How to install Geant 4 and build an application

IV International Geant4 School
October 23, 2016 - October 28, 2016
Belgrade - Serbia

Supported platforms & compilers

Geant4.10.2.p02

- Linux systems
  - Scienti… CERN 6 with gcc >=4.8.2, 4.9.X, >=5.2.…

    Geant4 has a… …ributions, including Debian, Ubuntu and open… …d)

Virtual Machine : CentOS7 Linux with gcc 4.8.5

- MacOSX systems

- Mac OS X 10.10 (Yosemite) with Apple-LLVM (Xcode) 7.0, 64bit

  Geant4 has also been successfully compiled on Mac OS X 10.7 (Lion) with clang 3.1 (Apple), (not officially supported)

- Windows systems
  - Windows 7 with Visual Studio (VS2013 or VS2015).

Check current Geant4 supported platforms in **http://geant4.web.cern.ch/geant4/**

Required software

- The Geant4 toolkit source code (10.02.p02)
- C++11 compiler: GCC >=4.8.2 to support C++11 installed
  - It is usually installed on your Linux. If not, you need to install it (*not shown here*)
- CMake 3.3 or higher
- CLHEP library
  - an internal version is now supplied with the geant4 source (since 9.5 version)
- The Geant4 data files
  - an automatic procedure can retrieve them (with CMake)

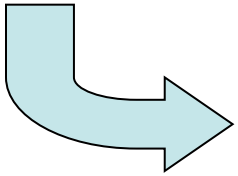
External software packages

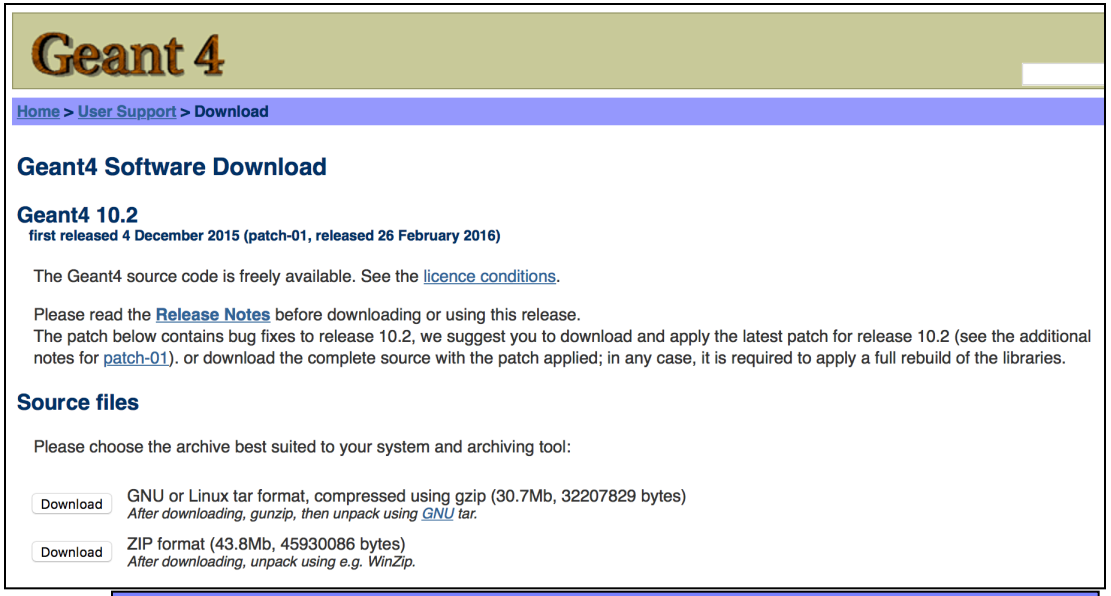
Suggested tools (optional):

- X11 OpenGL Visualization (Linux and Mac OS X)
  - Requires: X11, OpenGL or MesaGL (headers and libraries).
- Qt User Interface and Visualization (All Platforms)
  - Requires: Qt4, OpenGL or MesaGL (headers and libraries).
- Motif User Interface and Visualization (Linux and Mac)
  - Requires: Motif and X11, OpenGL or MesaGL headers and libraries.
- Open Inventor Visualization (All Platforms)
- X11 RayTracer Visualization (Linux and Mac OS X)
- DAWN postscript renderer
- HepRApp Browser
- VRML browser
- WIRED4 JAS Plug-In
- GDML Support (All Platforms)
- AIDA (Abstract Interface for Data Analysis)

Where to download the packages

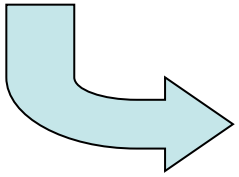
- **Geant4**

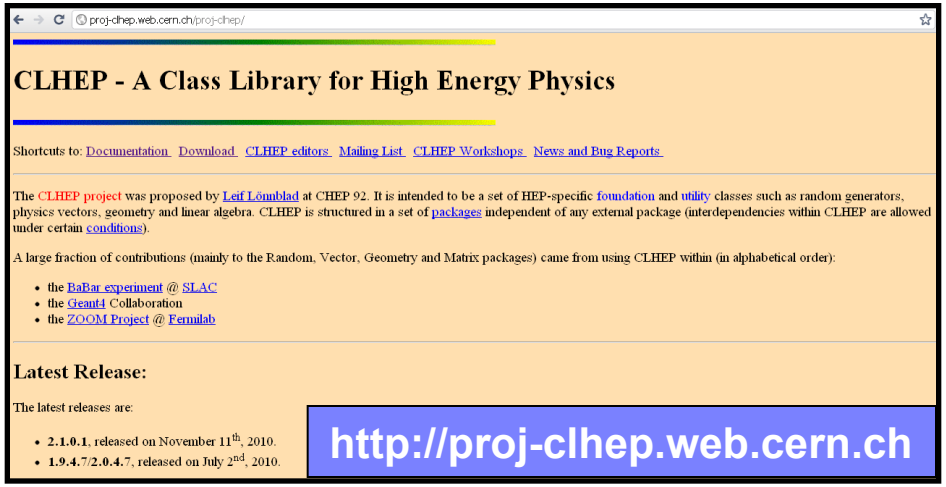




http://geant4.cern.ch/support/download.shtml

- **CLHEP**



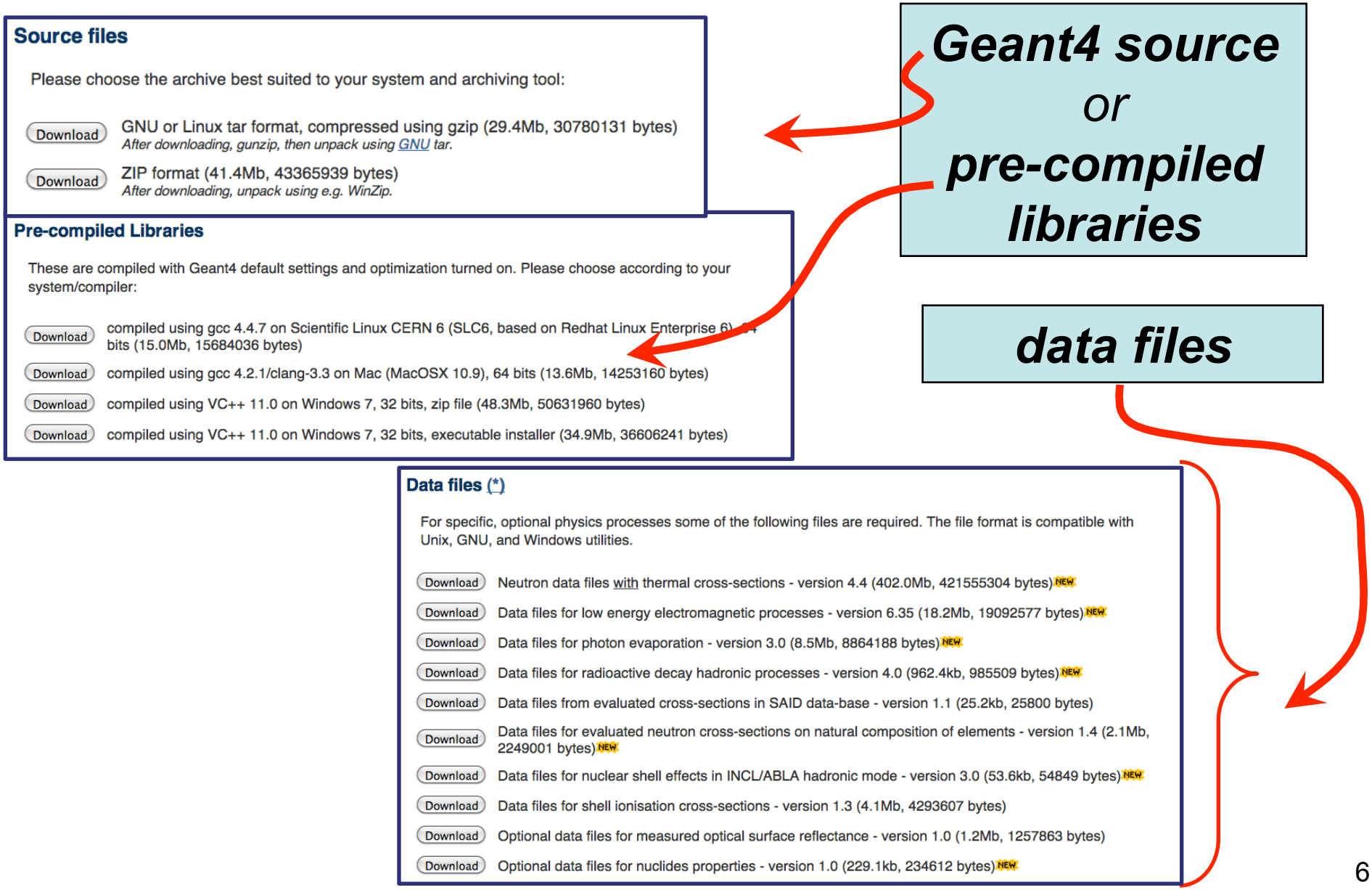


http://proj-clhep.web.cern.ch

Downloading Geant4 and data files

Geant4 source
or
pre-compiled libraries

data files



Source files

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

Download — GNU or Linux tar format, compressed using gzip (29.4Mb, 30780131 bytes)
After downloading, gunzip, then unpack using GNU tar.

Download — ZIP format (41.4Mb, 43365939 bytes)
After downloading, unpack using e.g. WinZip.

Pre-compiled Libraries

These are compiled with Geant4 default settings and optimization turned on. Please choose according to your system/compiler:

Download — compiled using gcc 4.4.7 on Scientific Linux CERN 6 (SLC6, based on Redhat Linux Enterprise 6), 64 bits (15.0Mb, 15684036 bytes)

Download — compiled using gcc 4.2.1/clang-3.3 on Mac (MacOSX 10.9), 64 bits (13.6Mb, 14253160 bytes)

Download — compiled using VC++ 11.0 on Windows 7, 32 bits, zip file (48.3Mb, 50631960 bytes)

Download — compiled using VC++ 11.0 on Windows 7, 32 bits, executable installer (34.9Mb, 36606241 bytes)

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 — Neutron data files with thermal cross-sections - version 4.4 (402.0Mb, 421555304 bytes) NEW

Download — Data files for low energy electromagnetic processes - version 6.35 (18.2Mb, 19092577 bytes) NEW

Download — Data files for photon evaporation - version 3.0 (8.5Mb, 8864188 bytes) NEW

Download — Data files for radioactive decay hadronic processes - version 4.0 (962.4kb, 985509 bytes) NEW

Download — Data files from evaluated cross-sections in SAID data-base - version 1.1 (25.2kb, 25800 bytes)

Download — Data files for evaluated neutron cross-sections on natural composition of elements - version 1.4 (2.1Mb, 2249001 bytes) NEW

Download — Data files for nuclear shell effects in INCL/ABLA hadronic mode - version 3.0 (53.6kb, 54849 bytes) NEW

Download — Data files for shell ionisation cross-sections - version 1.3 (4.1Mb, 4293607 bytes)

Download — Optional data files for measured optical surface reflectance - version 1.0 (1.2Mb, 1257863 bytes)

Download — Optional data files for nuclides properties - version 1.0 (229.1kb, 234612 bytes) NEW

Geant4 installation (10.2 version)

Working area & installation area

- Why two different areas ?
  - To allow centralized installation of the Geant4 kernel libraries and related sources in a multi-user environment
  - To decouple user-developed code and applications from the kernel
  - To allow an easy integration of the Geant4 software in an existing software framework

Two ways to proceed:

- Manually installing by env variables *(deprecated)*
- Using **CMake** *(recommended and officially supported)*

Installing Geant4 with *CMake*

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:
  - On Linux: it is recommended to use the CMake provided by the package management system of your distribution.

    In case it does not meet the minimum version requirement:
    1. download the latest version *(http://www.cmake.org/)*
    2. unzip the tar-ball
    3. `./bootstrap`, `make`, `make install`
  - On Mac: install it using the Darwin64 dmg installerpackage
  - On Windows: install it using the Win32 exe installerpackage

Geant4 installation with CMake

- Unpack the geant4 source package geant4.10.02.p02.tar.gz to a location of your choice:
  - ex.: /path/to/geant4.10.02.p02 → ***source directory***
- Create a directory in which to configure and run the build and store the build products (not inside the source dir!)
  - ex.: /path/to/geant4.10.02.p01-build → ***build directory***

```
$ mkdir /path/to/geant4.10.02.p02-build
```

- To configure, change into the build directory and run CMake:

```
$ cd /path/to/geant4.10.02.p02-build
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.02.p02-install
  /path/to/geant4.10.02.p02
```

  - CMAKE_INSTALL_PREFIX option is used to set the *install directory*
  - The second argument to CMake is the path to the source directory.

Geant4 installation with CMake

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

```
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.0-install /path/to/geant4.10.00
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- setting default compiler flags for CXX
-- 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
-- Found EXPAT: /usr/lib64/libexpat.so
-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for stddef.h
-- Looking for stddef.h -
```

If you see that, you are successful !!!

```
-- Configuring done
-- Generating done
-- Build files have been written to: /path/to/geant4.10.0-build
```

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

Geant4 installation with CMake

- After the configuration has run, CMake have generated Unix Makefiles for building Geant4. To run the build, simply execute make in the build directory:

```
$ make -jN
```

  - where N is the number of parallel jobs you require. The build will now run, and will output information on the progress of the build and current operations

- When build has completed, you can install Geant4 to the directory you specified earlier in CMAKE_INSTALL_PREFIX by running:

```
$ make install
```

Geant4 installation with CMake

- Additional arguments can be passed to CMake to activate optional components of Geant4 (***standard*** and ***advanced*** options):

  - **-DGEANT4_INSTALL_DATA=ON** *(recommended)*
    the additional external data libraries are automatically downloaded
  - **-DGEANT4_USE_OPENGL_X11=ON** *(recommended)*
    build the X11 OpenGL visualization driver
  - **-DGEANT4_BUILD_MULTITHREADED=ON** *(recommended)*
    build Geant4 libraries with support for multithreading
  - **-DGEANT4_USE_QT=ON** *(optional, but nice!!!)*
    build the Qt visualization driver

Geant4 installation with CMake

- If you want to *activate* additional options, simply rerun CMake in the build directory, passing it the extra options, and repeat:

```
$ cd /path/to/geant4.10.02.p02-build
$ cmake -DGEANT4_INSTALL_DATA=ON /path/to/geant4.10.02.p02

$ make -jN

$ make install
```

- If you want to *deactivate* a previously selected option:

```
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.02.p02-install
  -DGEANT4_USE_GDML=OFF /path/to/geant4.10.02.p02
```

You may also directly include the options since the beginning:

```
cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.02.p02-install -
DGEANT4_INSTALL_DATA=ON
-DGEANT4_USE_OPENGL_X11=ON -DGEANT4_USE_QT=ON /path/to/geant4.10.02.p02
```

Geant4 installation with CMake

- The install of Geant4 is contained under the directory chosen (CMAKE_INSTALL_PATH), with the following structure:

```
+- CMAKE_INSTALL_PREFIX
   +- bin/
   |  +- geant4-config    (UNIX ONLY)
   |  +- geant4.csh       (UNIX ONLY)
   |  +- geant4.sh        (UNIX ONLY)
   |  +- G4global.dll     (WINDOWS ONLY)
   |  +- ...
   +- include/
   |  +- Geant4/
   |     +- G4global.hh
   |     +- ...
   |     +- CLHEP/        (WITH INTERNAL CLHEP ONLY)
   |     +- tools/
   +- lib/                (MAY BE lib64 on LINUX)
   |  +- libG4global.so   (AND/OR .a, OR G4Global.lib ON WINDOWS)
   |  +- ...
   |  +- Geant4-10.0.0/
```

- To make the Geant4 binaries and libraries available on your PATH and library path and to set the variables for external data libraries:

```
$ source geant4.sh
```

N.B.: each time you open a new shell remember to source the `geant4.sh` script before executing an application !!!

Building an application with CMake

Application in Geant4

For instance: examples/basic/B1:

```
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
```

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 classes

```
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 classes

```
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
```

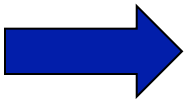
Macro file containing the commands

Building an application with cmake

- **To build an application** that uses the Geant4 toolkit, **it is necessary to include Geant4 headers in the application sources and link the application to the Geant4 libraries:**
  - using CMake → Geant4Config.cmake → writing a **`CMakeLists.txt`** script

containing commands which describe how to build the application:

```
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
    )
```



- Example of CMakeList structure:
  1. Cmake minimum version and set the 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 runtime script to the build directory
  7. Install the executable

Building an application with cmake

- **First step**: create a folder into your $HOME (ex. geant4-exercises)

```
$ mkdir geant4-exercises
```

- **Second step**: copy in the geant4-exercises folder one of the Geant4 examples, ex: the B1 example contained in the source folder

```
$ cp -r /usr/local/geant4/geant4.10.01.p02/examples/
basic/B1 $HOME
```

- **Third step**: create a build directory for the specific application (suggestion: build that alongside the application source directory):

```
$ cd $HOME
$ mkdir B1-build
```

Building an application with cmake

- Change to this build directory and run cmake to generate the Makefiles needed to build the B1 application. Pass cmake two arguments:

```
$ cd $HOME/B1-build
$ cmake -DGeant4_DIR=/path/to/geant4.10.02.p01-install/lib64/
Geant4-10.2.1 $HOME/B1
```

- cmake will now run to configure the build and generate Makefiles.:

```
$ cmake -DGeant4_DIR=/home/you/geant4-install/lib64/Geant4-10.0.0 $HOME/B1
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - 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
-- Configuring done
-- Generating done
-- Build files have been written to: /home/you/B1-build
```

Building an application with cmake

- The following files have been generated:

```
$ ls
CMakeCache.txt        exampleB1.in    init_vis.mac  run2.mac
CMakeFiles            exampleB1.out   Makefile      vis.mac
cmake_install.cmake   init.mac        run1.mac
```

- Once the Makefile is available we can do:

```
$ make -jN
```

- The following output should be displayed:

```
$ make
Scanning dependencies of target exampleB1
[ 16%] Building CXX object CMakeFiles/exampleB1.dir/exampleB1.cc.o
[ 33%] Building CXX object CMakeFiles/exampleB1.dir/src/B1PrimaryGeneratorA
ction.cc.o
[ 50%] Building CXX object CMakeFiles/exampleB1.dir/src/B1EventAction.cc.o
[ 66%] Building CXX object CMakeFiles/exampleB1.dir/src/B1RunAction.cc.o
[ 83%] Building CXX object CMakeFiles/exampleB1.dir/src/B1DetectorConstruct
ion.cc.o
[100%] Building CXX object CMakeFiles/exampleB1.dir/src/B1SteppingAction.cc
.o
Linking CXX executable exampleB1
[100%] Built target exampleB1
```

Building an application with cmake

- List again the content of the build directory, you see the executable:

```
$ ls
CMakeCache.txt       exampleB1      init.mac      run1.mac
CMakeFiles           exampleB1.in   init_vis.mac  run2.mac
cmake_install.cmake  exampleB1.out  Makefile      vis.mac
```

- Don't forget to source the geant4.sh script before executing the application!

- Run the application, simply with `./exampleB1`, the following output should be displayed:

```
$ ./exampleB1

*************************************************************
Geant4 version Name: geant4-10-00-ref-00 [MT]   (6-December-2013)
<< in Multi-threaded mode >>
Copyright : Geant4 Collaboration
Reference : NIM A 506 (2003), 250-303
WWW : http://cern.ch/geant4
*************************************************************
```

- And that's all !!!

Building an application with cmake

- For further details have a look at the Installation guide:

Geant 4

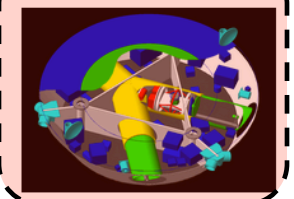
Download | User Forum | Gallery
Contact Us
Search Geant4

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas
energy, nuclear and accelerator physics, as well as studies in medical and space science
papers for Geant4 are published in *Nuclear Instruments and Methods in Physics Researc*
IEEE Transactions on Nuclear Science 53 No. 1 (2006) 270-278.

Applications

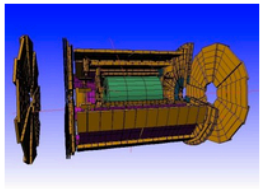
A sampling of applications, technology transfer and other uses of Geant4

User Support



Getting started, guides and information for users and developers

Publications



Validation of Geant4, results from experiments and publications

Col

Who mem infor

Events

- Geant4 Beginners Course, Belfast (Northern Ireland), **20-24 January 2014**.
- SLAC Geant4 Tutorial Course, Jen-Hsun Huang Engineering Center, Stanford (US), **3-6 Mar**
- 19th Geant4 Collaboration Meeting, Okinawa (Japan), **29 September - 4 October 2014**.
- **Past events**

Geant 4

Home > User Support

User Support

1. Getting started
2. Training courses and materials
3. Source code
   a. Download page
   b. LXR code browser -or- draft doxygen documentation
4. Frequently Asked Questions (FAQ)
5. Bug reports and fixes
6. User requirements tracker
7. User Forum
8. Documentation
   a. Introduction to Geant4
   b. Installation Guide
   c. Application Developers Guide
   d. Toolkit Developers Guide
   e. Physics Reference Manual
9. Examples
10. Physics lists
    a. Electromagnetic
    b. Hadronic
11. User Aids
    a. Tips for improving CPU performance

A Geant4 application

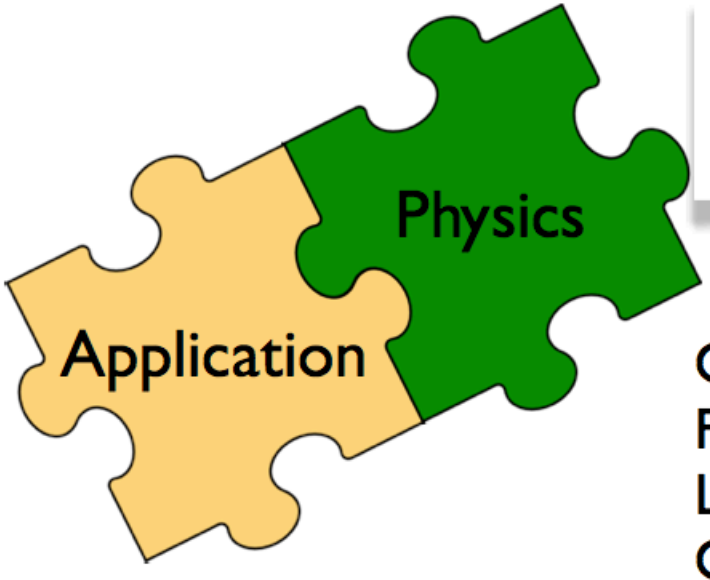
A Geant4 application

- Geant4 is a **toolkit**: no “main” program
- User is responsible of building an application
- Increased flexibility, but...

... more work to be done

A Geant4 application

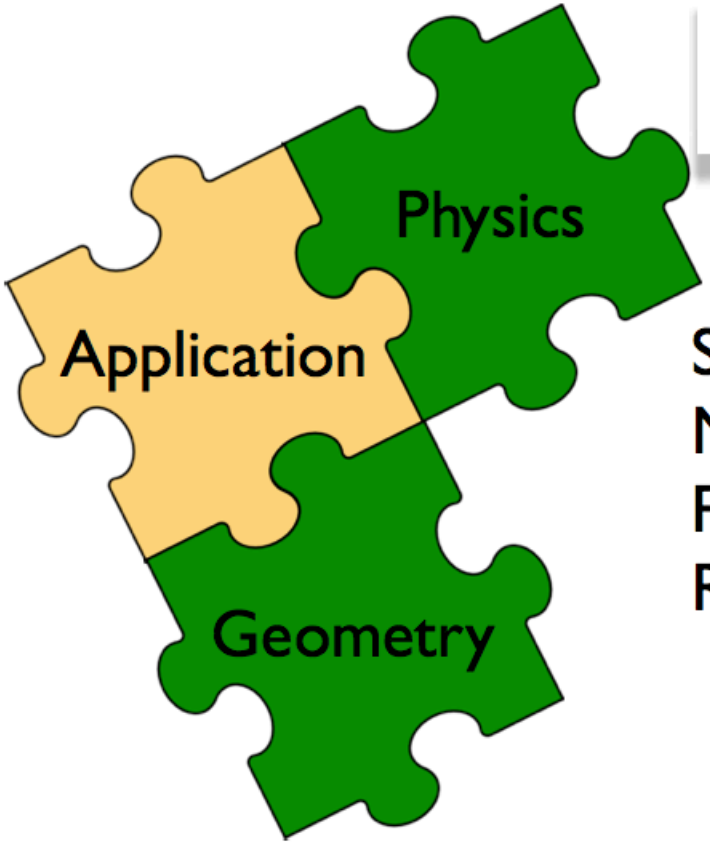
A Geant4 application



From Geant4:
One of the provided Physics lists or build/tailor your own

QGSP_BERT
FTFP_BERT
LHEP
QGSP_BIC
CHIPS
....

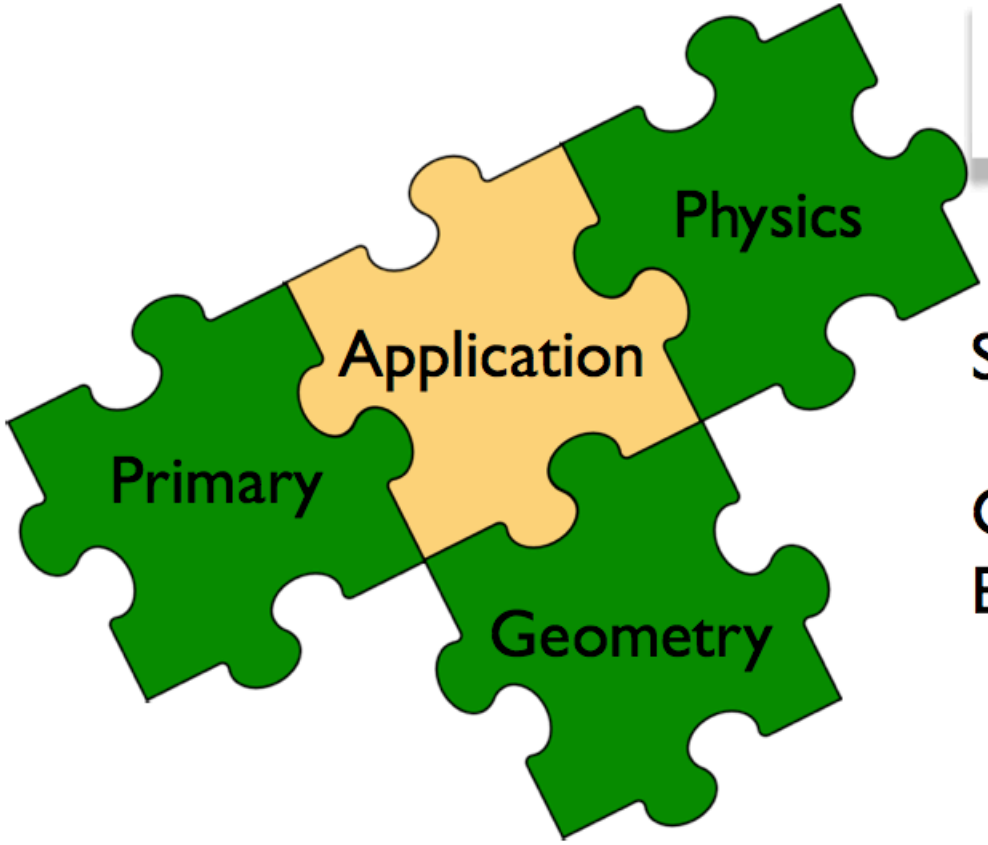
A Geant4 application



Describe your detector using G4 geometry

Shapes,
Materials,
Placements,
Read-out geo

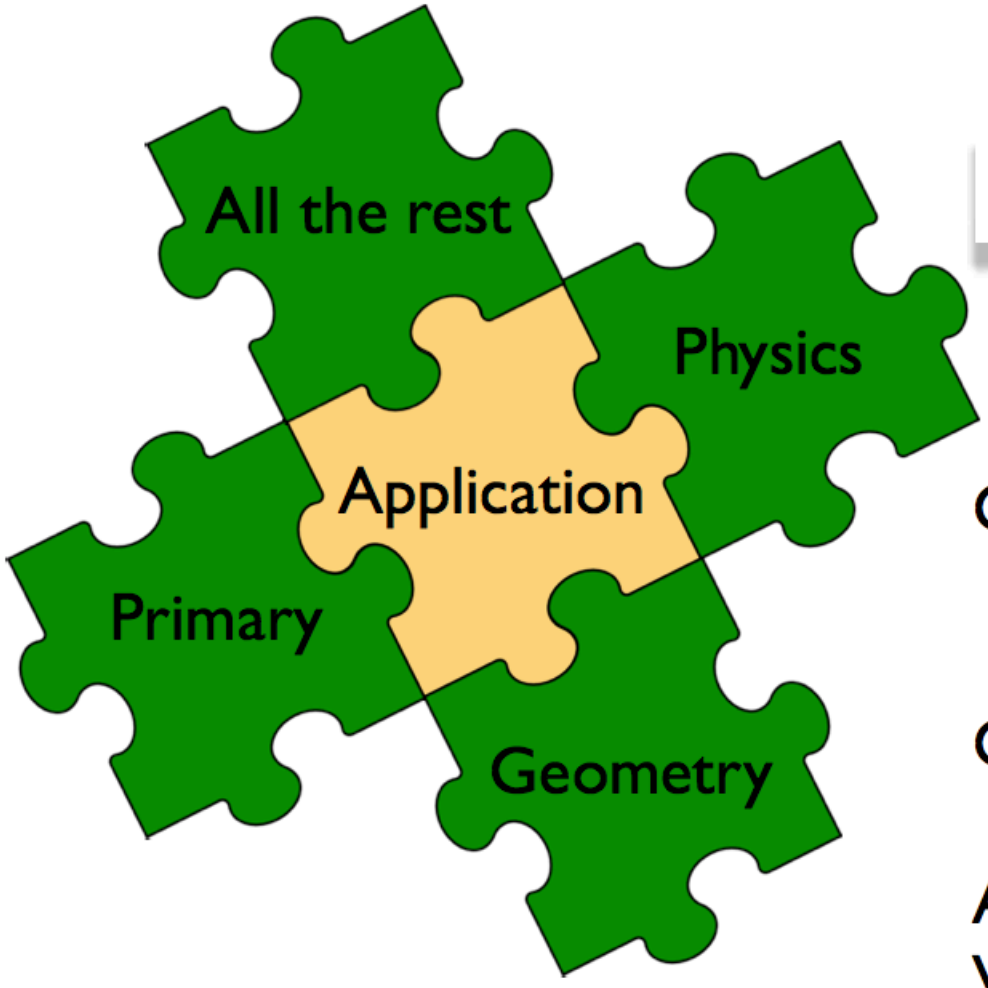
A Geant4 application



Describe the source of radiation

Simple Gun
test-beam like
Generic source
External input
HepMC files

A Geant4 application



Add all the rest

G4UserActions
interact with simulation
G4Hits/Digits
read-out
Analysis
Visualization

Thank you for your attention