8.5

- **Compilers**

- C++11 compliance
- such as GCC 4.8.5+, clang 3.6+, Visual C++ 14.0 (2015)

- **CMake** (configuration generation tool) 3.3+

- **System libraries** (as development packages):

- expat, xerces-c

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

CMake installation (if not provided)

- 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 3.3+ is not available:

1. [download](http://www.cmake.org/) the latest version (*http://www.cmake.org/*)
 2. [unzip](#) the tar-ball
 3. **./bootstrap, make, make install**
- **macOS:** install it using the Darwin64 [dmg installerpackage](#)
 - **Windows:** install it using the Win64/32 [exe installerpackage](#)

Note: You may also want to install [ccmake](#) and/or [cmake-gui](#) tools for user-friendly configuration

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...

② Download Geant4...

- Go to the Geant4 webpage:

<http://geant4.web.cern.ch/geant4/>



Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The three main reference papers for Geant4 are: *Nuclear Instruments and Methods in Physics Research A* **506** (2003) 250-303, *IEEE Transactions on Nuclear Science* **53** No. 1 (2006) 270-278, *Nuclear Instruments and Methods in Physics Research A* **835** (2016) 186-225.

[Applications](#)

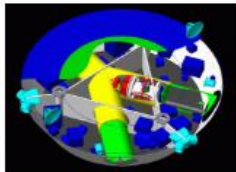
[User Support](#)

[Publications](#)

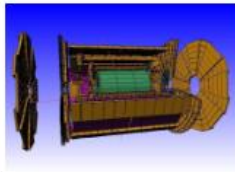
[Collaboration](#)



[A sampling of applications, technology transfer and other uses of Geant4](#)



[Getting started, guides and information for users and developers](#)



[Validation of Geant4, results from experiments and publications](#)



[Who we are: collaborating institutions, members, organization and legal information](#)

News

- 8 March 2017 - [2017 planned developments](#).
- 28 February 2017 - **Patch-01 to release 10.3** is available from the [Download](#) area.
- 27 January 2017 - **Patch-03 to release 10.2** is available from the [source archive](#) area.
- 17 February 2016 - **Patch-03 to release 10.1**



...download Geant4...

<http://geant4.web.cern.ch/geant4/support/download.shtml>



Geant 4 [Download](#) | [User Forum](#) | [Gallery](#) | [Contact Us](#)

Home > [User Support](#) > Download

Geant4 Software Download

Geant4 10.3

first released 9 December 2016 (patch-01, released 24 February 2017)

The Geant4 source code is freely available. See the [licence conditions](#).

Please read the [Release Notes](#) before downloading or using this release. The patch below contains bug fixes to release 10.3, we suggest you to download and apply the latest patch for release 10.3 (see the additional notes for [patch-01](#)), or download the complete source with the patch applied; in any case, it is required to apply a full rebuild of the libraries.

Source files

Please choose the archive best suited to your system and archiving tool:

Download	GNU or Linux tar format, compressed using gzip (31.0Mb, 32515303 bytes) <i>After downloading, gunzip, then unpack using GNU tar.</i>
Download	ZIP format (44.2Mb, 46375046 bytes) <i>After downloading, unpack using e.g. WinZip.</i>

Data files (*)

For specific, optional physics processes some of the following files are required. The file format is compatible with Unix, GNU, and Windows utilities.

Download	G4NDL4.5, Neutron data files with thermal cross-sections - version 4.5 (402.2Mb, 421710294 bytes)
Download	G4EMLOW6.50, data files for low energy electromagnetic processes - version 6.50 (27.0Mb, 28334495 bytes) NEW

[Search Geant4](#)

Related Links


- [Previous Releases of Geant4](#) (since release 9.6).
- [LXR source code browser](#).
- [GitHub](#)
- [GitLab @ CERN](#)
- [Windows CygWin installation note](#).

Download data (optional)

Alternative 1: download everything (slow connections)

Data files (*)

For specific, optional physics processes some of the following files are required. The file format is compatible with Unix, GNU, and Windows utilities.



Download	G4NDL4.5, Neutron data files <u>with</u> thermal cross-sections - version 4.5 (402.2Mb, 421710294 bytes)
Download	G4EMLOW6.50, data files for low energy electromagnetic processes - version 6.50 (27.0Mb, 28334495 bytes) ^{NEW}
Download	G4PhotonEvaporation4.3.2, data files for photon evaporation - version 4.3.2 (18.5Mb, 19392015 bytes) ^{NEW}
Download	G4RadioactiveDecay5.1.1, data files for radioactive decay hadronic processes - version 5.1.1 (1.0Mb, 1057172 bytes) ^{NEW}
Download	G4SAIDDATA1.1, data files from evaluated cross-sections in SAID data-base - version 1.1 (25.2kb, 25800 bytes)
Download	G4NEUTRONXS1.4, data files for evaluated neutron cross-sections on natural composition of elements - version 1.4 (2.1Mb, 2249001 bytes)
Download	G4ABLA3.0, data files for nuclear shell effects in INCL/ABLA hadronic mode - version 3.0 (53.6kb, 54849 bytes)
Download	G4PII1.3, data files for shell ionisation cross-sections - version 1.3 (4.1Mb, 4293607 bytes)
Download	RealSurface1.0, Optional data files for measured optical surface reflectance - version 1.0 (1.2Mb, 1257863 bytes)
Download	G4ENSDFSTATE2.1, Data files for nuclides properties - version 2.1 (283.9kb, 290690 bytes) ^{NEW}
Download	G4TENDL1.3, Optional data files for incident particles - version 1.3 (560.3Mb, 587566993 bytes) ^{NEW}

- Low Energy Nuclear Data (LEND) files can be downloaded from: <ftp://gdo-nuclear.ucllnl.org/pub/>

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

③ Configuration with CMake

- Extract the package into **source directory**

```
tar xzf geant4.10.03.p01.tag.gz
```

- Create the **build directory**

```
mkdir geant4-build
```

- Run CMake in the **build directory**

```
cd geant4-build
```

```
cmake [options...] ../geant4.10.03.p01
```

See next slide for directory explanation...

Choose name to your liking

Directories for installation

Source directory: where you unpack the source

`/usr/local/geant4/geant4.10.03.p01` 

Build directory: where you run CMake and build Geant4 (“working directory”)

`/usr/local/geant4/geant4.10.03.p01-build` 

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

`/usr/local/geant4/geant4.10.03.p01-install` 

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

CMake configuration options

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`
- ...

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/geant4.10.03.p01-install /usr/local/geant4/geant4.10.03.p01
```

```
-- 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/geant4.10.03.p01-build
```

If you see that, you are successful !!!



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



④ Compile...

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

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 required number of jobs (it is recommended to set this to the number of your processor's cores), e.g.:

```
make -j2
```



```
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 G4ENSDSTATE
[ 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/geant4.10.03.p01-install/include/Geant4/G4VModelCommand.hh
-- Installing: /usr/local/geant4/geant4.10.03.p01-install/include/Geant4/G4VModelFactory.hh
-- Installing: /usr/local/geant4/geant4.10.03.p01-install/include/Geant4/G4VTrajectoryModel.hh
-- Installing: /usr/local/geant4/geant4.10.03.p01-install/include/Geant4/G4VisTrajContext.hh
-- Installing: /usr/local/geant4/geant4.10.03.p01-install/include/Geant4/G4VisTrajContext.icc
```

Tip: If you want to combine the two steps (compilation + installation) into one, you can leave out the first step.

(Random) installation notes

- **Windows:** See the installation guide
<https://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/ch02s02.html>
- **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/Geant4-10.3.1/data/](#) sub-directory of your installation

⑤ Set-up environment

- Geant4 need properly set environment variables:

```
G4ABLADATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4ABLA3.0"  
G4ENSDFSTATEDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4ENSDFSTATE2.1"  
G4LEDDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4EMLW6.50"  
G4LEVELGAMMADATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/PhotonEvaporation4.3.2"  
G4NEUTRONHPDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4NDL4.5"  
G4NEUTRONXSDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4NEUTRONXS1.4"  
G4PIIDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4PII1.3"  
G4RADIOACTIVEDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/RadioactiveDecay5.1.1"  
G4REALSURFACEDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/RealSurface1.0"  
G4SAIDXSDATA="/usr/local/geant4/geant4.10.03.p01-install/share/Geant4-10.3.1/data/G4SAIDDATA1.1"  
LD_LIBRARY_PATH="...:/usr/local/geant4/geant4.10.03.p01-install/lib64"  
PATH="...:/usr/local/geant4/geant4.10.03.p01-install/bin"
```

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

```
source /usr/local/geant4/geant4.10.03.p01-install/bin/geant4.(c)sh
```

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

Your Geant4 is ready now.



**Can we continue to...
build an application?**

Build a Geant4 application

Application build process

- 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

Note: Recommended, not enforced!

① Application source structure in Geant4

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
```

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 (again)

- **CMake** is a build configuration tool
 - it takes configuration file ([CMakeLists.txt](#))
 - it finds all dependencies (in our case, [Geant4](#))
 - 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.

CMakeLists.txt

```
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()
```

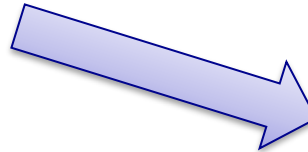
```
include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include)
```

```
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
  )
endforeach
```



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) Copy any macro files to the build directory

③ Build directory and CMake

- 1) If modifying the Geant4 examples, copy them to your \$HOME first:

```
cp -r /usr/local/geant4/geant4.10.03.p01/examples/basic/B1 ~
```

- 2) 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:

```
cmake -DGeant4_DIR=/usr/local/geant4/geant4.10.03.p01-install/lib64/Geant4-10.3.1/ ~/B1/
```

Path to Geant4

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**

(and don't get a cup of coffee)

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

```
make -j2
```

```
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
```

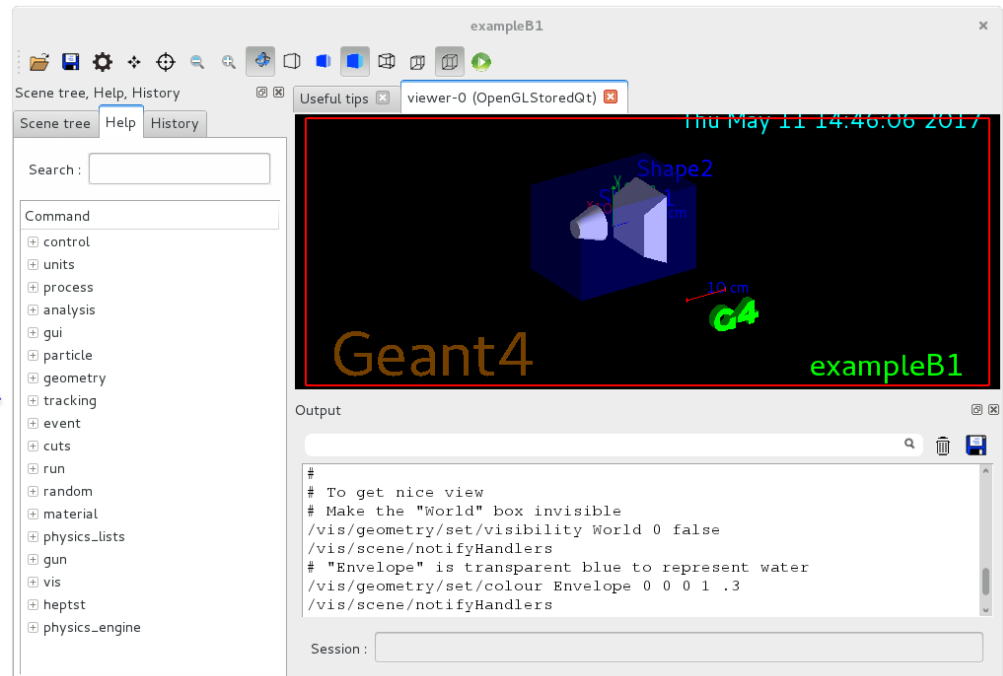


⑤ Run the application - GUI

- Just type the name of your application, including the `./` identifier of current directory (e.g. `./exampleB1`)
- By default, **graphical user interface** is started*

`./exampleB1`

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



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

Conclusion

Building an application is easy 😊