GEANT4 Course for PhD

Catania, Italy
4th December 2012

How to install Geant 4 and build an application

Geant 4 tutorial course

Outline

- Supported platforms & compilers
- Required software
- Where to download the packages
- Geant4 toolkit installation *(release 9.6)*
  - Configuring the environment manually
  - Using *CMake*
- CLHEP full version installation *(optional)*
- Building a Geant4 application with Cmake
- Example of a Geant4 application

Supported platforms & compilers

- Linux systems
  - Scientific Linux CERN SLC5, with gcc 4.1.2 or 4.3.X, 32/64bit
  - Scientific Linux CERN 6 with gcc 4.6.X, 64bit

    Geant4 has also been successfully compiled on other Linux distributions, including Debian, Ubuntu and openSUSE (not officially supported)
- MacOSX systems
  - Mac OS X 10.7 (Lion) and 10.8 (Mountain Lion) with gcc 4.2.1 (Apple), 64bit

    Geant4 has also been successfully compiled on Mac OS X 10.6.8 (Snow Leopard) with gcc 4.2.1 (Apple), (not officially supported)
- Windows systems
  - Windows 7 with Visual Studio 10 (VS2010).

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

Required software

- A UNIX shell and related basic UNIX commands
- C++ compiler
  - It is usually installed on your Linux. If not, you need to install it (*not shown here*)
- Cmake 2.6.4 or higher
- The Geant4 toolkit source code
- 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 I

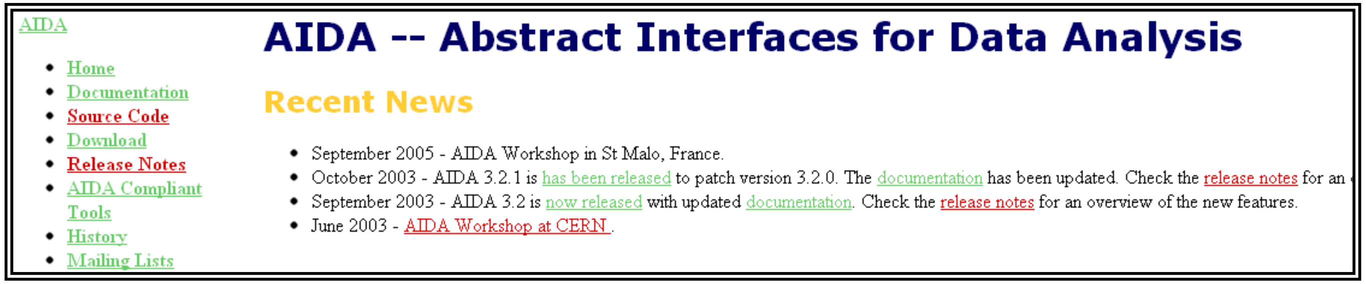
Visualization/GUI tools (optional):

- X11 OpenGL Visualization (Linux and Mac OS X)
  - Requires: X11, OpenGL or MesaGL (headers and libraries).
- Qt4 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)
- GDML Support (All Platforms)
- DAWN postscript renderer
- HepRApp Browser
- VRML browser
- WIRED4 JAS Plug-In

External software packages II

Software for analysis and histogramming (optional):

- AIDA (Abstract Interfaces for Data Analysis)
  - iAIDA (an implementation of AIDA in C++)
  - JAS3 (Java Analysis Studio)
  - Open Scientist (Interactive Analysis Environment)
  - rAIDA (a Root implementation of AIDA)



AIDA

- Home
- Documentation
- Source Code
- Download
- Release Notes
- AIDA Compliant Tools
- History
- Mailing Lists

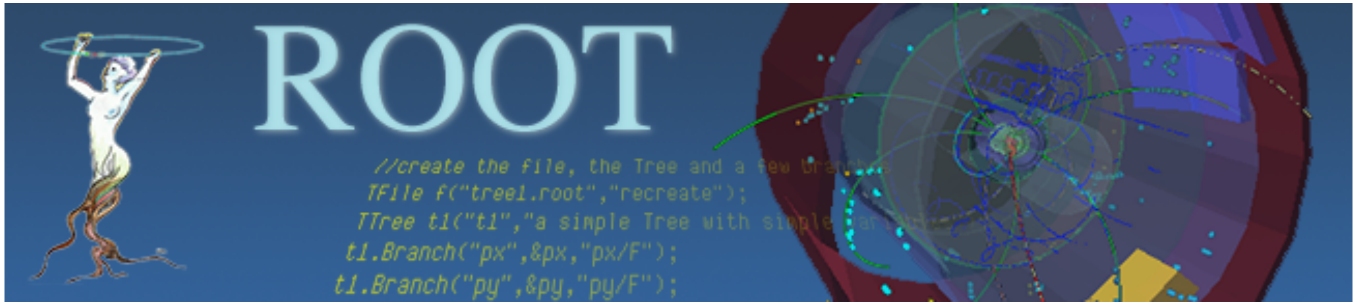
AIDA -- Abstract Interfaces for Data Analysis

Recent News

- September 2005 - AIDA Workshop in St Malo, France.
- October 2003 - AIDA 3.2.1 is has been released to patch version 3.2.0. The documentation has been updated. Check the release notes for an
- September 2003 - AIDA 3.2 is now released with updated documentation. Check the release notes for an overview of the new features.
- June 2003 - AIDA Workshop at CERN .

http://aida.freehep.org/

- ROOT (a data analysis framework)

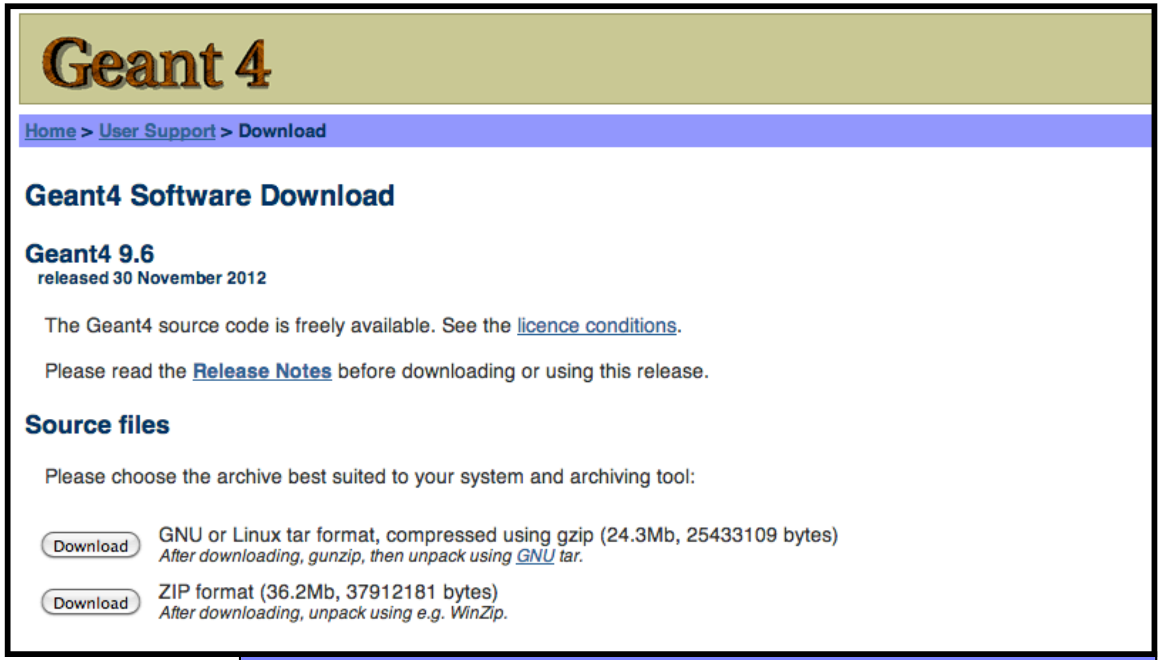


```
//create the file, the Tree and a few branches
TFile f("tree1.root","recreate");
TTree t1("t1","a simple Tree with simple vari
t1.Branch("px",&px,"px/F");
t1.Branch("py",&py,"py/F");
```

http://root.cern.ch/

Where to download the packages

- Geant4



Geant 4

Home > User Support > Download

Geant4 Software Download

Geant4 9.6

released 30 November 2012

The Geant4 source code is freely available. See the licence conditions.

Please read the **Release Notes** before downloading or using this release.

Source files

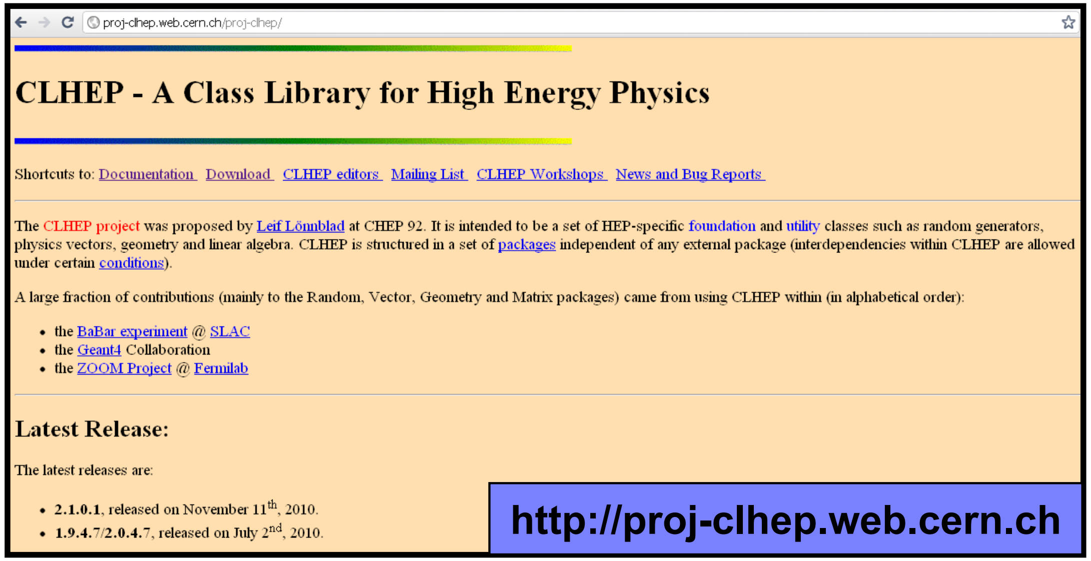
Please choose the archive best suited to your system and archiving tool:

Download — GNU or Linux tar format, compressed using gzip (24.3Mb, 25433109 bytes)
After downloading, gunzip, then unpack using GNU tar.

Download — ZIP format (36.2Mb, 37912181 bytes)
After downloading, unpack using e.g. WinZip.

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

- CLHEP



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

CLHEP - A Class Library for High Energy Physics

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

The CLHEP project was proposed by Leif Lönnblad at CHEP 92. It is intended to be a set of HEP-specific foundation and utility classes such as random generators, physics vectors, geometry and linear algebra. CLHEP is structured in a set of packages independent of any external package (interdependencies within CLHEP are allowed under certain conditions).

A large fraction of contributions (mainly to the Random, Vector, Geometry and Matrix packages) came from using CLHEP within (in alphabetical order):

- the BaBar experiment @ SLAC
- the Geant4 Collaboration
- the ZOOM Project @ Fermilab

Latest Release:

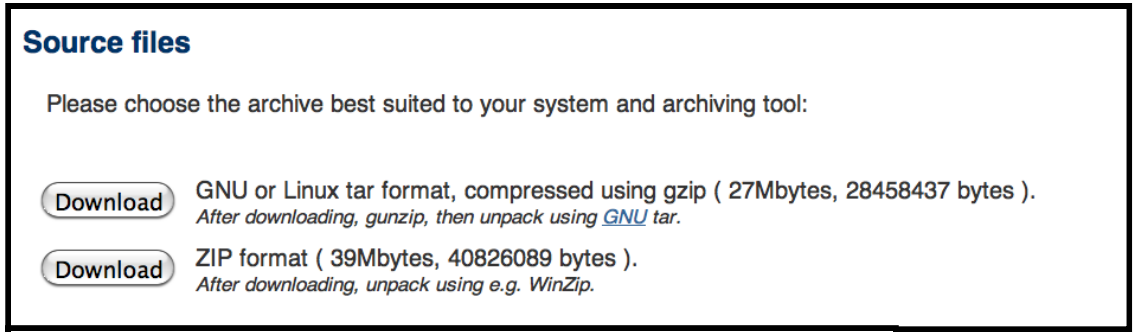
The latest releases are:

- **2.1.0.1**, released on November 11th, 2010.
- **1.9.4.7/2.0.4.7**, released on July 2nd, 2010.

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

Downloading Geant4 and data files

Geant4 source
or
pre-compiled libraries

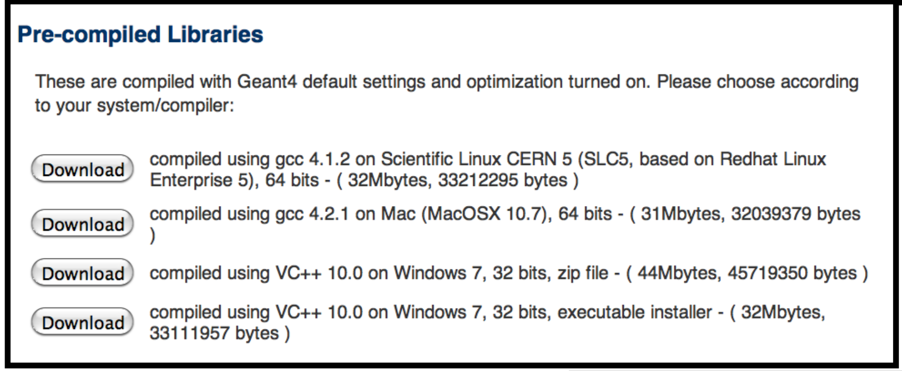


Source files

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

Download GNU or Linux tar format, compressed using gzip (27Mbytes, 28458437 bytes).
After downloading, gunzip, then unpack using GNU tar.

Download ZIP format (39Mbytes, 40826089 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:

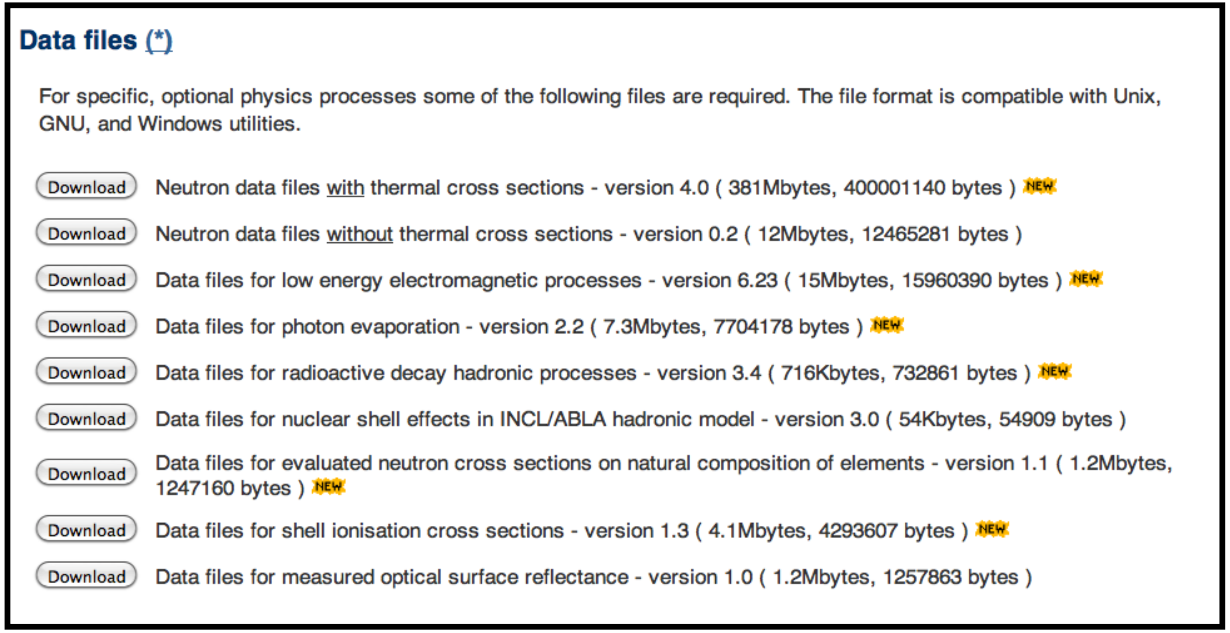
Download compiled using gcc 4.1.2 on Scientific Linux CERN 5 (SLC5, based on Redhat Linux Enterprise 5), 64 bits - (32Mbytes, 33212295 bytes)

Download compiled using gcc 4.2.1 on Mac (MacOSX 10.7), 64 bits - (31Mbytes, 32039379 bytes)

Download compiled using VC++ 10.0 on Windows 7, 32 bits, zip file - (44Mbytes, 45719350 bytes)

Download compiled using VC++ 10.0 on Windows 7, 32 bits, executable installer - (32Mbytes, 33111957 bytes)

data files



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 4.0 (381Mbytes, 400001140 bytes) NEW

Download Neutron data files without thermal cross sections - version 0.2 (12Mbytes, 12465281 bytes)

Download Data files for low energy electromagnetic processes - version 6.23 (15Mbytes, 15960390 bytes) NEW

Download Data files for photon evaporation - version 2.2 (7.3Mbytes, 7704178 bytes) NEW

Download Data files for radioactive decay hadronic processes - version 3.4 (716Kbytes, 732861 bytes) NEW

Download Data files for nuclear shell effects in INCL/ABLA hadronic model - version 3.0 (54Kbytes, 54909 bytes)

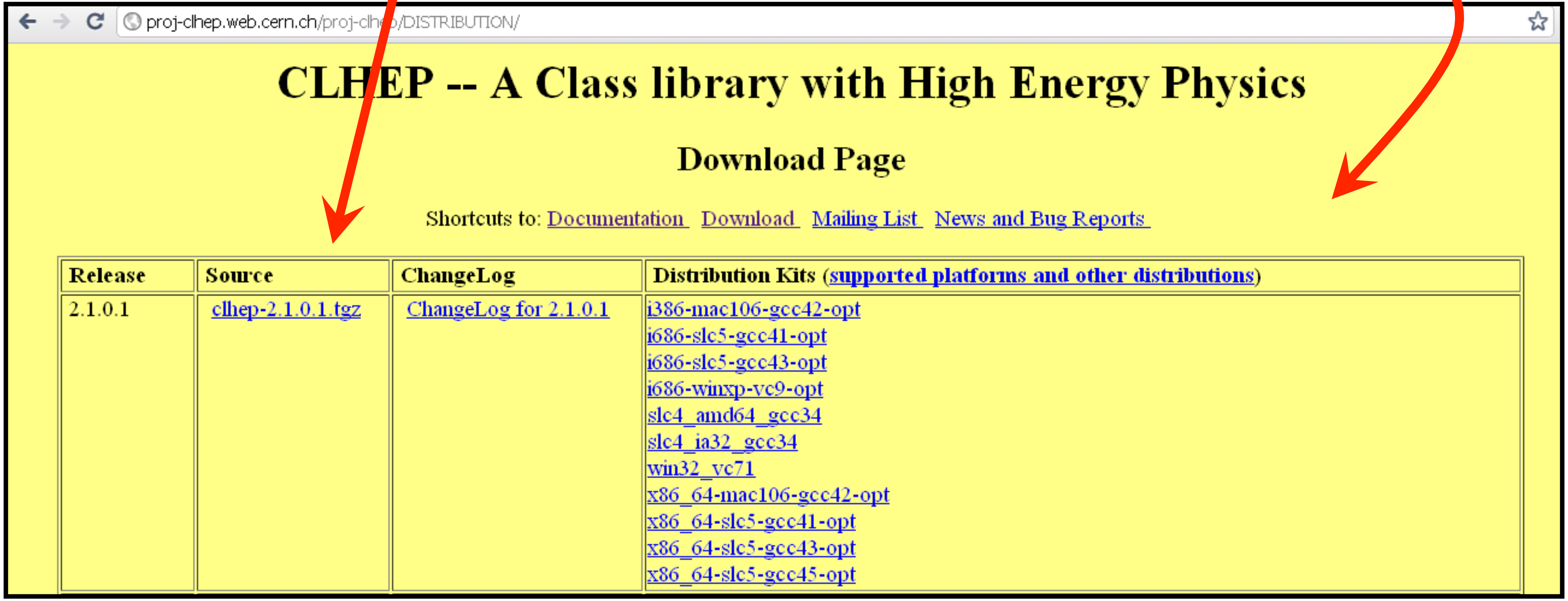
Download Data files for evaluated neutron cross sections on natural composition of elements - version 1.1 (1.2Mbytes, 1247160 bytes) NEW

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

Download Data files for measured optical surface reflectance - version 1.0 (1.2Mbytes, 1257863 bytes)

Downloading CLHEP (optionally)

Source code *or* ***pre-compiled libraries***



CLHEP -- A Class library with High Energy Physics

Download Page

Shortcuts to: Documentation Download Mailing List News and Bug Reports

Release	Source	ChangeLog	Distribution Kits (supported platforms and other distributions)
2.1.0.1	clhep-2.1.0.1.tgz	ChangeLog for 2.1.0.1	i386-mac106-gcc42-opt i686-slc5-gcc41-opt i686-slc5-gcc43-opt i686-winxp-vc9-opt slc4_amd64_gcc34 slc4_ia32_gcc34 win32_vc71 x86_64-mac106-gcc42-opt x86_64-slc5-gcc41-opt x86_64-slc5-gcc43-opt x86_64-slc5-gcc45-opt

Geant4 installation (9.6 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 geant4.9.6.tar.gz to a location of your choice:
  - ex.: /path/to/geant4.9.6 → *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/geant4.9.6-build → *build directory*

```
$ cd /path/to
$ mkdir geant4.9.6-build
$ ls
geant4.9.6  geant4.9.6-build
```

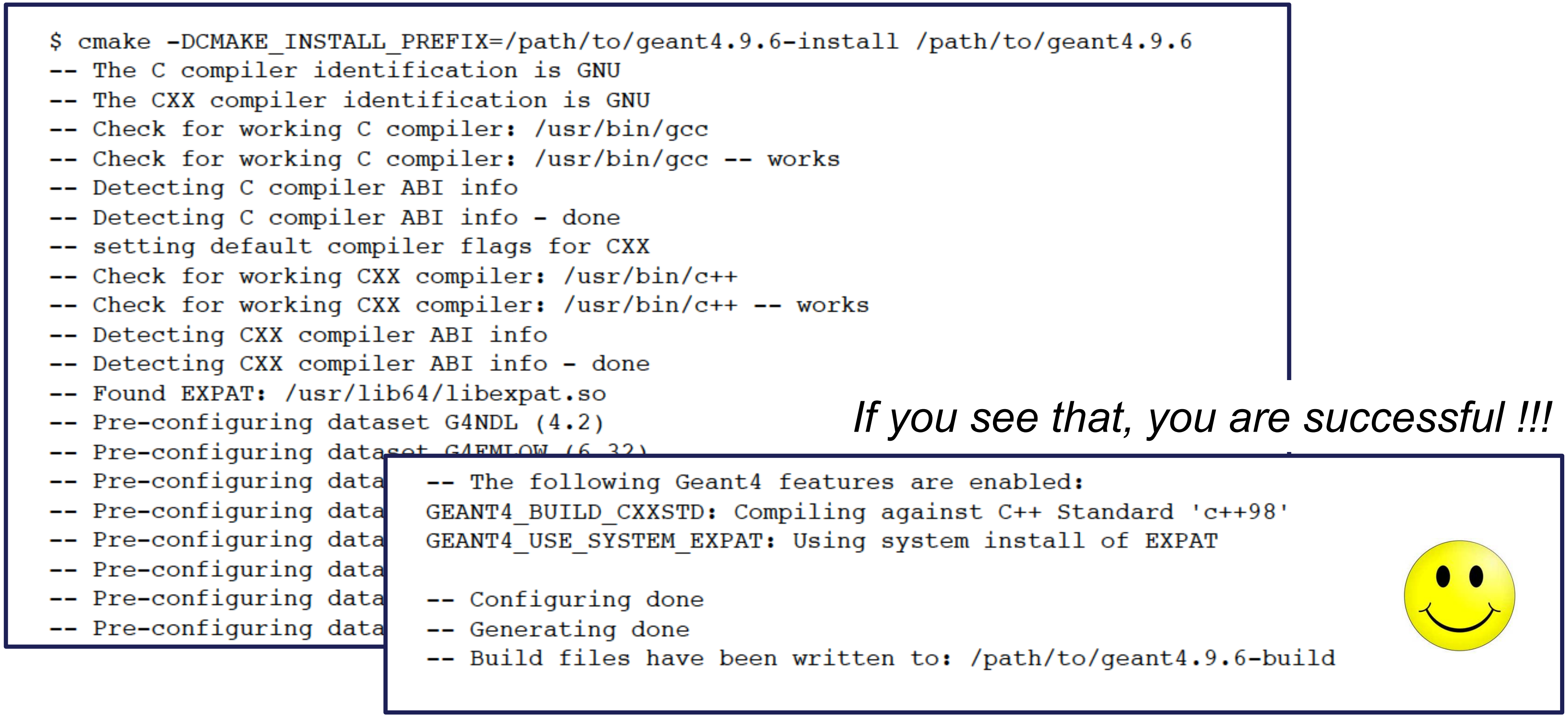
- To configure, change into the build directory and run CMake:

```
$ cd /path/to/geant4.9.6-build
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.9.6-install /path/to/geant4.9.6
```

  - 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/geant4.9.6-install /path/to/geant4.9.6
-- 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
-- Pre-configuring dataset G4NDL (4.2)
-- Pre-configuring dataset G4EMLOW (6.32)
-- Pre-configuring data
-- Pre-configuring data
-- Pre-configuring data
-- Pre-configuring data
-- Pre-configuring data
-- Pre-configuring data
```

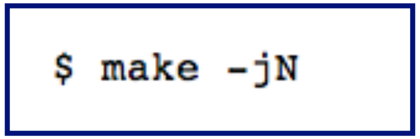
If you see that, you are successful !!!

```
-- The following Geant4 features are enabled:
GEANT4_BUILD_CXXSTD: Compiling against C++ Standard 'c++98'
GEANT4_USE_SYSTEM_EXPAT: Using system install of EXPAT

-- Configuring done
-- Generating done
-- Build files have been written to: /path/to/geant4.9.6-build
```

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

Geant4 installation with CMake

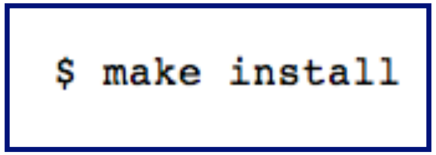
- After the configuration has run, CMake have generated Unix Makefiles for building Geant4. To run the build, simply execute make in the build directory:



```
$ 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

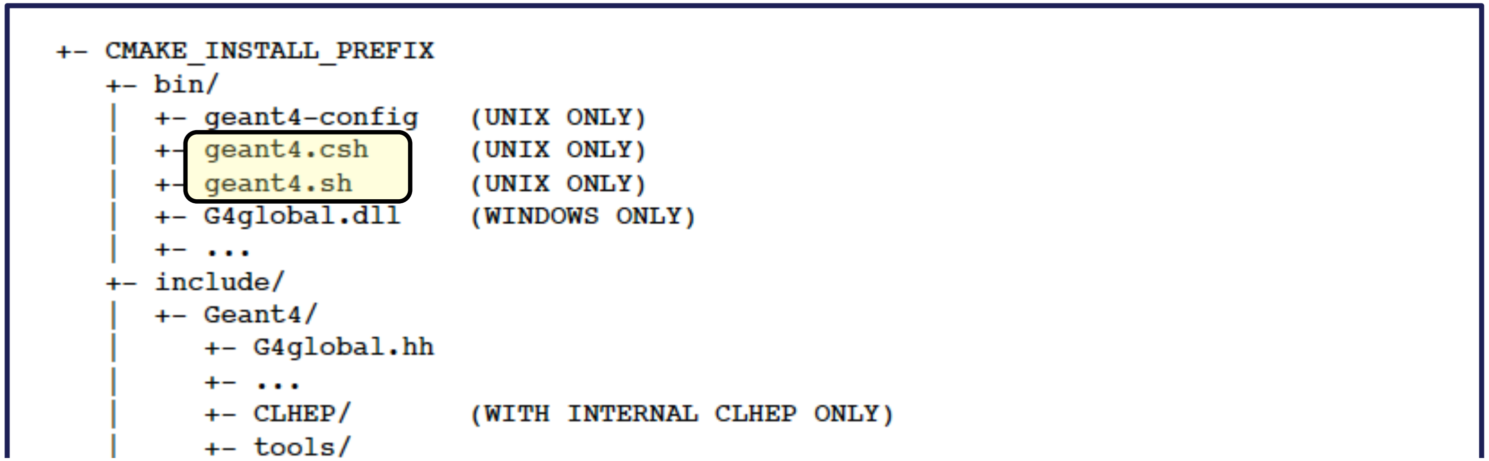
- Additional arguments can be passed to CMake to activate optional components of Geant4 (*standard* and *advanced* options):
  - **-DGEANT4_INSTALL_DATA=ON** *(recommended)*
    the additional external data libraries are automatically downloaded
  - **-DGEANT4_INSTALL_EXAMPLES=ON** *(recommended)*
    examples are installed
  - **-DGEANT4_USE_OPENGL_X11=ON** *(recommended)*
    build the X11 OpenGL visualization driver
  - -DGEANT4_USE_QT=ON *(optional, but nice!!!)*
    build the Qt visualization driver
  - -DGEANT4_USE_SYSTEM_CLHEP=ON *(optional)*
    external CLHEP are required

You can directly include the options since the beginning:

```
cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.9.6-install  -DGEANT4_INSTALL_DATA=ON
-DGEANT4_USE_OPENGL_X11=ON -DGEANT4_INSTALL_EXAMPLES=ON /path/to/geant4.9.6
```

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/
```

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

```
$ . geant4.sh
```

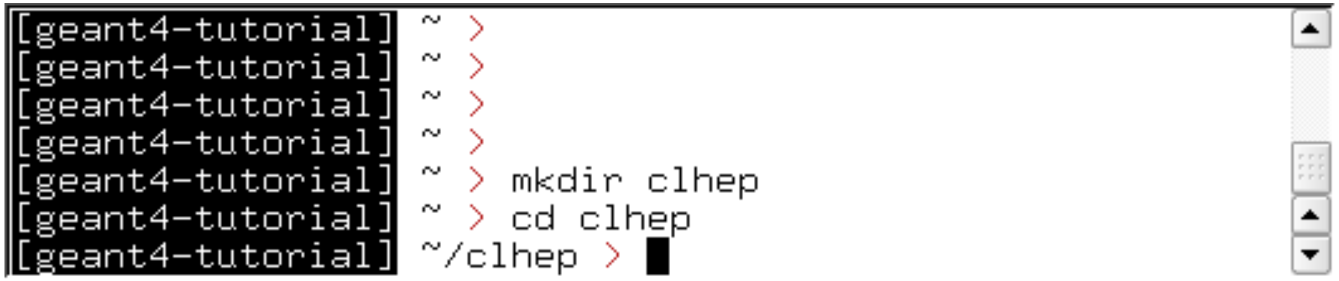
N.B.: each time you close the shell remember to source the `geant4.sh` *script before executing an application !!!*

- Alternatively, you may use the *geant4make.sh* (*.csh*) script to compile applications with GNUmakefile (*deprecated → G4.10*)

Installing CLHEP full version

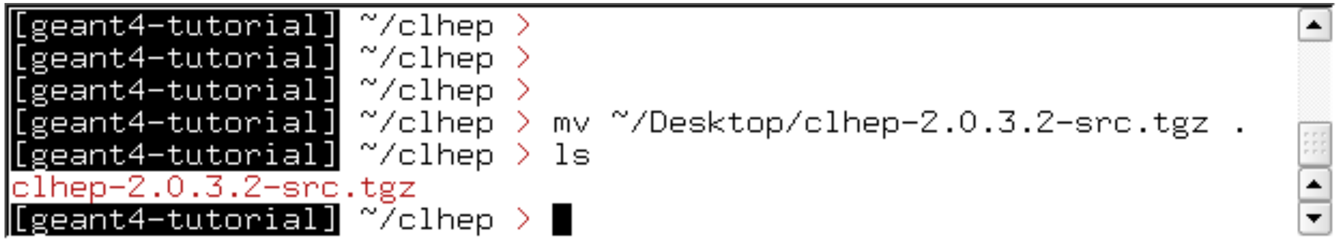
(not mandatory)

- Create a directory for the installation procedure (ex.:clhep)



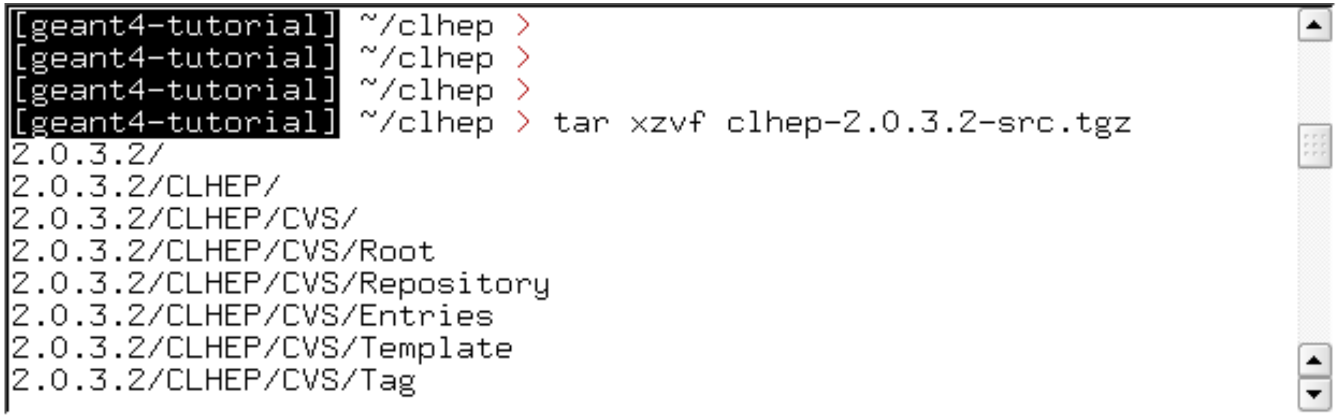
```
[geant4-tutorial] ~ >
[geant4-tutorial] ~ >
[geant4-tutorial] ~ >
[geant4-tutorial] ~ >
[geant4-tutorial] ~ > mkdir clhep
[geant4-tutorial] ~ > cd clhep
[geant4-tutorial] ~/clhep > █
```

- Move the downloaded tar-ball into this directory



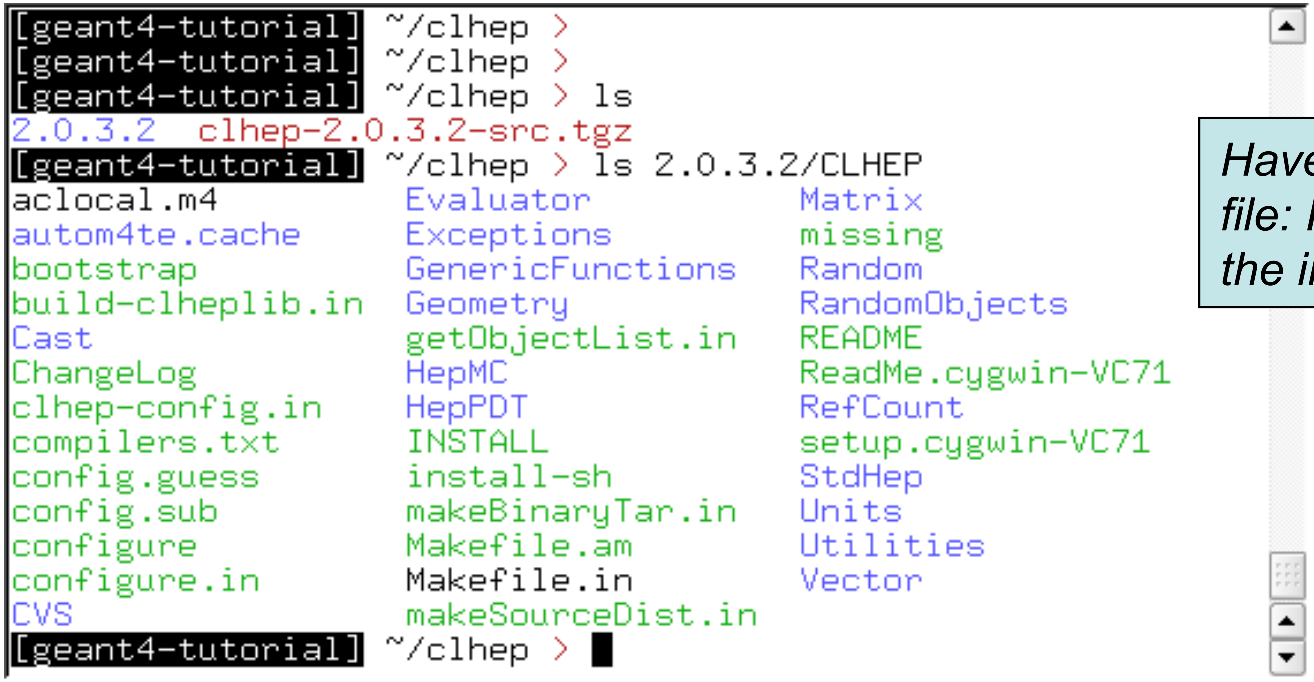
```
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep > mv ~/Desktop/clhep-2.0.3.2-src.tgz .
[geant4-tutorial] ~/clhep > ls
clhep-2.0.3.2-src.tgz
[geant4-tutorial] ~/clhep > █
```

- Unzip the extract tar-ball into this directory



```
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep > tar xzvf clhep-2.0.3.2-src.tgz
2.0.3.2/
2.0.3.2/CLHEP/
2.0.3.2/CLHEP/CVS/
2.0.3.2/CLHEP/CVS/Root
2.0.3.2/CLHEP/CVS/Repository
2.0.3.2/CLHEP/CVS/Entries
2.0.3.2/CLHEP/CVS/Template
2.0.3.2/CLHEP/CVS/Tag
```

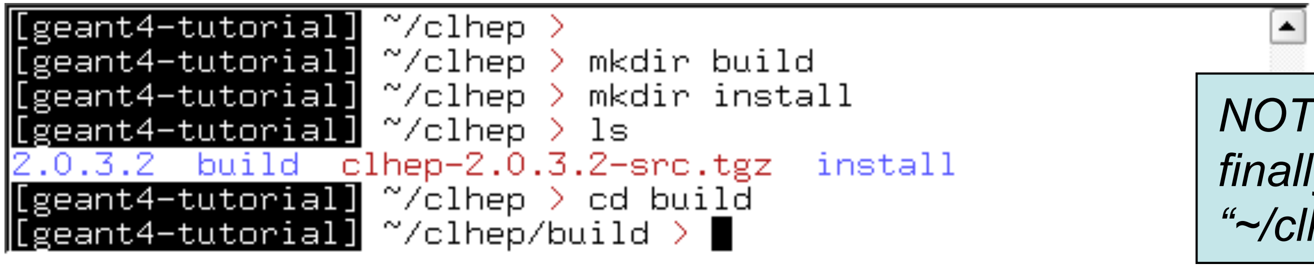
- The extracted CLHEP package can be found in the subdirectory 2.0.3.2/CLHEP”. Have a look at the content:



```
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep > ls
2.0.3.2  clhep-2.0.3.2-src.tgz
[geant4-tutorial] ~/clhep > ls 2.0.3.2/CLHEP
aclocal.m4          Evaluator           Matrix
autom4te.cache      Exceptions          missing
bootstrap           GenericFunctions    Random
build-clheplib.in   Geometry            RandomObjects
Cast                getObjectList.in    README
ChangeLog           HepMC               ReadMe.cygwin-VC71
clhep-config.in     HepPDT              RefCount
compilers.txt       INSTALL             setup.cygwin-VC71
config.guess        install-sh          StdHep
config.sub          makeBinaryTar.in    Units
configure           Makefile.am         Utilities
configure.in        Makefile.in         Vector
CVS                 makeSourceDist.in
[geant4-tutorial] ~/clhep > █
```

Have a look in the “INSTALL” file: It contains more details on the installation procedure

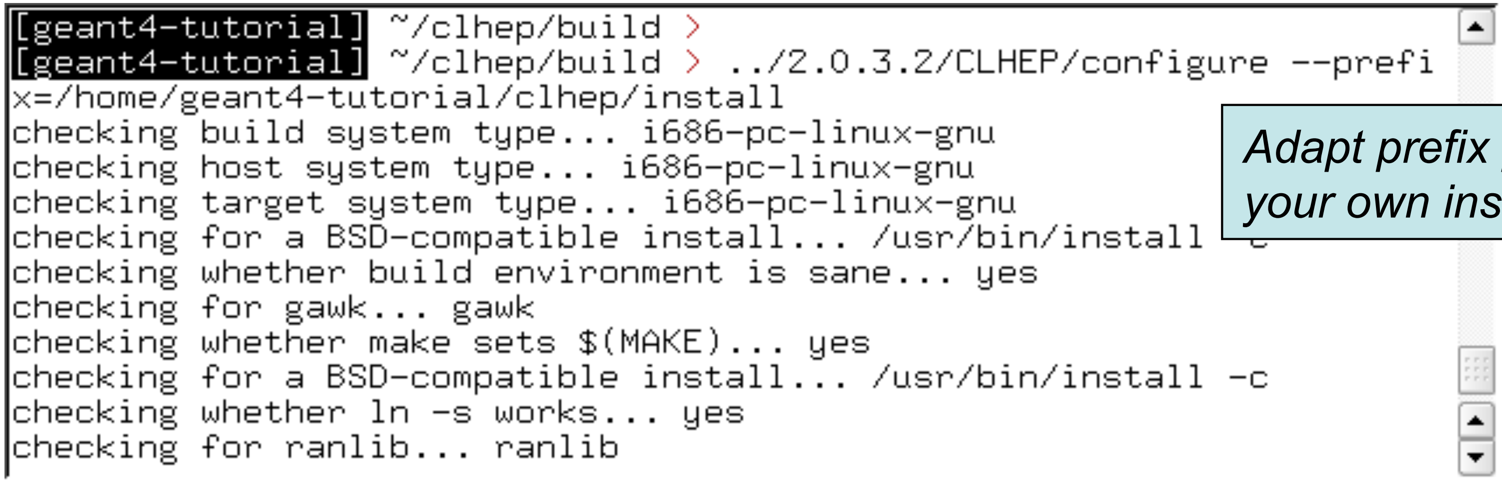
- Create two directories (inside our “clhep” directory), which are used for building and installing the package:



```
[geant4-tutorial] ~/clhep >
[geant4-tutorial] ~/clhep > mkdir build
[geant4-tutorial] ~/clhep > mkdir install
[geant4-tutorial] ~/clhep > ls
2.0.3.2  build  clhep-2.0.3.2-src.tgz  install
[geant4-tutorial] ~/clhep > cd build
[geant4-tutorial] ~/clhep/build > █
```

NOTE: The package will be finally installed in the directory “~/clhep/install”

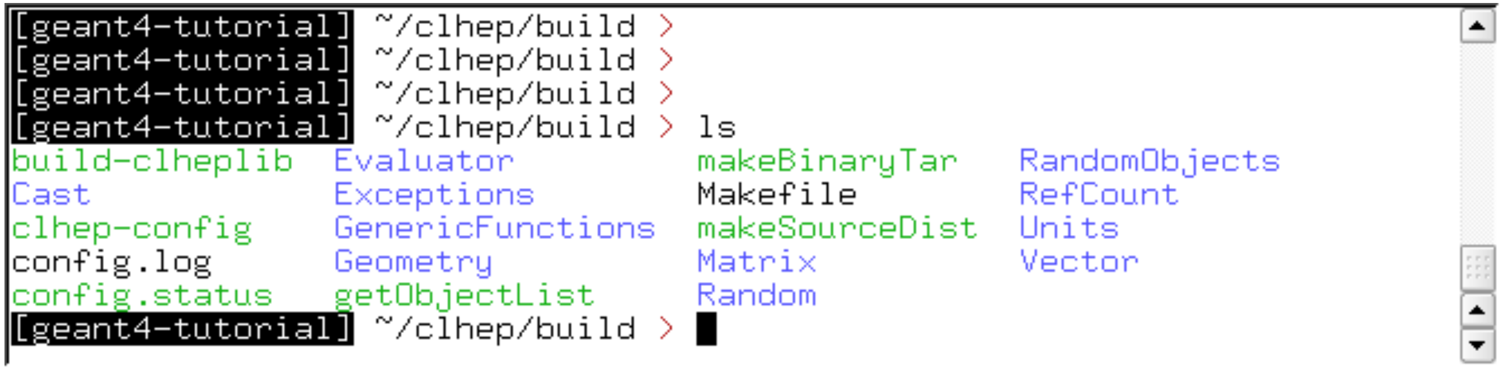
- Inside the “build” directory, call the CLHEP configure script (which is contained in the “2.0.3.2/CLHEP” directory).
  NOTE: As argument you need to specify the directory, where CLHEP should be installed. Thus the full command to be called is: ../2.0.3.2/CLHEP/configure --prefix=/home/geant4-tutorial/clhep/install



```
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build > ../2.0.3.2/CLHEP/configure --prefi
x=/home/geant4-tutorial/clhep/install
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for a BSD-compatible install... /usr/bin/install -c
checking whether ln -s works... yes
checking for ranlib... ranlib
```

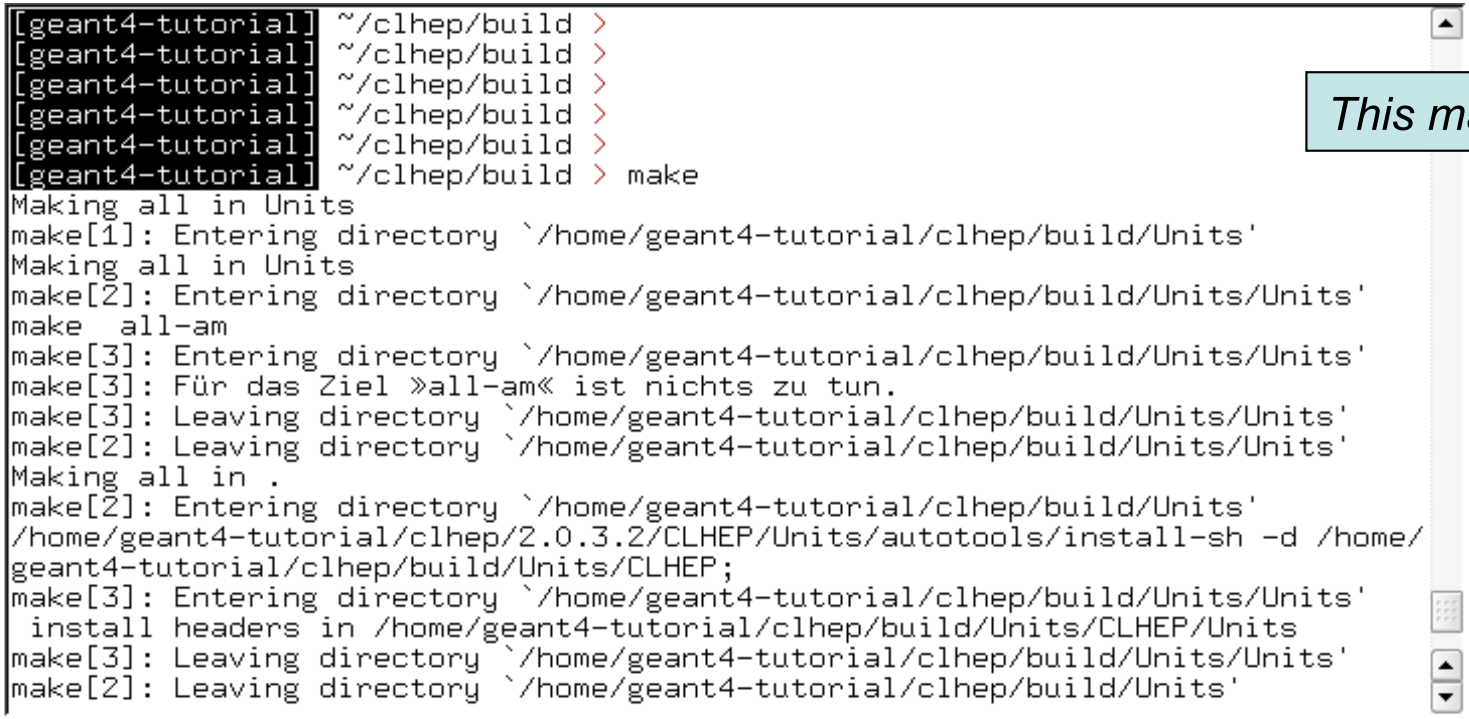
Adapt prefix path according to your own installation directory!

- The configure script checks for required programs and libraries,and creates some files, e.g. makefiles, and directories:



```
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build > ls
build-clheplib  Evaluator         makeBinaryTar   RandomObjects
Cast            Exceptions        Makefile        RefCount
clhep-config    GenericFunctions  makeSourceDist  Units
config.log      Geometry          Matrix          Vector
config.status   getObjectList     Random
[geant4-tutorial] ~/clhep/build > █
```

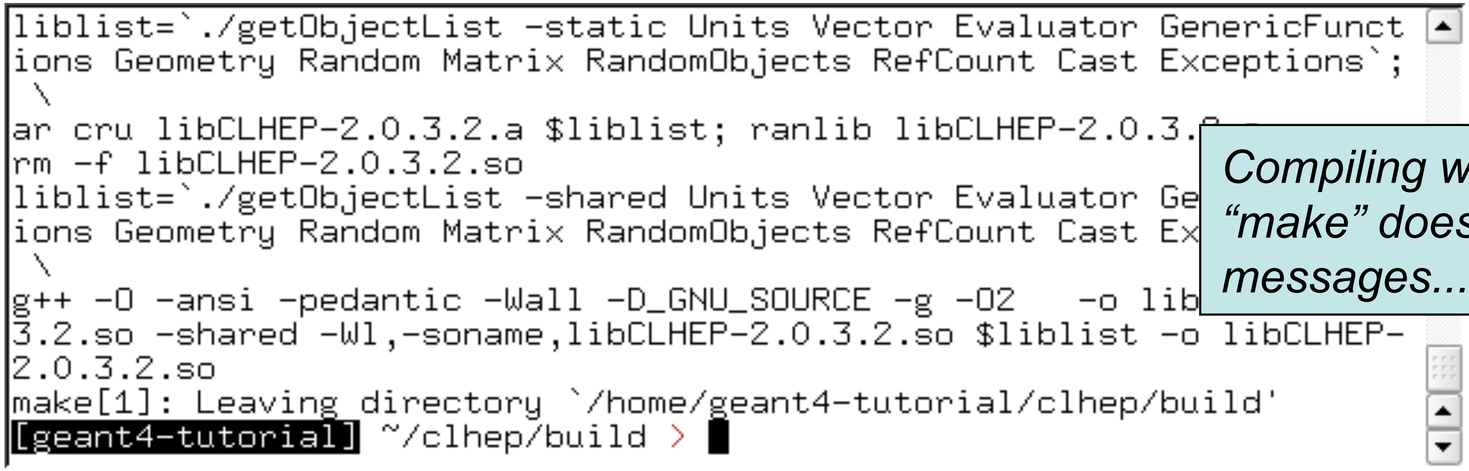
- If no error occured in the configure process, one can start to build the CLHEP package using the “make” command:



```
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build > make
Making all in Units
make[1]: Entering directory `/home/geant4-tutorial/clhep/build/Units'
Making all in Units
make[2]: Entering directory `/home/geant4-tutorial/clhep/build/Units/Units'
make  all-am
make[3]: Entering directory `/home/geant4-tutorial/clhep/build/Units/Units'
make[3]: Für das Ziel »all-am« ist nichts zu tun.
make[3]: Leaving directory `/home/geant4-tutorial/clhep/build/Units/Units'
make[2]: Leaving directory `/home/geant4-tutorial/clhep/build/Units/Units'
Making all in .
make[2]: Entering directory `/home/geant4-tutorial/clhep/build/Units'
/home/geant4-tutorial/clhep/2.0.3.2/CLHEP/Units/autotools/install-sh -d /home/
geant4-tutorial/clhep/build/Units/CLHEP;
make[3]: Entering directory `/home/geant4-tutorial/clhep/build/Units/Units'
 install headers in /home/geant4-tutorial/clhep/build/Units/CLHEP/Units
make[3]: Leaving directory `/home/geant4-tutorial/clhep/build/Units/Units'
make[2]: Leaving directory `/home/geant4-tutorial/clhep/build/Units'
```

This may take a while...

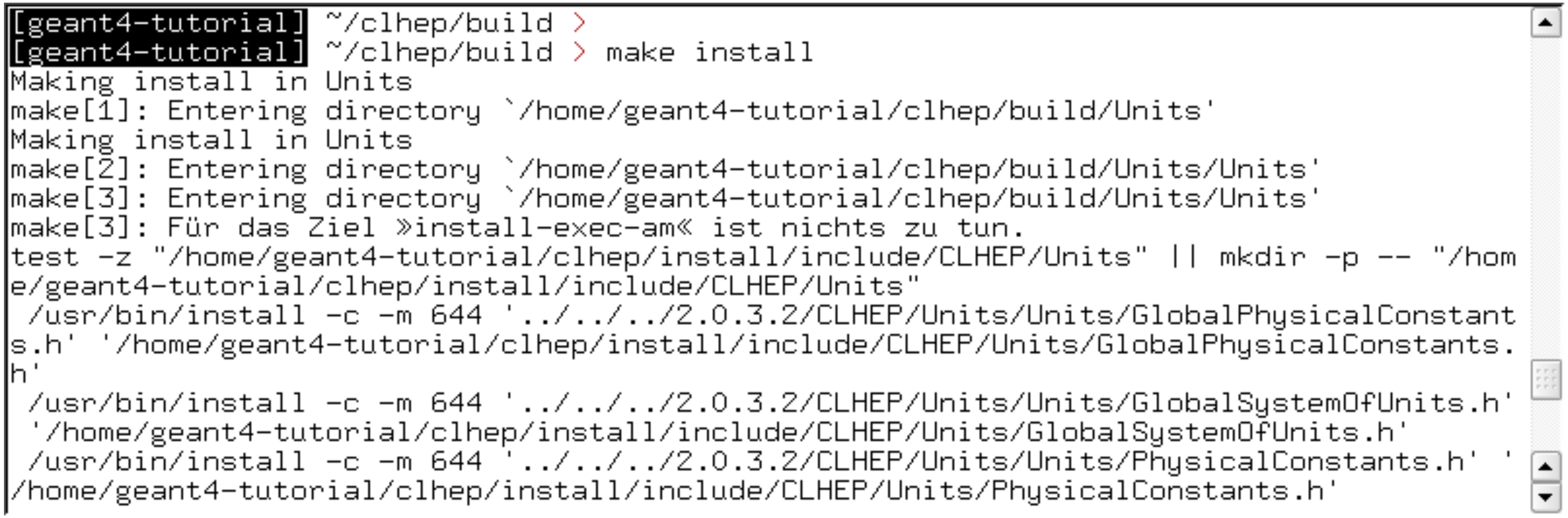
Only the initial and last output messages of the make command are shown



```
liblist=`./getObjectList -static Units Vector Evaluator GenericFunct
ions Geometry Random Matrix RandomObjects RefCount Cast Exceptions`;
 \
ar cru libCLHEP-2.0.3.2.a $liblist; ranlib libCLHEP-2.0.3.
rm -f libCLHEP-2.0.3.2.so
liblist=`./getObjectList -shared Units Vector Evaluator Ge
ions Geometry Random Matrix RandomObjects RefCount Cast Ex
 \
g++ -O -ansi -pedantic -Wall -D_GNU_SOURCE -g -O2    -o lib
3.2.so -shared -Wl,-soname,libCLHEP-2.0.3.2.so $liblist -o libCLHEP-
2.0.3.2.so
make[1]: Leaving directory `/home/geant4-tutorial/clhep/build'
[geant4-tutorial] ~/clhep/build > █
```

Compiling was successful if “make” does not exit with error messages...

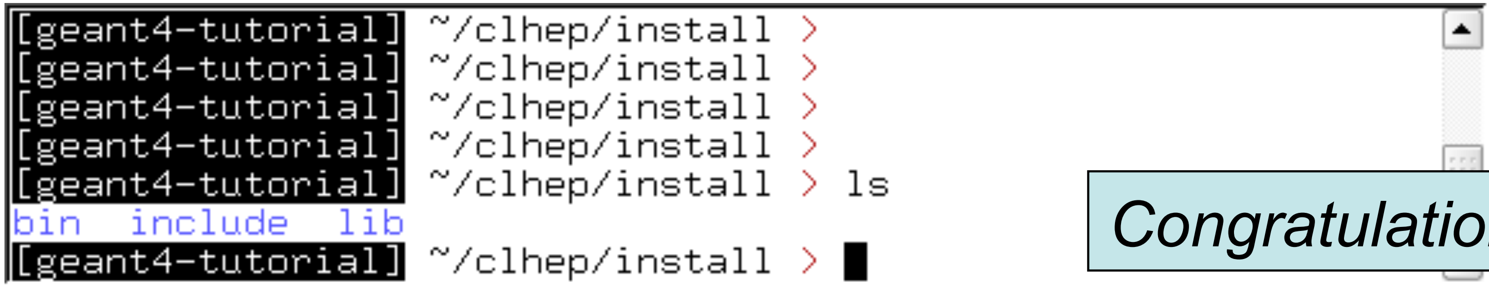
- Once the package was compiled successfully, CLHEP can be installed using the “make install” command:



```
[geant4-tutorial] ~/clhep/build >
[geant4-tutorial] ~/clhep/build > make install
Making install in Units
make[1]: Entering directory `/home/geant4-tutorial/clhep/build/Units'
Making install in Units
make[2]: Entering directory `/home/geant4-tutorial/clhep/build/Units/Units'
make[3]: Entering directory `/home/geant4-tutorial/clhep/build/Units/Units'
make[3]: Für das Ziel »install-exec-am« ist nichts zu tun.
test -z "/home/geant4-tutorial/clhep/install/include/CLHEP/Units" || mkdir -p -- "/hom
e/geant4-tutorial/clhep/install/include/CLHEP/Units"
 /usr/bin/install -c -m 644 '../../../2.0.3.2/CLHEP/Units/Units/GlobalPhysicalConstant
s.h' '/home/geant4-tutorial/clhep/install/include/CLHEP/Units/GlobalPhysicalConstants.
h'
 /usr/bin/install -c -m 644 '../../../2.0.3.2/CLHEP/Units/Units/GlobalSystemOfUnits.h'
 '/home/geant4-tutorial/clhep/install/include/CLHEP/Units/GlobalSystemOfUnits.h'
 /usr/bin/install -c -m 644 '../../../2.0.3.2/CLHEP/Units/Units/PhysicalConstants.h' '
/home/geant4-tutorial/clhep/install/include/CLHEP/Units/PhysicalConstants.h'
```

- The CLHEP libraries are now installed in the directory “~/clhep/install”

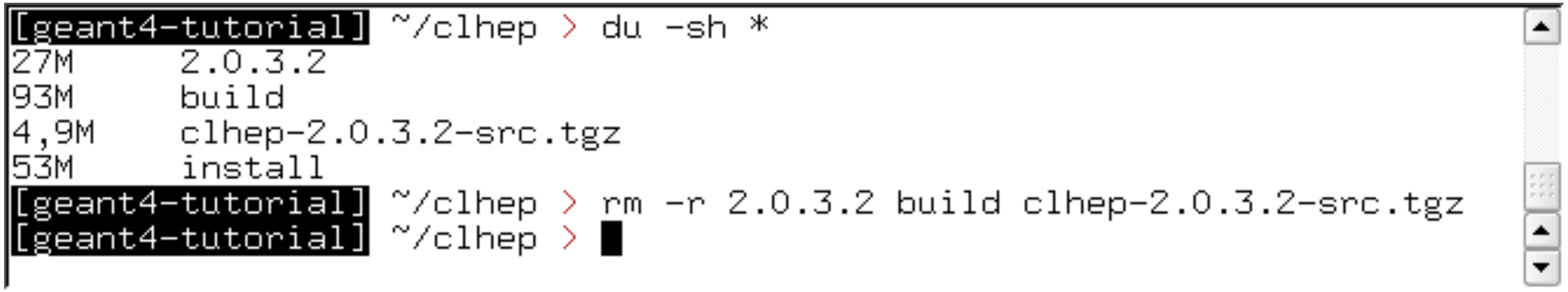
(NOTE: We specified the installation directory in the configure process; see the previous slides)



```
[geant4-tutorial] ~/clhep/install >
[geant4-tutorial] ~/clhep/install >
[geant4-tutorial] ~/clhep/install >
[geant4-tutorial] ~/clhep/install >
[geant4-tutorial] ~/clhep/install > ls
bin  include  lib
[geant4-tutorial] ~/clhep/install > █
```

Congratulations!

- What do the subdirectories in “~/clhep/install” contain?
  - include: Contains (in a defined directory tree structure) the C++ header files of CLHEP
  - lib: Contains the (static and shared) CLHEP libraries
  - bin: Contains configure scripts and the very useful “clhep- config” script
- Finally, to save some disk space, you can remove the “build” directory, as well as the tar-ball and the source package



```
[geant4-tutorial] ~/clhep > du -sh *
27M     2.0.3.2
93M     build
4,9M    clhep-2.0.3.2-src.tgz
53M     install
[geant4-tutorial] ~/clhep > rm -r 2.0.3.2 build clhep-2.0.3.2-src.tgz
[geant4-tutorial] ~/clhep > █
```

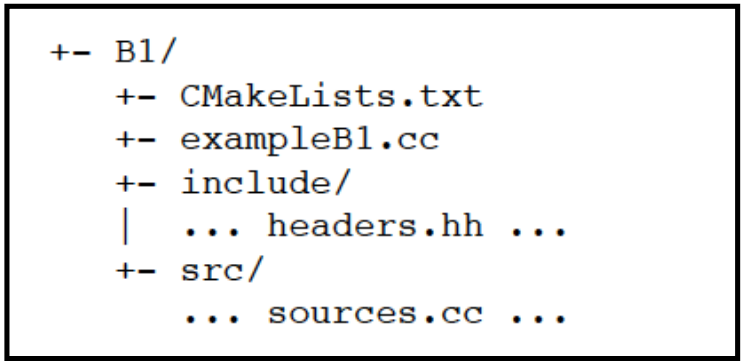
Building an application with CMake

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

  to locate Geant4 and describe the build of your application against it

- For istance: examples/basic/B1:

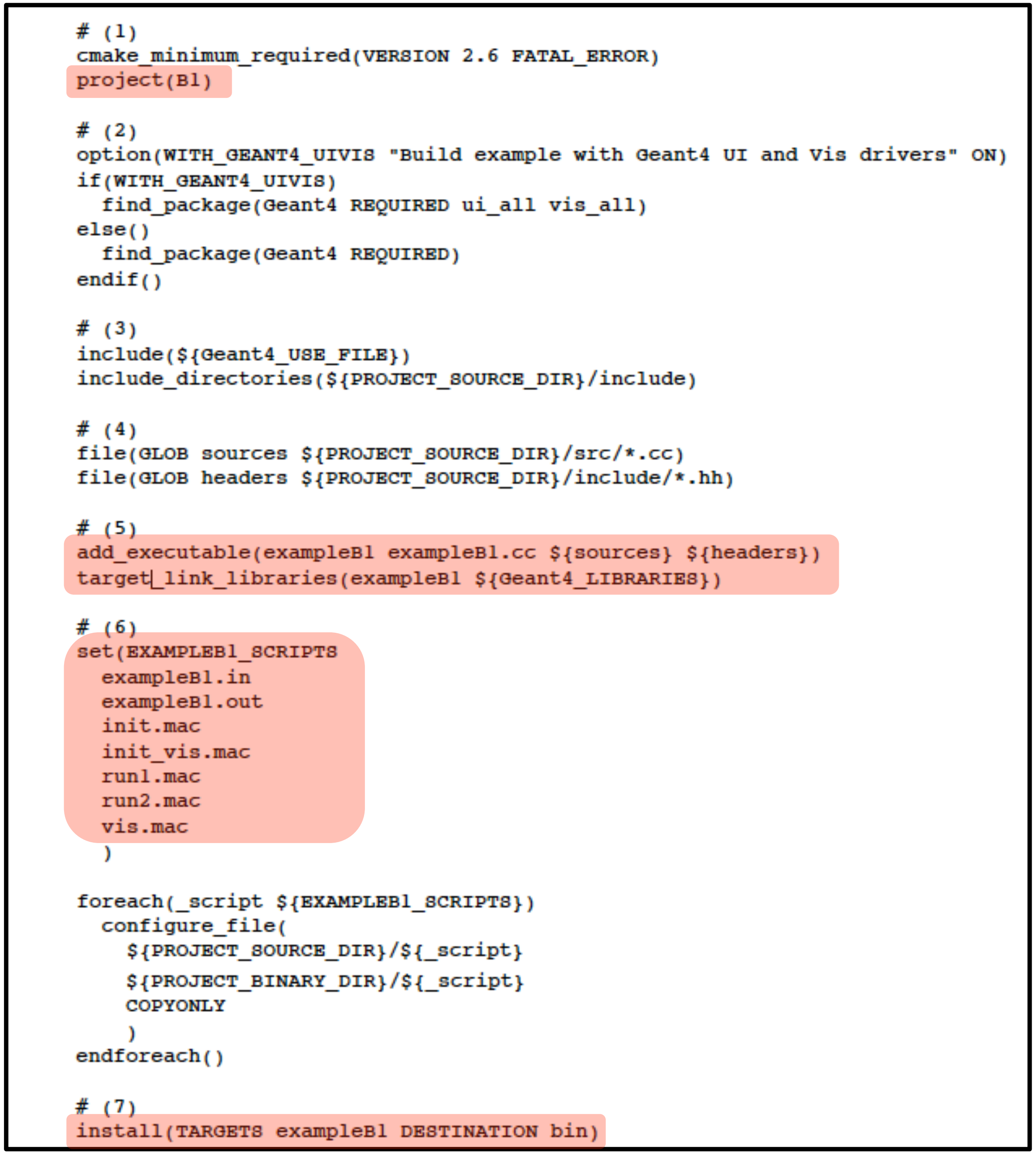


```
+- B1/
   +- CMakeLists.txt
   +- exampleB1.cc
   +- include/
   |   ... headers.hh ...
   +- src/
       ... sources.cc ...
```

Here, exampleB1.cc contains main() for the application, with include/ and src/ containing the implementation class headers and sources respectively.

CMakeLists.txt file has to be located in the root directory of the application.

Building an application with cmake



```
# (1)
cmake_minimum_required(VERSION 2.6 FATAL_ERROR)
project(B1)

# (2)
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()

# (3)
include(${Geant4_USE_FILE})
include_directories(${PROJECT_SOURCE_DIR}/include)

# (4)
file(GLOB sources ${PROJECT_SOURCE_DIR}/src/*.cc)
file(GLOB headers ${PROJECT_SOURCE_DIR}/include/*.hh)

# (5)
add_executable(exampleB1 exampleB1.cc ${sources} ${headers})
target_link_libraries(exampleB1 ${Geant4_LIBRARIES})

# (6)
set(EXAMPLEB1_SCRIPTS
  exampleB1.in
  exampleB1.out
  init.mac
  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()

# (7)
install(TARGETS exampleB1 DESTINATION bin)
```

- The text file CMakeLists.txt is the CMake script containing commands which describe how to build the exampleB1 application
- Example of 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 build directory for the specific application (suggestion: build that alongside the application source directory):

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

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

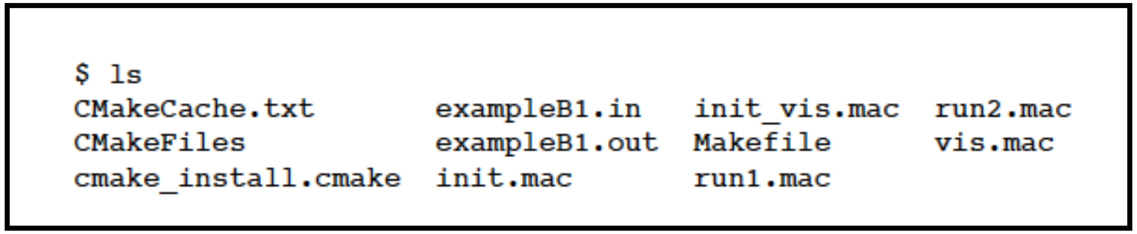
```
$ cd $HOME/B1-build
$ cmake -DGeant4_DIR=/home/you/geant4-install/lib64/Geant4-9.6.0 $HOME/B1
```

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

```
$ cmake -DGeant4_DIR=/home/you/geant4-install/lib64/Geant4-9.6.0 $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
```

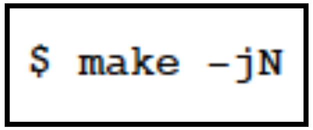
Building an application with cmake

- The following files have been generated:



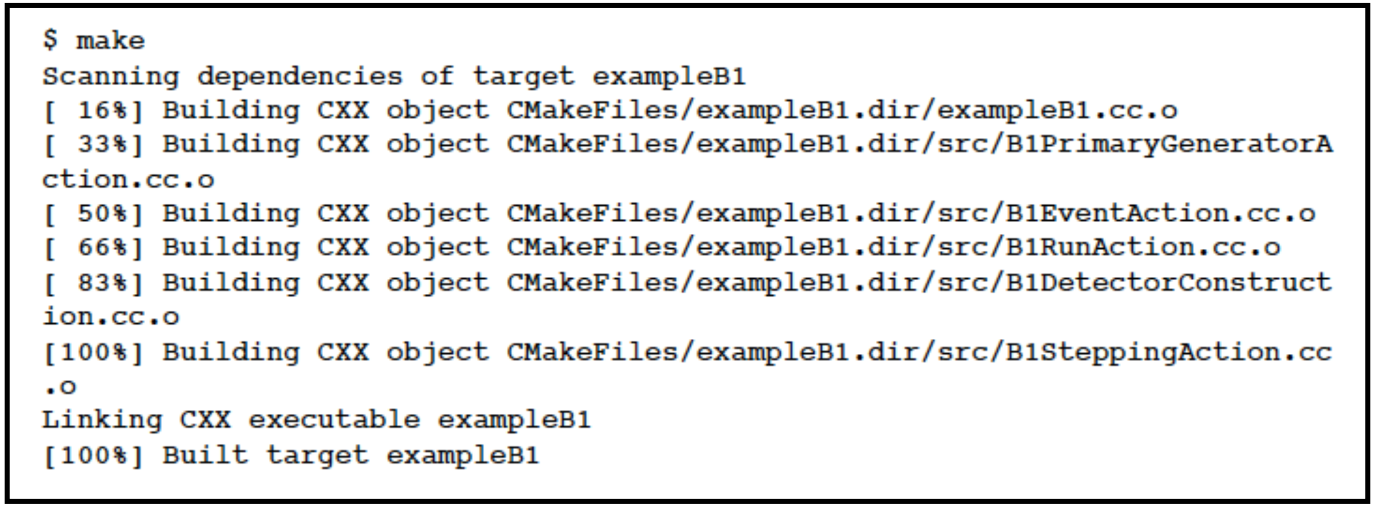
```
$ ls
CMakeCache.txt       exampleB1.in   init_vis.mac  run2.mac
CMakeFiles           exampleB1.out  Makefile      vis.mac
cmake_install.cmake  init.mac       run1.mac
```

- Once the Makefile is available we can do:



```
$ make -jN
```

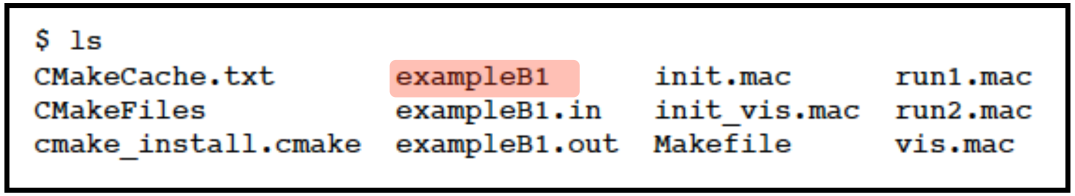
- The following output should be displayed:



```
$ 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
```

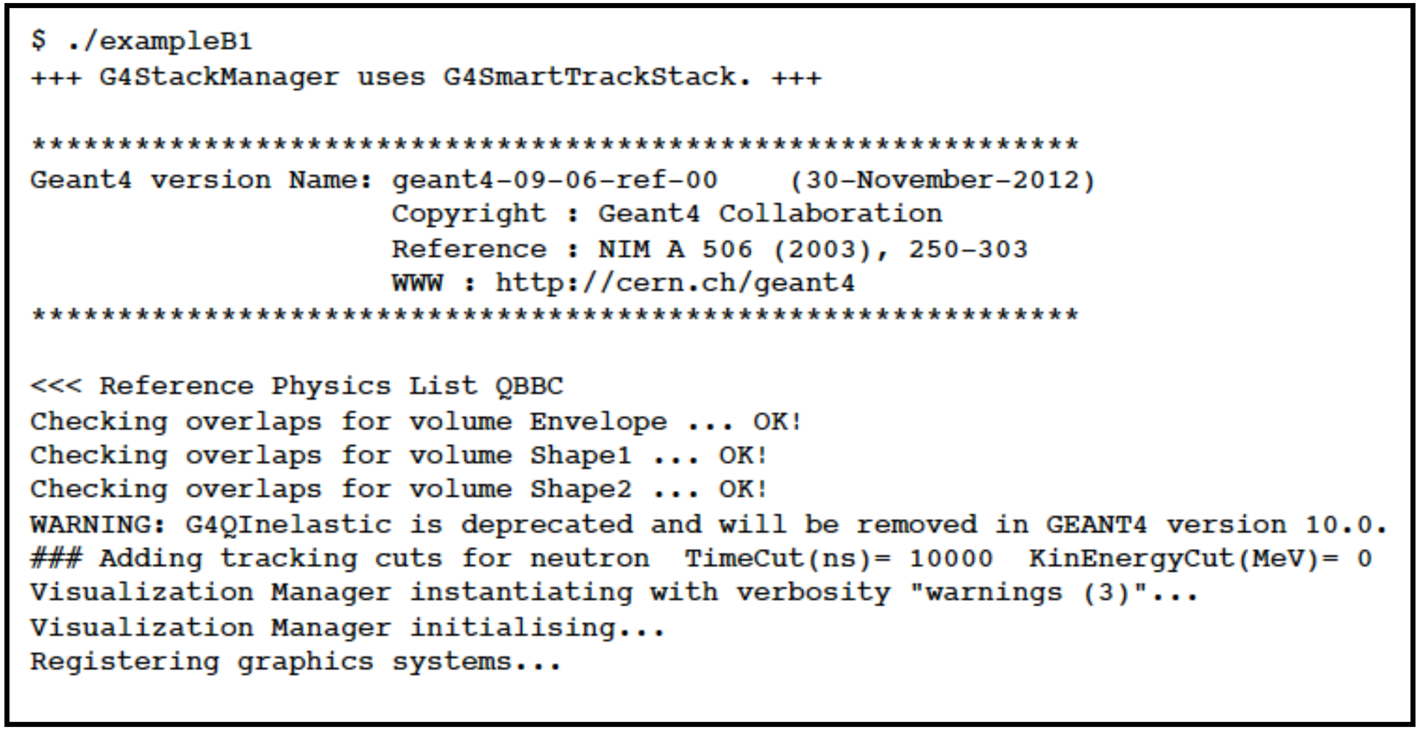
Building an application with cmake

- 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
```

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

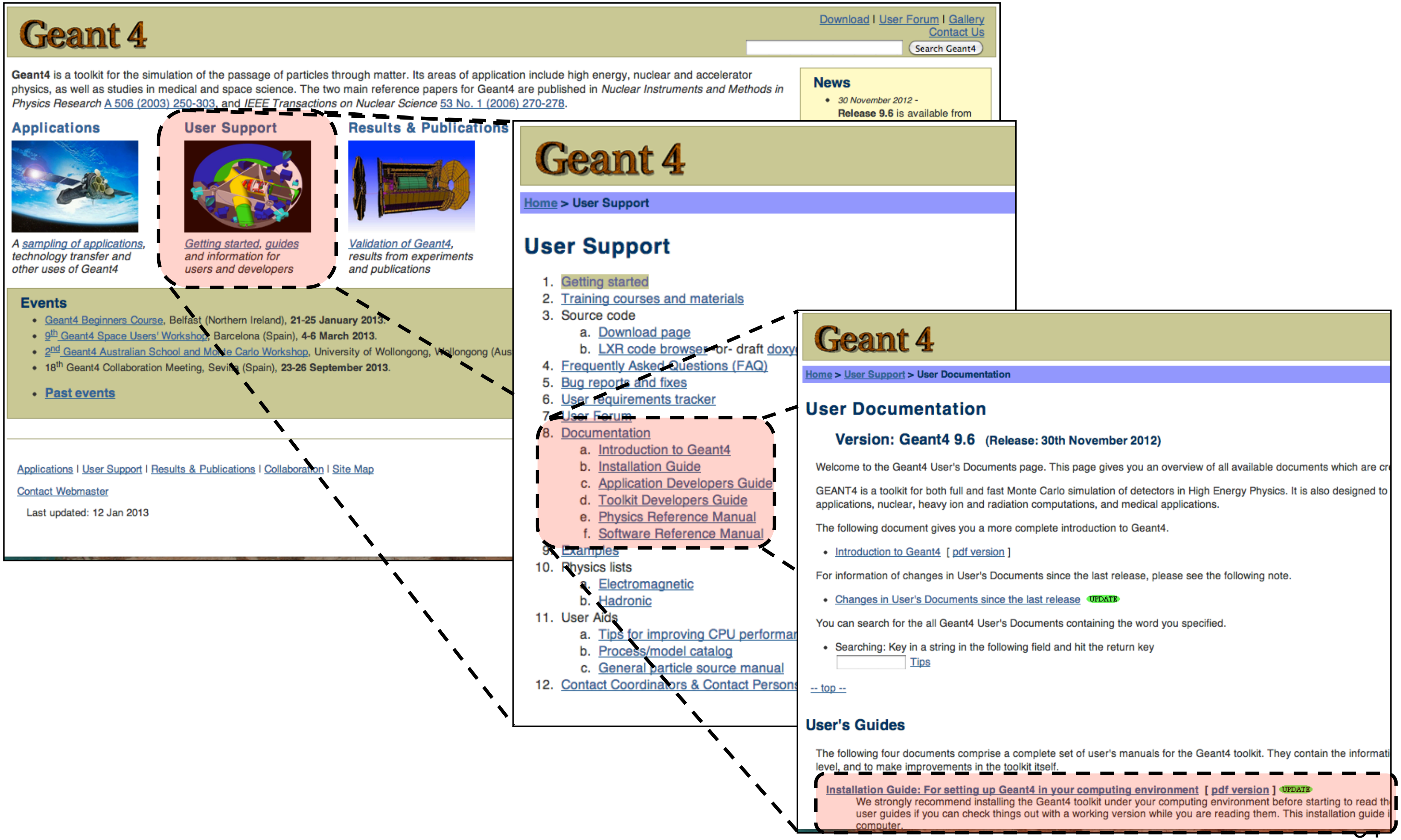


```
$ ./exampleB1
+++ G4StackManager uses G4SmartTrackStack. +++

*************************************************************
Geant4 version Name: geant4-09-06-ref-00    (30-November-2012)
                     Copyright : Geant4 Collaboration
                     Reference : NIM A 506 (2003), 250-303
                     WWW : http://cern.ch/geant4
*************************************************************


<<< Reference Physics List QBBC
Checking overlaps for volume Envelope ... OK!
Checking overlaps for volume Shape1 ... OK!
Checking overlaps for volume Shape2 ... OK!
WARNING: G4QInelastic is deprecated and will be removed in GEANT4 version 10.0.
### Adding tracking cuts for neutron  TimeCut(ns)= 10000  KinEnergyCut(MeV)= 0
Visualization Manager instantiating with verbosity "warnings (3)"...
Visualization Manager initialising...
Registering graphics systems...
```

- 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 of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The two main reference papers for Geant4 are published in *Nuclear Instruments and Methods in Physics Research* A 506 (2003) 250-303, and *IEEE Transactions on Nuclear Science* 53 No. 1 (2006) 270-278.

News

- *30 November 2012 -*
  Release 9.6 is available from

Applications

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

User Support

Getting started, guides and information for users and developers

Results & Publications

Validation of Geant4, results from experiments and publications

Events

- Geant4 Beginners Course, Belfast (Northern Ireland), **21-25 January 2013**.
- 9th Geant4 Space Users' Workshop, Barcelona (Spain), **4-6 March 2013**.
- 2nd Geant4 Australian School and Monte Carlo Workshop, University of Wollongong, Wollongong (Aus
- 18th Geant4 Collaboration Meeting, Sevilla (Spain), **23-26 September 2013**.
- **Past events**

Applications | User Support | Results & Publications | Collaboration | Site Map

Contact Webmaster

Last updated: 12 Jan 2013

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 doxy
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
   f. Software Reference Manual
9. Examples
10. Physics lists
    a. Electromagnetic
    b. Hadronic
11. User Aids
    a. Tips for improving CPU performar
    b. Process/model catalog
    c. General particle source manual
12. Contact Coordinators & Contact Persons

Geant 4

Home > User Support > **User Documentation**

User Documentation

Version: Geant4 9.6 (Release: 30th November 2012)

Welcome to the Geant4 User's Documents page. This page gives you an overview of all available documents which are cr

GEANT4 is a toolkit for both full and fast Monte Carlo simulation of detectors in High Energy Physics. It is also designed to applications, nuclear, heavy ion and radiation computations, and medical applications.

The following document gives you a more complete introduction to Geant4.

- Introduction to Geant4 [pdf version]

For information of changes in User's Documents since the last release, please see the following note.

- Changes in User's Documents since the last release UPDATE

You can search for the all Geant4 User's Documents containing the word you specified.

- Searching: Key in a string in the following field and hit the return key
  Tips

-- top --

User's Guides

The following four documents comprise a complete set of user's manuals for the Geant4 toolkit. They contain the informati level, and to make improvements in the toolkit itself.

Installation Guide: For setting up Geant4 in your computing environment [pdf version] UPDATE
We strongly recommend installing the Geant4 toolkit under your computing environment before starting to read th user guides if you can check things out with a working version while you are reading them. This installation guide i computer.

The Geant4 example categories

- Under `../geant4.9.6-install/share/Geant4-9.6.0/examples`:
  - Basic examples
    - Most typical use-cases Geant4 application (keeping simplicity and easy of use)
  - Novice examples
    - Applications ranging from non-interacting particle to very complex detectors simulation
  - Extended examples (Demonstration of Geant4 specific usage)
    - Electromagnetic
    - Analysis
    - Biasing
    - Visualization
    - ....
  - Advanced examples (Simulation of real experimental set-up or devices)
    - Brachytherapy
    - Gammaray_telescope
    - Medical_linac
    - Hadrontherapy

A Geant4 application

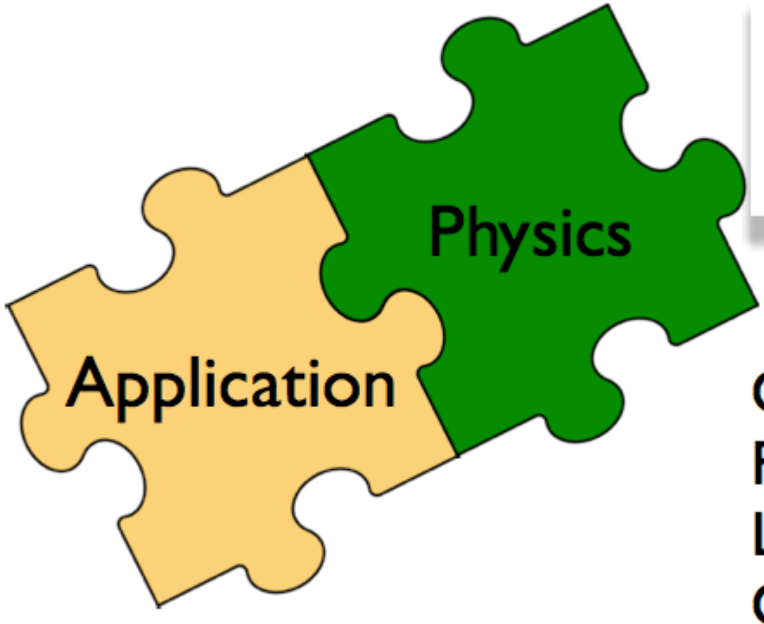
A Geant4 application

- 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

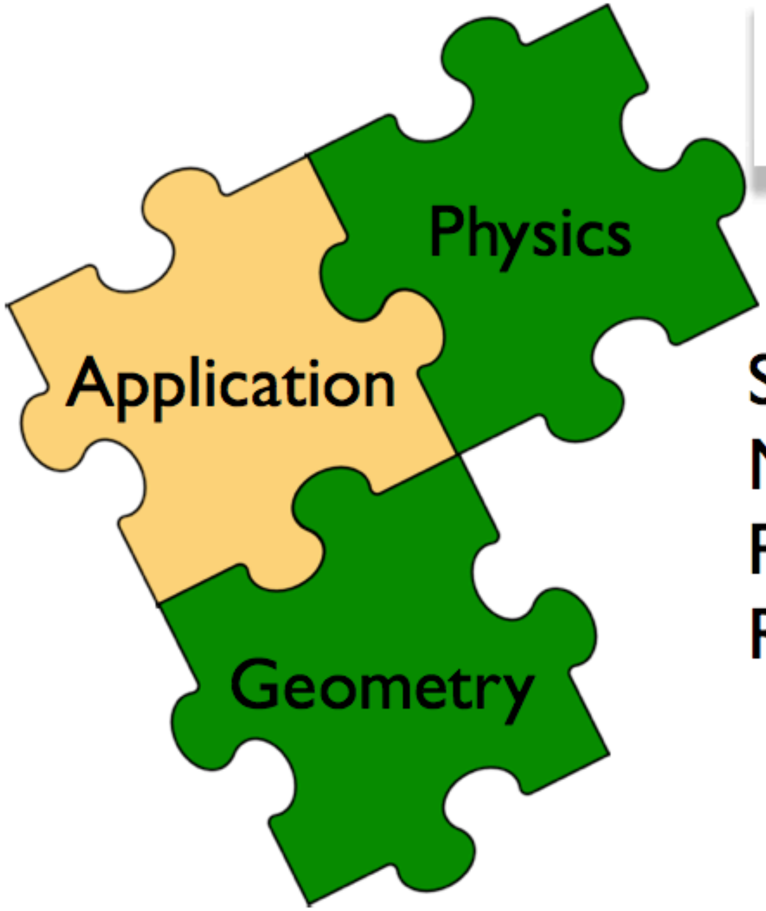
A Geant4 application



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

QGSP_BERT
FTFP_BERT
LHEP
QGSP_BIC
CHIPS
....

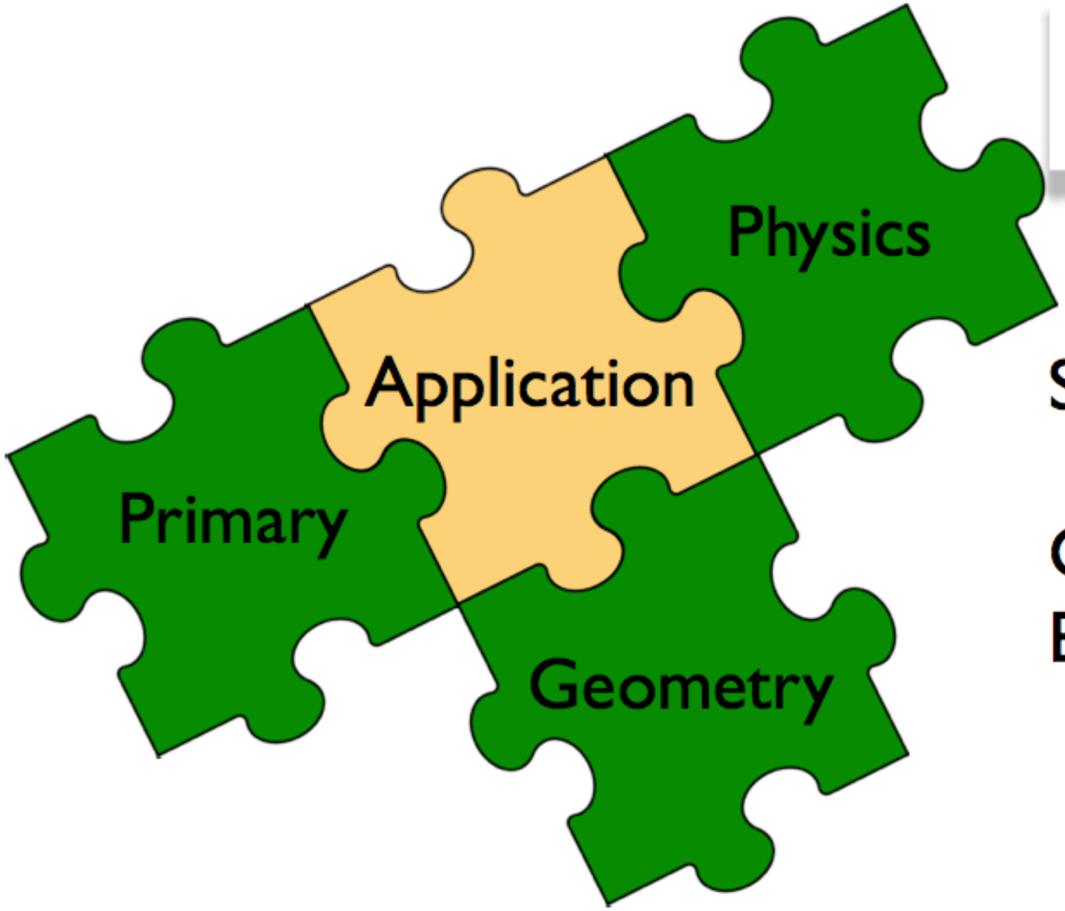
A Geant4 application



Describe your detector
using G4 geometry

Shapes,
Materials,
Placements,
Read-out geo

A Geant4 application



Describe the source of radiation

Simple Gun

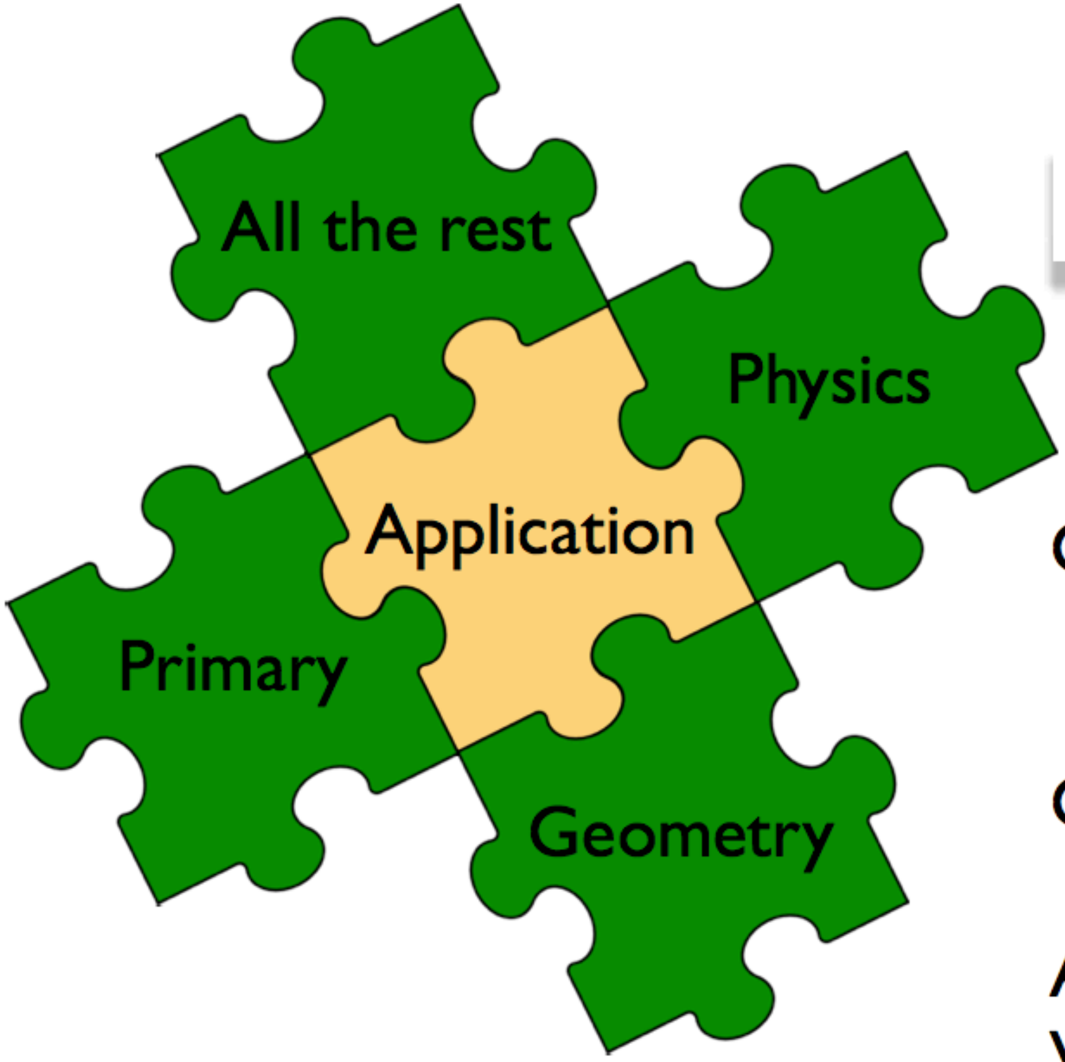
test-beam like

Generic source

External input

HepMC files

A Geant4 application



Add all the rest

G4UserActions
interact with simulation

G4Hits/Digits
read-out

Analysis

Visualization

Thanks for your attention