## **Geant4 Installation Guide**

#### Building and Installing Geant4 for Users and Developers

#### Version: geant4 10.0

Publication date 6 December 2013

**Geant4 Collaboration** 

## Geant4 Installation Guide : Building and Installing Geant4 for Users and Developers

by Geant4 Collaboration

Version: geant4 10.0

Publication date 6 December 2013

#### **Table of Contents**

1. Getting Started	. 1
1.1. Supported and Tested Platforms	. 1
1.2. Software Required to Build Geant4	. 1
1.2.1. CLHEP Library	. 2
1.3. Software Required to Build Optional Components of Geant4	. 2
1.4. Software Suggested for Use With Geant4	
2. Building and Installing Geant4	
2.1. Building and Installing on Unix Platforms	. 4
2.2. Building and Installing on Windows Platforms	
2.3. Geant4 Build Options	
2.3.1. Standard Options	14
2.3.2. Advanced Options	17
2.3.3. Selecting a Different Compiler and Changing Flags	19
2.3.4. Using an Initial Cache File for Build Options	21
3. Setting Up and Using an Install of Geant4	
3.1. Geant4 Installation Locations	
3.2. Building Applications with Geant4	25
3.2.1. Using CMake to build Applications: Geant4Config.cmake	26
3.2.2. Using Geant4Make to build Applications: binmake.gmk	45
3.2.3. Other Unix Build Systems: geant4-config	
3.3. Note on Geant4 Datasets	48
4. CMake and Build Tools For Geant4 Developers	53
4.1. Developing Geant4 using Make, Xcode, Visual Studio and Eclipse	53
4.1.1. Using the Eclipse IDE	53
4.2. Command Line Help with Make	54
4.3. Building Quickly and Efficiently with Multiple Build Directories	54
4.4. Building Test Applications Against Your Development Build	55
5. Help And Support	56
5.1. Getting Help	56
5.2. Further Information	56

#### **Chapter 1. Getting Started**

Geant4 uses CMake to configure a build system for compiling and installing the toolkit headers, libraries and support tools. This document covers the basics of using CMake to build and install Geant4 together with an overview of the most commonly used advanced features. We also provide a basic overview of how to build an application that uses Geant4. For more information on CMake itself, the CMake Help and Documentation should be consulted.

#### **1.1. Supported and Tested Platforms**

Geant4 is officially supported on the following operating system and compiler combinations:

• Scientific Linux CERN 6 with gcc 4.7.X, 64bit

Geant4 has also been successfully compiled on other Linux distributions, including Debian, Ubuntu and openSUSE. Please note that distributions other than SLC are not officially supported. However, we welcome feedback and patches for non-SLC platforms!

• Mac OS X 10.8 (Mountain Lion) with clang 3.3 (Apple-LLVM 5.0), 64bit.

Geant4 has also been successfully compiled on Mac OS X 10.7 (Lion) with clang 3.1 (Apple), but this version of Mac OS X is not officially supported. Please note that Mac OS X 10.9 (Mavericks) is not officially supported yet.

• Windows 7 with Visual Studio 11 (VS2011).

Multithreading of Geant4 is currently not supported on the Windows platform.

The following platforms and compilers are also tested, but not officially supported:

• Scientific Linux CERN 6 with gcc 4.4.X, 4.8.X.

Scientific Linux CERN 5 with gcc 4.7.X, 64bit.

Scientific Linux CERN 5 with Intel C++ 14, 64bit.

- Mac OS X 10.7 (Lion) Clang 3.1.
- Windows 7 with Visual Studio 12 (experimental, single threaded only).
- Windows 7 with Visual Studio 9 (VS2008) (single threaded only).

We welcome user feedback and/or bug reports via our HyperNews Forum and Bugzilla.

#### 1.2. Software Required to Build Geant4

The following minimal set of software *must* be present to build Geant4:

- Geant4 Toolkit Source Code.
- CMake 2.6.4 or higher.

If you intend to use the Clang compiler, then you must use CMake 2.8.2 or higher in order to correctly identify this compiler.

We recommend version 2.8.X as this includes many extra useful features, but it is not required apart from the proviso on Clang usage above.

- C++ Compiler:
  - Linux: GNU Compiler Collection.

• Mac OS X: Clang (Xcode 4).

In Xcode 4, you will also need to install the command line tools (via Xcode->Preferences->Downloads).

In Xcode 5, you may need to run xcode-select --install from the command line if the above method does not work.

- Windows: Visual Studio 2011, either Express or Higher Editions.
- Linux/Mac only: Make

On Linux, we recommend that you use CMake as provided through the package management system of your distribution, unless this does not meet the minimum version requirement of CMake 2.6.4. In that case, we recommend you install it using the Linux binary installer for the latest version of CMake, available with instructions from the Kitware download site. This installer is highly portable and should work on the vast majority of distributions.

On Mac and Windows, CMake is not installed by default, so we recommend that you install it using the most recent Darwin64 dmg (Mac) or Win32 exe (Windows) installers supplied by the Kitware download site.

On Linux, it is strongly recommended to use the g++ compiler supplied by the package management system of your distribution (Linux). On OS X, you must use the clang compiler as supplied with Xcode (Mac).

#### 1.2.1. CLHEP Library

Geant4 distributes a minimal version of the CLHEP library sources with the toolkit to help cross-platform usage. This internal version of the CLHEP library is built and used by default, so having an external install of CLHEP is no longer a prerequisite for Geant4.

However, Geant4 can still be configured to use an existing install of CLHEP if required by your use case. This configuration is done by passing extra options to CMake, and if your require this feature you should consult Section 2.3.

#### **1.3. Software Required to Build Optional Components of Geant4**

Geant4 has several optional components which if enabled require further software to be preinstalled on your system. These components and their requirements are listed below.

• GDML Support (All Platforms)

*Requires*: Xerces-C++ headers and library.

• Qt User Interface and Visualization (All Platforms)

Requires: Qt4 headers and libraries, OpenGL or MesaGL headers and libraries.

We recommend Qt4, though you may optionally use the newer Qt5. Note however that Qt5 support is still experimental.

On Mac OS X, you should use Qt 4.8.X.

• Motif User Interface and Visualization (*Linux and Mac OS X*)

Requires: Motif and X11 headers and libraries, OpenGL or MesaGL headers and libraries.

• X11 OpenGL Visualization (*Linux and Mac OS X*)

Requires: X11 headers and libraries, OpenGL or MesaGL headers and libraries.

On Mac OS X 10.8 (Mountain Lion), X11 can be obtained through the XQuartz project.

• WIN32 OpenGL Visualization (Windows)

Requires: OpenGL or MesaGL headers and libraries.

• Open Inventor Visualization (All Platforms)

*Requires*: Coin3D with SoXt(SoWin) graphics binding on Linux/Mac(Windows). Coin3D is a free implementation of Open Inventor.

• X11 RayTracer Visualization (Linux and Mac OS X)

Requires: X11 headers and libraries.

On Mac OS X 10.8 (Mountain Lion), X11 can be obtained through the XQuartz project.

On Linux, it is strongly recommend that you use the binary packages as supplied through the package management system of your distribution. If you require a component that uses OpenGL, we also recommend that you install the OpenGL package supplied for your video card (e.g. NVIDIA). You should consult the documentation of your distribution for information on the packages that provide the needed software libraries and headers.

On Mac and Windows, we strongly recommend installing any required packages through binary dmg/exe installers supplied through the vendor links above. Note that Visual Studio supplies an install of OpenGL on Windows. Mac OS X 10.8 supplies OpenGL, but if you need X11 you will need to install the XQuartz app. Installation and use of packages on Mac through MacPorts, fink or homebrew is not tested or supported, but you may build Geant4 using packages supplied by these package management systems with that caveat.

#### **1.4. Software Suggested for Use With Geant4**

Geant4 includes many cross-platform file-based visualization drivers, together with the lightweight inexlib for basic analysis. Geant4 does not require any additional software over and above that listed in Section 1.2 to *build and install* these components.

However, you may wish to install the software suggested below to make use of these components when running your Geant4 application. We again emphasize that you do not need these packages to build and install Geant4.

- DAWN postscript renderer (for use with DAWN visualization driver).
- HepRApp Browser (for use with HepRep visualization driver).
- WIRED4 JAS Plug-In (for use with HepRep visualization driver).
- VRML Browser (for use with VRML visualization driver).
- OpenScientist interactive environment for analysis.
- AIDA implementation such as OpenScientist, iAIDA, JAS3 or rAIDA.
- gMocren volume visualizer for Geant4 medical simulations.

For more details on Geant4's visualization and analysis components, you should consult the relevant sections in the Geant4 User's Guide for Application Developers.

### Chapter 2. Building and Installing Geant4 2.1. Building and Installing on Unix Platforms

Unpack the Geant4 source package geant4.10.00.tar.gz to a location of your choice. For illustration *only*, this guide will assume it's been unpacked in a directory named /path/to, so that the Geant4 source package sits in a subdirectory

/path/to/geant4.10.00

We refer to this directory as the *source directory*. The next step is to create a directory in which to configure and run the build and store the build products. This directory should not be the same as, or inside, the source directory. In this guide, we create this *build directory* alongside our source directory:

```
$ cd /path/to
$ mkdir geant4.10.0-build
$ ls
geant4.10.00 geant4.10.0-build
```

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

```
$ cd /path/to/geant4.10.0-build
$ cmake -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.0-install /path/to/geant4.10.00
```

Here, the CMake Variable CMAKE\_INSTALL\_PREFIX is used to set the *install directory*, the directory under which the Geant4 libraries, headers and support files will be installed. It must be supplied as an absolute path. The second argument to CMake is the path to the source directory. In this example, we have used the absolute path to the source directory, but you can also use the relative path from your build directory to your source directory.

Additional arguments may be passed to CMake to activate optional components of Geant4, such as visualization drivers, or tune the build and install parameters. See Section 2.3 for details of these options. If you run CMake and decide afterwards you want to activate additional options, simply rerun CMake in the build directory, passing it the extra options plus the build directory. For example, after running CMake as above, you may wish to activate the installation of Geant4's datasets, so you would run

```
$ cd /path/to/geant4.10.0-build
$ cmake -DGEANT4_INSTALL_DATA=ON .
```

On executing the CMake command, it will run to configure the build and generate Unix Makefiles to perform the actual build. CMake has the capability to generate buildscripts for other tools, such as Eclipse and Xcode, but please note that *we do not support user installs of Geant4 with these tools*. On Linux, you will see output similar to:

```
$ 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 - found
```

```
-- Check size of off64_t
-- Check size of off64_t - done
-- Looking for fseeko
-- Looking for fseeko - found
-- Looking for unistd.h
-- Looking for unistd.h - found
-- Pre-configuring dataset G4NDL (4.4)
-- Pre-configuring dataset G4EMLOW (6.35)
-- Pre-configuring dataset PhotonEvaporation (3.0)
-- Pre-configuring dataset RadioactiveDecay (4.0)
-- Pre-configuring dataset G4NEUTRONXS (1.4)
-- Pre-configuring dataset G4PII (1.3)
-- Pre-configuring dataset RealSurface (1.0)
-- Pre-configuring dataset G4SAIDDATA (1.1)
-- Pre-configuring dataset G4ABLA (3.0)
-- Pre-configuring dataset G4ENSDFSTATE (1.0)
  *WARNING*
    Geant4 has been pre-configured to look for datasets
    in the directory:
    /path/to/geant4-10.0-install/share/Geant4-10.0.0/data
    but the following datasets are NOT present on disk at
    that location:
    G4NDL (4.4)
    G4EMLOW (6.35)
    PhotonEvaporation (2.3)
    RadioactiveDecay (3.7)
    G4NEUTRONXS (1.4)
    G4PII (1.3)
    RealSurface (1.0)
    G4SAIDDATA (1.1)
    G4ABLA (3.0)
    G4ENSDFSTATE (1.0)
    If you want to have these datasets installed automatically
    simply re-run cmake and set the GEANT4_INSTALL_DATA
    variable to ON. This will configure the build to download
    and install these datasets for you. For example, on the
    command line, do:
    cmake -DGEANT4_INSTALL_DATA=ON <otherargs>
    The variable can also be toggled in ccmake or cmake-qui.
    If you're running on a Windows system, this is the best
    solution as CMake will unpack the datasets for you
    without any further software being required
    Alternatively, you can install these datasets manually
    now or after you have installed Geant4. To do this,
    download the following files:
    http://geant4.cern.ch/support/source/G4NDL.4.4.tar.gz
    http://geant4.cern.ch/support/source/G4EMLOW.6.35.tar.gz
    http://geant4.cern.ch/support/source/G4PhotonEvaporation.3.0.tar.gz
    http://geant4.cern.ch/support/source/G4RadioactiveDecay.4.0.tar.gz
    http://geant4.cern.ch/support/source/G4NEUTRONXS.1.4.tar.gz
    http://geant4.cern.ch/support/source/G4PII.1.3.tar.gz
    http://geant4.cern.ch/support/source/RealSurface.1.0.tar.gz
    http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
    http://geant4.cern.ch/support/source/G4ABLA.3.0.tar.gz
    http://geant4.cern.ch/support/source/G4ENSDFSTATE.1.0.tar.gz
    and unpack them under the directory:
    /path/to/geant4-10.0-install/share/Geant4-10.0.0/data
    As we supply the datasets packed in gzipped tar files,
    you will need the 'tar' utility to unpack them.
    Nota bene: Missing datasets will not affect or break
               compilation and installation of the Geant4
               libraries.
```

-- 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/geant4.10.0-build

On Mac OS X, the output will have slight differences, but the last three lines at least should be the same. These indicate a successful configuration.

The warning message about datasets is simply an advisory. Due to the size of the datasets, Geant4 will try and reuse any datasets it can find under the data installation prefix, in our example case /path/to/geant4.10.0-install/share/Geant4-10.0.0/data. If any datasets are not found here, it will pre-configure the setup scripts for using Geant4 (described in Section 3.1 and Section 3.2) to point to this location and emit the message to advise you on the steps you need to take to manually install the datasets at a time of your convenience.

Datasets are *not* required to be present to build Geant4, but may be required to run your application, depending on the physics models you use. If you wish to download and install the datasets automatically as part of your build of Geant4, simply add the option -DGEANT4\_INSTALL\_DATA=ON to the arguments passed to CMake. Note that this requires a working network connection and will download around 0.5GB of data. If you already have the datasets present on your system, you can point Geant4 to their location. See the GEANT4\_INSTALL\_DATADIR option described Section 2.3.1 for more details.

If you see any errors at this point, carefully check the error messages output by CMake, and check your install of CMake and C++ compiler first. The default configuration of Geant4 is very simple, and provided CMake and the compiler are installed correctly, you should not see errors.

After the configuration has run, CMake will 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 (e.g. if your machine has a dual core processor, you could set N to 2).

The build will now run, and will output information on the progress of the build and current operations. If you need more output to help resolve issues or simply for information, run make as

\$ make -jN VERBOSE=1

Once the build has completed, you can install Geant4 to the directory you specified earlier in CMAKE\_INSTALL\_PREFIX by running

```
$ make install
```

in the build directory. The libraries, headers and resource files are installed under your chosen install prefix in a standard Unix-style hierarchy of directories, described below in Section 3.1. If you are performing a staged install for packaging or deployment, the CMake generated Makefiles support the DESTDIR variable for copying to a temporary location. To uninstall Geant4 you can run

```
$ make uninstall
```

which will remove all installed files but not any installed directories.

#### 2.2. Building and Installing on Windows Platforms

Unpack the Geant4 source package, geant4\_10\_00.zip to a location of your choice. For illustration *only*, this guide will assume it's been unpacked in a directory C:\Users\Ben\Documents\geant4\, so that the Geant4 source package resides in the subdirectory

```
C:\Users\Ben\Documents\geant4\geant4_10_00
```

We refer to this directory as the source directory.

Whilst CMake can be used from the Windows cmd program, we recommend using the CMake GUI for configuring the build. The steps required to use the CMake GUI and Visual Studio to build and install Geant4 are described below.

1. Open the CMake (cmake-gui) executable, and click on the *Browse Source*... button in the top right hand corner of the window. Use the file browser popup to locate the Geant4 source directory, and click *OK*.

+						
Recycle Bin	Skype					
Recycle bin	CMske 2.8.12.1 -					
ę.	File Tools Options Help					
<b>.</b>	Where is the source code:				Browse Source	
Adobe Reader X	Where to build the binaries:			•	Browse Build	
	Search:		📝 Grouped 🛛 🕅 Advanced	de Add Entry	Remove Entry	
	Name	Value				
CCleaner		(				
		Browse For Folder	<b>•</b> ו			
		Enter Path to Source				
E		Downloads	*			
CMake		Favorites				
(cmake-gui)		🔰 Links 4 📔 My Documents				
_		■ Wy bocuments ■ B geant4				
3		⊿ 🌽 geant4( 🖢 00				
Cygwin		D b b cmake b b config Date creat	nd: 26/11/2013 12:49			
		Environments	E			
		examples				
		ReleaseNotes Isource				
Mozilla Firefox		Visual Studio 2010				
	Press Configure to update and display new	INTY MUSIC				
	Configure Generate Current Generator: None	My Pictures My Videos				
OpenOffice		Saved Games	-			
OpenOffice 4.0.0		Make New Folder OK	Cancel			
1000						
PDFCreator						
<b>@</b>	2 📋 🛛 🔺					▲ 🕪 🏴 📁 13:48 28/11/2013

2. Now we create a directory in which to create the Visual Studio project files and hold the build products. This directory should not be the same as, or inside, the the source directory. Here we will create this *build directory* alongside our source directory.

Click on the *Browse Build*... button in the top right hand side of the CMake GUI window. Use the file browser popup to browse back to C:\Users\Ben\Documents\geant4\, and click on the *Make New Folder* button. Rename the created folder to *geant4.10.0-build*, and click on the *OK* button. The two text entries at the top of the GUI should now contain C:/Users/Ben/Documents/geant4/geant4\_10\_00 and C:/Users/Ben/Documents/geant4/geant4\_10\_00 and C:/Users/Ben/Documents/geant4/geant4\_10\_00 and C:/Users/Ben/Documents/geant4/geant4\_10\_00 and C:/Users/Ben/Documents/geant4.10.0-build respectively (*Note*: CMake always represents Windows paths with forward slashes).

À.	Options Help				
Adobe Reader X Where to build	urce code: C:/Users/Ben/Documents/geant4/geant4_10_00			Browse Source	
Reader X Search:	ne un rantes:		🗹 Grouped 📃 Advanced 📑		
Name		Value			
CCleaner CMake CMake Cygwin Friefou Corrigue Corrigue Corrigue Corrigue Corrigue Corrigue Corrigue PBCCreator	Press Configure to update and display new Generate Current Generator: None	Browne for Folder Enter Path to Buld Ben Contacts Contacts Foundads Foundads Foundads My Documents Germke Ge	Cancel		

3. Click on the *Configure* button on the bottom left hand side of the GUI.

Recycle Bin	Skype										
<u>, </u>	CMake 2.8.12.1 - C:/Us File Tools Options Where is the source code:	Help	ents/geant4/geant4 iocuments/geant4/ge						Browse Source		
Adobe Reader X	Where to build the binaries:								Browse Build		
	Search:						🗹 Grouped 📄 Advancer	d 🚭 Add Entry	🗱 Remove Entry		
	Name				Value						
CCleaner											
CMake (cmake-gui)											
Cygwin											
Mozilla Firefox											
				to update and display nev	values in red, the	n press Generate to ger	nerate selected build files.				
OpenOffice 4.0.0	Configure Genera	te Current Ge	nerator: None								
PDFCreator											
<b>(</b>	🤌 [									- • P (1	13:51 28/11/201

4. In the popup window, select *Visual Studio 2010*. Ensure only the *Use default native compilers* radio box is checked. Then click on the *Finish* button.

Recycle Bin	Stype CMake 2.8.12.1 - Cl/Userr/Ben/Documents/geant4/geant4.18.0-b File Tools Options Help Where is the source code: Cl/Users/Ben/Documents/geant4/geant4_1/				Browse Source		
Adobe Reader X	Where to build the binaries: C:/Users/Ben/Documents/geant4/geant4.10	0.0-build			Browse Build		
	Search:		🗹 Grouped 🔝 Advanced 🏾	💠 Add Entry	3 Remove Entry		
CCleaner CCleaner (cMake (cmake-gui) Cygwin Cygwin	Name	Specify the generator for this project Visual Stude 10 Use default native complets Specify tookhain file for cross-compling Specify spitons for cross-compling	Finish N Conc	•			
Mozilla Firefox	Press Configure to upda	te and display new values in red, then press Generate to	generate selected build files.				
	Configure Generate Current Generator: None						
OpenOffice 4.0.0 PDFCrestor							
	2 📋 🖸 🔺					• • • •	13:51 28/11/201

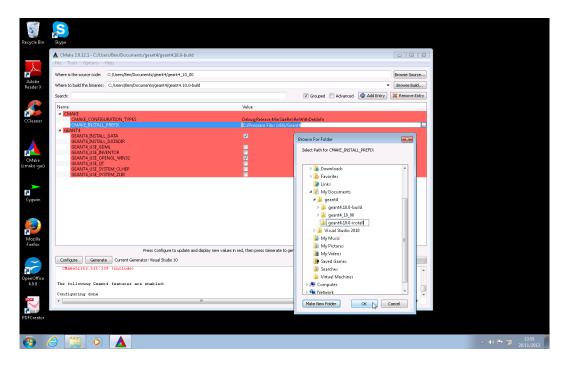
5. CMake will then run to check features and provide an initial configuration. All being well, you will see red highlighted entries in the main options window including *CMAKE* and *GEANT4* entries. Note that the red highlighting means CMake has not fully resolved all configuration variables yet, not that has been an error. On Express editions of Visual Studio, you may see a warning about missing system runtime libraries in the logging window at the bottom of the GUI, but this can be ignored for a standard install. On the latest versions of CMake, you may also see warnings relating to LINK\_INTERFACE\_LIBRARIES and these can be ignored. Any other errors will be reported in this logging window.

				_
2	S			
Recycle Bin	Skype			
		ers/Ben/Documents/geant4/geant4/geant4.10.0-build		
C	File Tools Options			
<b>N</b>	Where is the course code:	C:/Users/8en/Documents/geant4/geant4_10_00 Browse Source		
Adobe Reader X				
Reader X		C:/Lisers/Ben/Documents/geant4/geant4.10.0-build		
	Search:	🕼 Grouped 📄 Advanced 🌗 Add Entry		
	Name	Value		
CCleaner	<ul> <li>CMAKE</li> <li>GEANT4</li> </ul>			
colonici				
5				
CMake (cmake-gui)				
_				
15				
Cygwin				
<b>,</b>				
Mozilla				
Mozilla Firefox				
		Press Configure to update and display new values in red, then press Generate to generate selected build files.		
	Configure Generat	e Current Generator: Visual Studio 10		
	unagebists.txt:13	include)		
OpenOffice 4.0.0	The following Geant	4 features are enabled:		
	Configuring done			
	•	т		
Tiou D				
PDFCreator				
	2		- • P 🔁 ,	1
<b>1</b>	2 📋 🔘		· • • • • • • • • • • • • • • • • • • •	

6. By default, CMake will configure the build to install Geant4 under C:/Program Files/Geant4. If you do not wish to use this directory, or do not have permission to install there, you can change the installation location. Click on the arrow next to the the *CMAKE* entry in the central options display to expand all CMake specific configuration options. Scroll down to locate the CMAKE\_INSTALL\_PREFIX entry. Click on the

entry and edit the path directly or via the file browser popup. In this example we will modify this *install directory* to C:/Users/Ben/Documents/geant4/geant4.10.0-install.

If you need to activate extra components of Geant4, click on the expansion arrow next to the *GEANT4* entry in the central options display to expand all Geant4 specific configuration options. Simply click on the tick box next to an option to select it if you require. *For illustration only*, we will enable two options, GEANT4\_INSTALL\_DATA and GEANT4\_OPENGL\_WIN32 which will download and install physics datasets for you and enable OpenGL support respectively. Note that GEANT4\_INSTALL\_DATA requires a working network connection so that datasets can be downloaded, so if you are working offline you should disable it. These, and other options, are documented in Section 2.3.



7. Once you have adjusted any options, click on the *Configure* button again. After this has finished, all the options listed in the central option display should be white. If you still have entries in red, click *Configure* to reconfigure until all entries are white.

Recycle Bin	Skype							
	A CMake 2.8.12.1 - C:/Use	rs/Ben/Documents/geant4/geant4.10.0-build						
- X	File Tools Options	Help						
	Where is the source code:	C:/Users/Ben/Documents/geant4/geant4_10_00				Browse Source		
Adobe Reader X	Where to build the binaries:	C:/Users/Ben/Documents/geant4/geant4.10.0-build			•	Browse Build		
	Search:		[	Grouped 📄 Advanced	💠 Add Entry	🗱 Remove Entry		
	Name	Va	lue					
	CMAKE							
CCleaner	GEANT4							
CMake (cmake-gui)								
Cygwin								
Mozilla Firefox		Press Configure to update and display new values in red,	hen press Generate to generate	selected build files.				
	Configure							
OpenOffice 4.0.0	GEANT4_USE_OPENGL_5	4 features are enabled: IN32: Build OpenGL driver with Win32 support						
	Configuring done					-		
$\sim$	•	m				•		
PDFCreator								
roreator								
	2							13:57

8. Now click on the *Generate* button to generate the Visual Studio project. All being well, CMake will generate the solution files and finally report *Configuring done, Generating done* in the logging window at the bottom of the GUI. You can now close the CMake GUI.

3	S						
Recycle Bin	Skype						
	A CMake 2.8.12.1 - C:/Use	rs/Ben/Documents/geant4/geant4.10.0-build					
X	File Tools Options	Help					
	Where is the source code:	C:/Users/Ben/Documents/geant4/geant4_10_00				Browse Source	
Adobe Reader X	Where to build the binaries:	C:/Users/Ben/Documents/geant4/geant4.10.0-build			•	Browse Build	
	Search:			📝 Grouped 🛛 Advanced	💠 Add Entry	3 Remove Entry	
	Name		Value				
	<ul> <li>CMAKE</li> <li>GEANT4</li> </ul>						
CCleaner	P ODAININ						
CMake (cmake-gui)							
(cmake-gui)							
3							
Cygwin							
Mozilla							
Mozilla Firefox							
		_	ew values in red, then press Generate to gene	rate selected build files.			
$\sim$	Configure Generat	Current Generator: Visual Studio 10					
	The following Geant	4 features are enabled:				*	
OpenOffice 4.0.0	GEANT4_USE_OPENGL_0	IN32: Build OpenGL driver with Win32 suppo	rt				
4.0.0	Configuring done						
ROS	Generating done					*	
$\sim$	•						
PDFCreator							
<b>1</b>	2 🚞 🔾						13:59 28/11/2013

9. Now start up Visual Studio, and choose *Open Project*. This guide is somewhat specific to Visual Studio 2010 Express, but the solutions should appear with the same names in other versions. Browse to your build directory, and open the Geant4.sln Microsoft Visual Studio Solution file. Note that it may take some time to fully configure, open and parse all files in the solution. You may also use the MSBuild command line utility to build the solution directly if you are familiar with that tool, but we do not as yet document or support this.

D10 Express C+t Stated Laboration ment + My Documents + geant4108-buil folder + 2038 Expre Documents library geant410.0-build Name Marketiles		rh geant4.10.0-build	express deve	ckly create exciting interactive
ment: ) My Document: ) geant4.10.4-buil folder 2030 Expre Documents library geant4.10.0-build Name	d > • 49 Sean	xh geant4.10.0-build IⅢ ▼ 🛄	P velopers qui	
ment: > My Document: > geant4 > geant4.18.8-buil folder 2010 Expre Documents library geant4.10.0-build Name	d > • 49 Sean	xh geant4.10.0-build IⅢ ▼ 🛄	P velopers qui	
ment: > My Document: > geant4 > geant4.18.8-buil folder 2010 Expre Documents library geant4.10.0-build Name	d > • 49 Sean	xh geant4.10.0-build IⅢ ▼ 🛄	P velopers qui	
Tolder		ii • 🔟	velopers qui	
++ 2010 Expre		ii • 🔟	velopers qui express deve	
++ 2010 Expre	Date modified		velopers qui express deve	
geant4.10.0-build	Date modified	Arrange by: Folder -	express deve	
Name	Date modified		idea to grea	lopment environment, improved at application has never been easie
		-		nd the latest and coolest projects o
CMakeFiles	Date mounted	Туре	2	
	28/11/2013 13:58	File folder		
🏓 data	28/11/2013 13:56	File folder		
🕌 Externals	28/11/2013 13:56	File folder		
InstallTreeFiles	28/11/2013 13:56	File folder		
Modules	28/11/2013 13:53	File folder	E	
source	28/11/2013 13:58	File folder		
	28/11/2013 13:58	VC + + Project		
🞇 G4ABLA	28/11/2013 13:58	VC++ Project		
	28/11/2013 13:58	VC + + Project		
)-install 🔅 G4NDL	28/11/2013 13:58	VC++ Project		
nn G4NEUTRONXS	28/11/2013 13:58	VC + + Project		
2010 3 G4PII	28/11/2013 13:58	VC++ Project		
	28/11/2013 13:58	VC + + Project		
	28/11/2013 13:58	VC++ Project		
	28/11/2013 13:58	VC++ Project	-	
÷ 1			•	
	s Carbon Control Contr	Image: Second	Image: Source	Image: Second

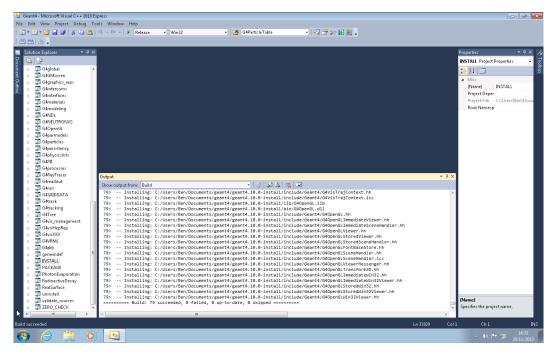
10. By default, the Visual Studio solution builds the Debug Configuration, so if you want an optimized build, you should select the *Release* option from the drop down *Solutions Configurations* menu in the toolbar (Visual Studio 2010).

🤹 Geant4 - Microsoft Visual C ++ 2010 Express		9 🛛 🗙
File Edit View Project Debug Tools Window Help		
: 🔂 = 🔄 = 💕 🚽 🐉 🐁 🐁 🖏 👘 = 🔍 = 🕨 Debug 🔹 Win32	🗸 🎯 G4ParticleTable 🔹 🔍 🕾 😥 🔐 💂	
i 🕮 🚧 👗 🚽 Debug		
MinSizeRel		_
Control Explorer     → ♀     ♀     Release       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○       0     →     ○     ○		i × 🌾
RefWithDebMfo Configuration Manager	_G40penGL-archive_Project Pro	p • 🗟
5 🔉 🎇 G4global 🔹		p • Foolboo
🚊 🕞 🎇 G4GMacren	Misc	
	(Name) G4OpenGL-ar	-
💱 🕨 🧱 G4intercoms	Project Deper	CHIV
G4interfaces     G4interfaces     G4interfaces	Project File C:\Users\Ben\	locu
<ul> <li>Journals</li> <li>Journals</li> <li>Journals</li> </ul>	Root Namesp	000
Gamodeling	nocrancy	
GANEUTRONKS		
640penGL		
G4parmodels		
G4persistency		
G4physicslists		
⊳ 🎇 G4PII		
G4processes		
G4RayTracer Output	<del>~</del> म ×	
G4readout     Show output from:	· 2 2 3 3 2	
G4run     G4SAIDDATA		
Gatrack		
G4tracking		
▷ 🖓 G4Tree		
G4vis_management		
G4visHepRep		
G4vis000C		
👂 🎇 G4VRML		
G4zlib		
👂 🎇 genwindef 🗮		
INSTALL		
PACKAGE		
Kaling PhotonEvaporation     KadioactiveDecay		
<ul> <li>a RadioactiveDecay</li> <li>a RealSurface</li> </ul>		
p an reasonace		
> 3 validate_sources	(Name)	
ZERO_CHECK	Specifies the project name.	
4 III +		
Par da		
Ready		
🚱 🥝 🚞 🖸 🖳	- () (* (0 26/11)	31 /2013

11. In the Solution Explorer, right click on the *INSTALL* solution, and select *Build* from the popup menu. The solution will now build and install to the directory you chose for CMAKE\_INSTALL\_PREFIX earlier. If you wish to build a different configuration (*Release* if you built *Debug* first, for example), then simply change the solution configuration as described in the previous paragraph, and build the *INSTALL* solution again.

😫 Geant4 - Microsoft V						
	ect Debug Tools Windo					
	※ 14 15   ウ・ペ・	Release	<ul> <li>Win32</li> </ul>	- 🧭 G4ParticleTable	-   💀 🕾 🗶 🚽 🗸	
i 🕮 🔠 🚠 🖕						
Solution Explorer Course Solution Explorer Course Solution Explorer Course Solution Solution Course Solution So	<b>-</b> ₽ ×					Properties - 4 × 🗴
g 陆 🗿						INSTALL Project Properties -
🗧 🕟 🎇 G4global	A					INSTALL Project Properties -
🗯 🕟 🦉 G4GMocre						4 Misc
G4graphics B R G4graphics	i_reps					(Name) INSTALL
B G4interface						Project Deper
b 🐉 G4material						Project File C:\Users\Ben\Docu
▶ 🗿 G4m 📷 ▶ 🗿 G4NI	Build					Root Namesp
▷ 31 G4NE ▷ 31 G4NE	Rebuild					
b 🧱 640p	Clean					
⊳ 🎇 G4pa	Project Only	•				
⊳ 🎇 G4pa ⊳ 🎇 G4pe	Project Dependencies					
⊳ 🕌 G4pe ⊳ 🎇 G4ph	Project Build Order					
⊳ 🎇 G4PE	Build Customizations					
⊳ 🎇 G4pri	Add	•				
⊳ 🎇 G4Ra ⊳ 🎇 G4rea	References					- ü ×
⊳ 🞇 G4ru	Set as StartUp Project			•   \$   \$ \$ \$ \$ \$	2	
64SA	Debug	•				
⊳ 🎇 G4tra 🐰 ⊳ 🎇 G4tra 🝶	Cut	Ctrl+X				
S GATE -	Paste	Ctrl+V				
🕞 🎇 Gávis 🗡	Remove	Del				
⊳ 🎇 G4vis	Rename	F2				
b States b States b G4VR	Unload Project					
⊳ 🞇 G4zlil	Rescan Solution					
🕞 🔉 🛐 genw	Properties					
INSTALL PACKAGE						
PACKAGE	noration					
b 🧱 Radioactiv	eDecay					
RealSurfacion	e 👘					
▷ 3 uninstall ▷ 3 validate_so						(Name)
Validade_sd						Specifies the project name.
<	•					
Scanning #includes for a	dditional files (36295 of 4121					
						13:33
🕙 🥝	🗒 O 🔛					▲ (>) ■ 13:33 26/11/2013

12. If you see a successful build, you can exit Visual Studio.



The file and directory structure of the installation follows that of the Unix build, and is described in Section 3.1.

#### 2.3. Geant4 Build Options

Both Section 2.1 and Section 2.2 give the minimal proceedure to build and install Geant4. Many additional options can be passed to CMake to adjust the way Geant4 is built and installed and to enable optional components.

On the command line, these options may be set by passing their name and value to the cmake command via – D flags, for example

```
$ cmake -DCMAKE_INSTALL_PREFIX=/opt/geant4 -DGEANT4_USE_GDML=ON \\
/path/to/geant4.10.00
```

would configure the build of Geant4 for installation under /opt/geant4 and compilation of support for GDML.

In the CMake GUI, the options are listed as textbox entries, and values may be set directly by clicking on the entry for the option and entering the requested information (for example, if a path is required, the GUI will pop up a file browser).

On Unix, CMake also provides a curses based interface, ccmake, which can be used to browse and set options in the terminal. Please see the CMake documentation for more information on this command.

If you have already created a build directory and used CMake to configure the build, you can always rerun CMake in that directory with new options to regenerate the buildscripts (Makefiles or IDE solutions). The curses based ccmake command is very useful in this case for browsing the current configuration and for updating it if required. In the CMake GUI, you should set the *Where is the source code:* path to that of your source directory, and the *Where to build the binaries:* path to that of the build directory you wish to reconfigure. You will then need to rebuild and reinstall to pick up the changes. You can also *deactivate* a previously selected option to remove a component from the build. For example, we could do

```
$ cmake -DCMAKE_INSTALL_PREFIX=/opt/geant4 -DGEANT4_USE_GDML=OFF \\
/path/to/geant4.10.00
```

to explicitly remove support for GDML from a build (In the CMake GUI, we would simply uncheck the tick box for GEANT4\_USE\_GDML). Note however that if you reconfigure to *unset* an option and rebuild and reinstall, your

install may contain files installed by the previously set option (for example headers). In this case, you may wish to build the uninstall target before reconfiguring.

Options are divided into *Standard* options, which any user or developer can set directly, and *Advanced* options, which in general are only needed by advanced users, developers or to give very fine control over the build and and install. Some options enable components of Geant4 which require external software (as listed in Section 1.2). If these options are enabled, the required software will be searched for, and hence there are also options which control where CMake should look for these packages. If a required software package is not found, then CMake will exit with an error message detailing what was not found.

#### 2.3.1. Standard Options

We list standard options here in logical order. If you use CMake's curses or GUI interfaces, they will be listed alphabetically.

- CMAKE\_INSTALL\_PREFIX
  - Sets the installation prefix for Geant4. Equivalent to --prefix in Autotools. Its default is platform dependent:

Unix: /usr/local

Windows: C:\Program Files\Geant4

It should be supplied as an absolute path, otherwise CMake will interpret its value relative to your build directory.

See also the CMAKE\_INSTALL\_XXXDIR Advanced Options for fine control of installation locations.

- CMAKE\_BUILD\_TYPE : (DEFAULT : Release)
  - Controls the type of build, at present only the additional flags passed to the compiler. It defaults to Release which gives an optimized build with no debugging symbols. The most useful values are:

Release : Optimized build, no debugging symbols

Debug : Debugging symbols, no optimization

RelWithDebInfo: Optimized build with debugging symbols

Note that if you use a build system which supports multiconfiguration builds (e.g. Xcode, Visual Studio), this variable has no effect as all build types are available inside the CMake generated project.

• • To build Geant4 libraries with support for multithreading on Mac OSX 10.7, C compiler has to be set as clang. At present, this is only supported on Unix systems.

**REQUIRES:** Compiler with support for Thread Local Storage, pthread libraries and headers.

- GEANT4\_BUILD\_MULTITHREADED : (DEFAULT : OFF, Unix Only)
  - If set to ON, build Geant4 libraries with support for multithreading. At present, this is only supported on Unix systems.

REQUIRES: Compiler with support for Thread Local Storage, pthread libraries and headers.

To build Geant4 libraries with support for multithreading on Mac OSX 10.7, the following two additional options are required for both building libraries and building an application. If the user uses an external CLHEP library, it also has to be compiled with these clang options.

```
• CMAKE_C_COMPILER=clang : (no default, Mac OSX 10.7 only)
CMAKE_CXX_COMPILER=clang++ : (no default, Mac OSX 10.7 only)
```

- GEANT4\_INSTALL\_DATADIR : (DEFAULT : CMAKE\_INSTALL\_DATAROOTDIR)
  - Installation directory for Geant4 datasets. It can be supplied as a path relative to CMAKE\_INSTALL\_PREFIX or as an absolute path. It is always searched for existing datasets, which if present will be reused.
- GEANT4\_INSTALL\_DATA : (DEFAULT : OFF)
  - If set to ON, download and install any Geant4 datasets missing from GEANT4\_INSTALL\_DATADIR. Each dataset will be unpacked and installed in the directory pointed to by GEANT4\_INSTALL\_DATADIR.

**REQUIRES** : A working network connection. It is highly recommended to switch this option on if you have a network connection to give the best integration with application development.

- GEANT4\_USE\_GDML : (DEFAULT : OFF | ON if XERCESC\_ROOT\_DIR is set)
  - If set to ON, build the G4persistency library with support for GDML.

REQUIRES : Xerces-C++ libraries and headers, see the XERCESC\_ROOT\_DIR option.

- XERCESC\_ROOT\_DIR
  - If your Xerces-C++ installation is in a non-standard location, set this variable to the root directory of the installation (i.e. the directory containing the include and lib subdirectories for Xerces-C++). If this is not sufficient to locate Xerces-C++, see the Advanced XERCESC\_INCLUDE\_DIR and XERCESC\_LIBRARY options.
- GEANT4\_USE\_G3TOG4 : (DEFAULT : OFF)
  - If set to ON, build the G3ToG4 library for reading ASCII call list files generated from Geant3 geometries.
- GEANT4\_USE\_QT (DEFAULT : OFF)
  - If set to ON, build Qt4/5 User Interface and Visualization drivers.

REQUIRES : Qt4 or Qt5 and OpenGL libraries and headers. See also the QT\_QMAKE\_EXECUTABLE option if CMake has trouble locating your Qt4 installation. Qt5 will be searched for first, and if not found Qt4 will be searched for. Please note that Qt5 support is still experimental.

- QT\_QMAKE\_EXECUTABLE
  - If your Qt4 installation is in a non-standard location, set this variable to point to the qmake executable of the Qt4 installation you wish to use. If you have a system install on Linux or the binary SDK install on other platforms, Qt4 will in general be found automatically (CMake should also honor the QTDIR environment variable).
- GEANT4\_USE\_XM (DEFAULT : OFF, Unix Only)
  - If set to ON, build Motif User Interface and Visualization drivers.

**REQUIRES** : Motif and OpenGL libraries and headers. In most cases, these should be found automatically, but if not, see the Advanced MOTIF\_INCLUDE\_DIR and MOTIF\_LIBRARIES options.

- GEANT4\_USE\_OPENGL\_X11 (DEFAULT : OFF, Unix Only)
  - If set to ON, build the X11 OpenGL visualization driver.

**REQUIRES** : X11 and OpenGL libraries and headers.

- GEANT4\_USE\_OPENGL\_WIN32 (DEFAULT : OFF, Windows Only)
- If set to ON, build the Win32 OpenGL visualization driver.

REQUIRES : OpenGL libraries and headers. If you are using Visual Studio, then this should supply the needed headers and libraries.

- GEANT4\_USE\_INVENTOR (DEFAULT : OFF)
  - If set to ON, build the OpenInventor visualization driver.

REQUIRES : Coin3D Open Inventor implementation, SoXt (Unix) or SoWin (Windows) binding, and OpenGL libraries and headers. CMake will use coin-config and soxt-config if present to locate the Coin3D and SoXt implementation respectively, and will honor the COINDIR environment variable. In case of issues with locating the Inventor implementation, see the Advanced INVENTOR\_INCLUDE\_DIR, INVENTOR\_LIBRARY, INVENTOR\_SOXT\_INCLUDE\_DIR, INVENTOR\_SOXT\_LIBRARY and INVENTOR\_SOWIN\_LIBRARY options.

KNOWN ISSUE : Use of clang compiler and Debug build mode will cause the Inventor driver build to fail with errors relating to Inventor specific debugging functions.

- GEANT4\_USE\_RAYTRACER\_X11 (DEFAULT : OFF, Unix only)
  - If set to ON, build RayTracer visualization driver with X11 support.

**REQUIRES** : X11 Headers and Libraries.

- GEANT4\_USE\_SYSTEM\_CLHEP (DEFAULT : OFF | ON if CLHEP\_ROOT\_DIR set)
  - If set to ON, build Geant4 with an external install of CLHEP. You *should not* set this unless your usage of Geant4 mandates a specific external CLHEP installation (e.g. if your experiment's software uses CLHEP in other tools and requires consistent use of the same CLHEP across the software). If the CLHEP\_ROOT\_DIR option is not set, CLHEP will be searched for under standard system paths.

**REQUIRES** : CLHEP libraries and headers.

- CLHEP\_ROOT\_DIR
  - If you wish GEANT4 to use a specific installation of CLHEP, set this variable to point to the root install directory of the CLHEP installation you wish to use. This directory should contain the include and lib subdirectories containing the CLHEP headers and library respectively. If this is not sufficient to locate CLHEP, see the Advanced CLHEP\_INCLUDE\_DIR and CLHEP\_LIBRARY options.
- GEANT4\_USE\_SYSTEM\_EXPAT (DEFAULT : ON)
  - If set to ON, build Geant4 with an external install of Expat. In this case, the Expat headers and library will be searched for under the standard system paths. If these are not sufficient to locate the required Expat installation, see the Advanced EXPAT\_INCLUDE\_DIR and EXPAT\_LIBRARY options.

Whilst Expat is installed on the vast majority of systems, it may be missing in certain instances. In these cases, simply switch this option to OFF and Geant4 will build and use its internal version of Expat.

**REQUIRES** : Expat library and headers.

- GEANT4\_USE\_SYSTEM\_ZLIB (DEFAULT : OFF)
  - If set to ON, build Geant4 with an external install of zlib. In this case, the zlib headers and library will be searched for under the standard system paths. If these are not sufficient to locate the required zlib installation, see the Advanced ZLIB\_INCLUDE\_DIR and ZLIB\_LIBRARY options.

**REQUIRES** : Zlib library and headers.

#### 2.3.2. Advanced Options

Most builds should never need to touch the advanced options, but should you need more control or CMake has problems locating needed software packages, they can be very helpful. We only list the options and variables most relevent for Geant4. For help on core CMake variables and options, you should consult the Reference Documentation section of the main CMake documentation , and specifically the sections on Variables.

Advanced options are not displayed by default in CMake's curses and GUI interfaces, but can be displayed by pressing t in ccmake, or clicking the 'advanced' check box in the CMake GUI. Note that displaying advanced options will also display a large number of options and variables used by CMake for build configuration and to cache the locations of Third Party packages. On the command line, advanced options and variables can be set like the standard ones listed earlier using –D arguments.

- GEANT4\_USE\_NETWORKDAWN : (DEFAULT : OFF, Unix Only)
  - If set to ON, build network server/client support for DAWN visualization driver. You do *not* need this to view DAWN files.
- GEANT4\_USE\_NETWORKVRML : (DEFAULT : OFF, Unix Only)
  - If set to ON, build network server/client support for VRML visualization driver. You do *not* need this to view VRML files.
- GEANT4\_INSTALL\_DATA\_TIMEOUT : (DEFAULT : 1500)
  - Sets the time in seconds allowed for download of each Geant4 dataset. The value can be increased if you are on a slow network connection and require more time to download.

**REQUIRES** : a working network connection.

- BUILD\_SHARED\_LIBS : (DEFAULT : ON)
- If set to ON build Geant4 shared libraries.
- BUILD\_STATIC\_LIBS : (DEFAULT : OFF)
  - If set to ON, build Geant4 static libraries.
- CMAKE\_INSTALL\_BINDIR : (DEFAULT : bin)
  - Installation directory for Geant4 executables. It can be supplied as a path relative to CMAKE\_INSTALL\_PREFIX or as an absolute path.
- CMAKE\_INSTALL\_INCLUDEDIR : (DEFAULT : include)
  - Installation directory for Geant4 C/C++ headers. It can be supplied as a path relative to CMAKE\_INSTALL\_PREFIX or as an absolute path. The headers will always be installed in a subdirectory of CMAKE\_INSTALL\_INCLUDEDIR named Geant4.
- CMAKE\_INSTALL\_LIBDIR : (DEFAULT : lib(+?SUFFIX))
  - Installation directory for object code libraries. It can be supplied as a path relative to CMAKE\_INSTALL\_PREFIX, or an absolute path. When the default is used, SUFFIX will be set to 64 on 64bit Linux platforms apart from Debian systems.
- CMAKE\_INSTALL\_DATAROOTDIR : (DEFAULT : share)
  - Installation directory for read-only architecture-independent data files. It can be supplied as a path relative to CMAKE\_INSTALL\_PREFIX, or an absolute path.
- XERCESC\_INCLUDE\_DIR

- If CMake cannot locate your Xerces-C++ installation, set this to the directory containing the Xerces-C++ headers (e.g. if you have /foobar/xercesc/util/XercesVersion.hpp, then set this to /foo-bar).
- XERCESC\_LIBRARY
  - If CMake cannot locate your Xerces-C++ installation, set this to the full path to the Xerces-C++ library, e.g. /usr/lib/libxerces-c.so
- MOTIF\_INCLUDE\_DIR
  - If CMake cannot locate your Motif installation, set this to the directory containing the Motif headers (e.g. if you have /foobar/Xm/Xm.h, then set this to /foobar).
- MOTIF\_LIBRARIES
  - If CMake cannot locate your Motif installation, set this to the full path to the Motif library, e.g. /usr/lib/libXm.so
- INVENTOR\_INCLUDE\_DIR
  - If CMake cannot locate your OpenInventor installation, set this to the directory containing the Inventor headers (e.g. if you have /foobar/Inventor/So.h, then set this to /foobar).
- INVENTOR\_LIBRARY
  - If CMake cannot locate your Inventor installation, set this to the full path to the Inventor library, e.g. /usr/lib/libCoin.so
- INVENTOR\_SOWIN\_LIBRARY (Windows only)
  - If CMake cannot locate your Inventor installation, set this to the full path to the Inventor SoWin binding library, e.g. C:\Program Files\Coin\sowin.dll.
- INVENTOR\_SOXT\_INCLUDE\_DIR (Unix only)
  - If CMake cannot locate your Inventor installation, set this to the directory containing the Inventor SoXt binding headers (e.g. if you have /foobar/Inventor/SoXt/SoXt.h, then set this to /foobar).
- INVENTOR\_SOXT\_LIBRARY (Unix only)
  - If CMake cannot locate your Inventor installation, set this to the full path to the Inventor SoXt binding library, e.g. /usr/lib/libSoXt.so.
- CLHEP\_INCLUDE\_DIR (*If GEANT4\_USE\_SYSTEM\_CLHEP selected*)
  - If CMake cannot locate your external CLHEP installation, set this to the directory containing the CLHEP headers (e.g. if you have /foobar/CLHEP/Vector/defs.h, then set this to /foobar).
- CLHEP\_LIBRARY (*If GEANT4\_USE\_SYSTEM\_CLHEP selected*)
  - If CMake cannot locate your CLHEP installation, set this to the full path to the CLHEP library, e.g. /usr/lib/libCLHEP.so
- EXPAT\_INCLUDE\_DIR (*If GEANT4\_USE\_SYSTEM\_EXPAT selected*)
  - If CMake cannot locate your external EXPAT installation, set this to the directory containing the EXPAT headers (e.g. if you have /foobar/expat.h, then set this to /foobar).
- EXPAT\_LIBRARY (*If GEANT4\_USE\_SYSTEM\_EXPAT selected*)

- If CMake cannot locate your EXPAT installation, set this to the full path to the EXPAT library, e.g. /usr/ lib/libexpat.so
- ZLIB\_INCLUDE\_DIR (*If GEANT4\_USE\_SYSTEM\_ZLIB selected*)
  - If CMake cannot locate your external zlib installation, set this to the directory containing the zlib headers (e.g. if you have /foobar/zlib.h, then set this to /foobar).
- ZLIB\_LIBRARY (If GEANT4\_USE\_SYSTEM\_ZLIB selected)
  - If CMake cannot locate your zlib installation, set this to the full path to the zlib library, e.g. /usr/lib/ libz.so
- GEANT4\_BUILD\_CXXSTD : (DEFAULT : c++98, Unix Only)
  - Compile Geant4 against given C++ standard ( c++98, including addenda, c++0x, or c++11. Geant4 is written in c++98, and you should use this option if your application requires support for the newer standards. If you set the variable to a standard the compiler does not support, and error will be emitted.

REQUIRES : GNU, Clang or Intel C++ compilers.

- GEANT4\_BUILD\_STORE\_TRAJECTORY : (DEFAULT : ON)
  - If set to ON, store trajectories in event processing. It can be switched to OFF to give a degree of performance improvement, but you will *not* be able to visualize events.
- GEANT4\_BUILD\_VERBOSE\_CODE : (DEFAULT : ON)
  - If set to ON, build Geant4 libraries with extra verbosity. It can be switched to OFF to give a degree of performance improvement, but you will not have as much information output should you run into problems or need to debug.
- GEANT4\_BUILD\_EXAMPLES : (DEFAULT : OFF)
  - If set to ON, build all Geant4 example applications using current Geant4 build. *WARNING:* this option is for Geant4 system testing only and is not intended for use by users studying and working on the examples. *No support is, or will be, provided for user builds of the examples using this option.*
- GEANT4\_ENABLE\_TESTING : (DEFAULT : OFF)
  - If set to ON, build and run Geant4 testing suites. *WARNING:* this option is for Geant4 system testing only and is not intended for use by users. *No support is, or will be, provided for user builds with this option.*
- GEANT4\_USE\_USOLIDS : (DEFAULT : OFF)
  - If set to ON, build Geant4 with support for USolids in geometry. *WARNING:* the use of USolids is experimental and should be used with caution
- GEANT4\_USE\_WT : (DEFAULT : OFF)
  - If set to ON, build Geant4 Wt web based visualization driver. *WARNING:* this driver is experimental and should be used with caution.

REQUIRES : Wt libraries and headers, OpenGL libraries and headers, Boost headers and Boost signals library.

#### 2.3.3. Selecting a Different Compiler and Changing Flags

CMake will, by default, select the first C and C++ compilers it finds in your path. Geant4's CMake scripts configure a default set of flags based on the compiler identity, as follows

- GNU Compiler Collection
  - CMAKE\_CXX\_FLAGS : -W -Wall -pedantic -Wno-non-virtual-dtor -Wno-long-long -Wwrite-strings -Wpointer-arith -Woverloaded-virtual -pipe
  - CMAKE\_CXX\_FLAGS\_RELEASE : -02 -DNDEBUG
  - CMAKE\_CXX\_FLAGS\_DEBUG : -g
  - CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO : -02 -g
- Clang
  - CMAKE\_CXX\_FLAGS : -W -Wall -pedantic -Wno-non-virtual-dtor -Wno-longlong -Wwrite-strings -Wpointer-arith -Woverloaded-virtual -pipe -Qunusedarguments
  - CMAKE\_CXX\_FLAGS\_RELEASE : -02 -DNDEBUG
  - CMAKE\_CXX\_FLAGS\_DEBUG : -g
  - CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO : -O2 -g
- Microsoft Visual C++
  - CMAKE\_CXX\_FLAGS : -GR -EHsc -Zm200 -nologo -D\_CONSOLE -D\_WIN32 -DWIN32 -DOS -DXPNET -D\_CRT\_SECURE\_NO\_DEPRECATE
  - CMAKE\_CXX\_FLAGS\_RELEASE : -MD -O2 -DNDEBUG
  - CMAKE\_CXX\_FLAGS\_DEBUG : -MDd -Od -Zi
  - CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO : -MD -O2 -Zi
- Intel C and C++ Compilers
  - CMAKE\_CXX\_FLAGS : -w1 -Wno-non-virtual-dtor -Wpointer-arith -Wwritestrings -fp-model precise
  - CMAKE\_CXX\_FLAGS\_RELEASE : -O2 -DNDEBUG
  - CMAKE\_CXX\_FLAGS\_DEBUG : -g
  - CMAKE\_CXX\_FLAGS\_RELWITHDEBINFO : -02 -g

For the GNU, Clang and Intel compilers, an additional flag selecting the C++ standard to compile against will be set. By default, this will use the c++98 standard. This can be changed if the compiler version supports it by setting the GEANT4\_BUILD\_CXXSTD to the required standard, as described in Section 2.3.2.

When Geant4 is built with support for multithreading (GEANT4\_BUILD\_MULTITHREADED set to ON), the following additional flags are added to all build types for the GNU, Clang and Intel compilers:

• -DG4MULTITHREADED -ftls-model=initial-exec -pthread

Please note that at the current time, multithreading support is not available on Windows platforms.

If you are using an unsupported or unrecognized (by Geant4) compiler, CMake will default to a standard and very simple set of flags for that compiler. We strongly recommend that you use the default compiler and flags, but both can be modified if your use case requires it. To specify the C and C++ compilers to be used, you can set the CC and CXX variables

```
... assuming clang/clang++ are in the PATH ...
```

```
$ CC=clang CXX=clang++ cmake <otherargs>
... or ...
$ export CC=clang
$ export CXX=clang++
$ cmake <otherargs>
```

You can also specify a full path should the compilers not be in the PATH. You can also specify the C and C++ compilers via the CMAKE\_<LANG>\_COMPILER options:

\$ cmake -DCMAKE\_C\_COMPILER=clang -DCMAKE\_CXX\_COMPILER=clang++ <otherargs>

Use of CMAKE\_<LANG>\_COMPILER will take precedence over any setting of CC or CXX in the environment or on the command line.

Whilst you *can* change the compiler after an initial configuration with CMake, it is not recommended as you may need to reset some variables by hand. If you do perform this step, it can only be done by rerunning CMake and passing the new compiler via the CMAKE\_<LANG>\_COMPILER argument(s), as the CC and CXX variables have no effect on subsequent runs of CMake in a given build directory. You may also need to remove the CMakeCache.txt file from the build directory before running CMake again. If you are building Geant4 using several compilers and/or versions, we strongly recommend creating one build directory per compiler system. Whilst this takes extra disk space, it provides a clean separation between different builds and also allows fast incremental builds against a single source directory.

Compiler flags can be interactively modified through the ccmake and CMake GUI interfaces. As compiler flags are an advanced option, you will need to activate viewing of advanced options. You may then edit the flags as you wish.

CMake is also aware of the CFLAGS and CXXFLAGS variables, so you may set these on the command line or as environment variables. However, note that this will only *prepend* extra flags to the default CMAKE\_<LANG>\_FLAGS.

If you need to completely change the compiler flags, then you can set CMAKE\_<LANG>\_FLAGS directly as a -D option to CMake. This will override all defaults set by Geant4's CMake scripts.

#### 2.3.4. Using an Initial Cache File for Build Options

As Geant4, and CMake in general, has many configurable options, remembering and typing out the CMake command line can be tedious and potentially error prone once you start to use a significant number of options. To ease this task and provide reproducible builds, you can write options as CMake set() commands into a so-called initial cache script. For example, to select Clang as the compiler and enable Qt support, we could write the following

```
set(CMAKE_C_COMPILER clang CACHE STRING "")
set(CMAKE_CXX_COMPILER clang++ CACHE STRING "")
set(GEANT4_USE_QT ON CACHE BOOL "")
```

into a file, say, mysettings.cmake. We could then pass this file to CMake to configure the Geant4 build with these settings:

\$ cmake -C /home/mysettings.cmake /path/to/geant4.10.00

Any settings in the supplied script will take priority over the defaults, so this can be a useful way to manage different builds in a reproducible way. Note that the set() commands must use the CACHE argument to ensure they are loaded into the CMake cache.

# Chapter 3. Setting Up and Using an Install of Geant4

#### 3.1. Geant4 Installation Locations

If you choose the default installation paths, then your install of Geant4 is completely contained under the directory you chose for CMAKE\_INSTALL\_PREFIX, with the structure

```
+- CMAKE_INSTALL_PREFIX
   +- bin/
     +- geant4.csh (UNIA CALL)
-- geant4.csh (UNIA CALL)
-- of (UNIX ONLY)
                         (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/
         +- Geant4Config.cmake
         +- Geant4ConfigVersion.cmake
         +- Geant4LibraryDepends.cmake
         +- Geant4LibraryDepends-Release.cmake
         +- UseGeant4.cmake
         +- Linux-q++
                          (OR Darwin-g++ UNIX ONLY SOFTLINK -> ..)
         +- Modules/
   +- share
      +- Geant4-10.0.0
                          (IF GEANT4_INSTALL_DATA WAS SET)
         +- data/
         +- examples/
         +- geant4make/
            +- geant4make.csh
            +- geant4make.sh
            +- config/
```

If you wish to make the Geant4 binaries and libraries available via your PATH and library path (LD\_LIBRARY\_PATH on Linux, DYLD\_LIBRARY\_PATH on Mac OS X), together with default environment variables for locating datasets, you should source the relevant script in CMAKE\_INSTALL\_PREFIX/bin

On interactive bourne shells (e.g. bash), do (assuming you are in CMAKE\_INSTALL\_PREFIX/bin):

\$ . geant4.sh

This command can also be used to setup the Geant4 environment in other Bourne shell scripts. You can also supply the full path to the script rather than changing to the directory containing it.

On interactive C shells, do (assuming you are in CMAKE\_INSTALL\_PREFIX/bin):

\$ source geant4.csh

In an interactive session you can also supply the full path to the script rather than changing to the directory containing it. The C shell script cannot be sourced directly inside other shell scripts due to a limitation of the C shell which prevents the script being able to locate itself. If you need to source the C shell script inside another, then you can use the command

cd CMAKE\_INSTALL\_PREFIX/bin ; source geant4.csh

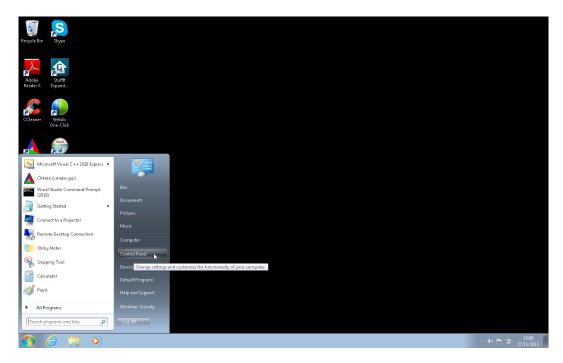
where you should replace <code>CMAKE\_INSTALL\_PREFIX/bin</code> with the directory you installed <code>geant4.csh</code> in. You can also use the command

```
source CMAKE_INSTALL_PREFIX/bin/geant4.csh CMAKE_INSTALL_PREFIX/bin
```

where as above you should replace CMAKE\_INSTALL\_PREFIX/bin with the directory where geant4.csh is located.

On Windows, you should add the directory containing the Geant4 dll files to your PATH environment variable. On Windows 7/XP, this can be done via the Control Panel as follows

1. Open the Windows Control Panel.



2. Open the *System* item in the Control Panel.

Recycle Bin	Skype				
Adobe Reader X	Stuffit Expand	🕢 - 💽 + Control Panel + All Con	trol Panel Items 🕨	• 47 Search Control Panel D	
		Adjust your computer's settings		View by: Small icons 💌	
CCleaner	WebEx One-Click	🏲 Action Center	🐑 Administrative Tools	🛃 AutoPlay	
	NUMBER OF STREET	Backup and Restore	Real BitLocker Drive Encryption	🔽 Color Management	
FI		Credential Manager	🔮 Date and Time	🐨 Default Programs	
CMake	NSIS	📑 Desktop Gadgets	🚔 Device Manager	Devices and Printers	
(cmake-gui)		💻 Display	🕒 Ease of Access Center	Flash Player (32-bit)	
		Folder Options	Fonts	🔠 Getting Started	
		📢 HomeGroup	🚑 Indexing Options	😥 Internet Options	
3		勴 Java (32-bit)	🕮 Keyboard	🖾 Location and Other Sensors	
Cygwin		Mail (32-bit)		Network and Sharing Center	
		Real Construction Area Icons	😹 NVIDIA Control Panel	🐻 Parental Controls	
		Performance Information and Tools	Personalization	Phone and Modem	
5		Power Options	Programs and Features	P Recovery	
Mozilla Firefox		🔊 Region and Language	log RemoteApp and Desktop Connections	4 Sound	
Firefox		Speech Recognition	Sync Center	1 System	
		🔔 Taskbar and Start Menu	Troubleshooting	& User A System	
		📑 Windows CardSpace	Idd Windows Defender	Winde View information about your computer, and change settings for	
OpenOffice		a Windows Update		hardware, performance, and remote	
4.0.0				connections.	
$\sim$					
PDFCreator					
roreator					
<b>(</b>	🥝 📋 🖸 🛯	<b>;</b>			▲ ●》 Image: ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

3. Click on the *Advanced system settings* link on the System window (on *Windows XP*, click on the *Advanced* tab).

	_						_
Recycle Bin	Skype						
Adobe Reader X	Stuffit Expand	💽 💌 🏽 🕨 Control Panel 🕨	All Control Panel Items 🔸 System	Search Control Panel	X		
CCleaner	WebEx One-Click	Control Panel Home Device Manager Remote settings System protection Advanced system settings	View basic information about your computer Windows edition Windows 7 Enterprise Copyright © 2009 Microsoft Corporation. All rights reserved. Service Pack 1	<b>~</b>			
CMake (cmake-gui) R Cygwin	NSIS		System Rating: Processor Installed menory (RAM): System type Pen and Touch:		. III		
Mozilla Firefox		See also Action Center Windows Update Performance Information and Tools	Computer name, domain, and workgroup settings Computer name. Full computer name. Computer deciption: Workgroup: Windows activation	Change settings			
OpenOffice 4.0.0			Windows is activated	ىلى مەلەر	·		
🌏	2					▲ ● ► 15: 27/11/	

4. Click on the *Environment Variables* button in the System Properties window.

Recycle Bin	Skype						
Adobe Reader X	Stuffit Expand	🚱 🔍 💌 🌁 🕨 Control Panel 🕨	All Control Pane	elitems > System - 4) Se	arch Control Panel	2	
CCleaner	WebEx One-Click	Control Panel Home P Device Manager Remote settings System protection	View basi Windows e Window Copyrig	Computer Name Hardware Advanced System Protection Remote		-	
CMake (cmake-gui)	NES	Advanced system settings	Service	You must be logged on as an Administrator to make most of these charges Performance Visual effects, processor scheduling, memory usage, and visual memory Settings			
Cygwin			Rating: Process Installed System Pen and	Settings		н	
Mozilla Firefox		See also	Computer r Compu Full con	System statup, system falure, and debugging information Settings	Change settings		
OpenOffice 4.0.0		Action Center Windows Update Performance Information and Tools	Compu Workgn Windows as Window	Environment Vargebes		•	
PDFCrestor							
<b>3</b>	6 📋 0 🛛	ș 🔌					▲ ♦> ● 15:19 27/11/2013

5. Select the PATH entry in the *User variables* list, and click the *Edit* button. If PATH is not present, click on the *New* and create it.

#### Setting Up and Using an Install of Geant4

sign average and a state of the		
Adobe Stuffit Reader X Stuffit	All Control Panel Items > System + 4 Search Control Panel Denie	
CCleaner WebEr Ore-Olick @ Device Manager @ Device Manager @ Renote settings @ System protection	View basic information about your computer Windows ef Window Corpute Corpute Hadware Advanced System Protection Rende	
CMike (rmike-gui)	Service Environment Variables User variables for Ben Variable Value Valu	
Cygwin	Process Installee System Pen and System variables Vorbide Vorbide Vorbide Vorbide	
Mozila Firefox See also Action Center	Compute Condect Cl/Wholows/system32/cnd.exe Compute Condect Cl/Wholows/system32/cnd.exe Compute Change settings Full corr Nutresc, C.F. No Nutresc, C.F. 2 Comput OS Windows_NT	
Windows Update Performance Information and CopenOffice 4.0.0	Windows a OK Cancel	
PDFCreator		
🚱 🍐 🚞 🔍 🐺 😣		▲ (*) ● 15:20 27/11/2013

6. In the popup *Edit User Variable* window, append the directory in which the Geant4 dlls are installed to the Variable value entry of the PATH variable (Note that on Windows, path entries are separated by semicolons). It's very important to append the Geant4 dll path if you have an existing PATH, otherwise other programs may stop working correctly! If the Variable value entry of the PATH variable is empty, or you've just created it, you can simply set the value to the directory in which the Geant4 dlls are installed. Once you have edited, click *OK*.

Retycle Bin Stype		
Adole Stufft Rester X Symid.	and + All Control Panel Rems + System	
CCleaner WebEr Orthonologie Remote settings	View basic information about your computer Windows edition W System Properties	
System protection     System protection     Advanced system settings     CMake     NXSS (cmake-gou)	ps Se Computer Name   Hardware   Advanced System Protection Remote	
Cygwin	Ra PATH C:Ulurer/Ben/Document/genet T29 %USERROPCHEVS/AppData)(cod/temp Prt 119 %USERROPCHEVS/AppData)(cod/temp Int Syr Edit:UserVariable	
Mozila Firefox See also Action Center	Comp Co Variable name: PATH Variable value:	
Corrottice 48.8		
POFCreator		- () (* 15:10 27/11/2013

#### 3.2. Building Applications with Geant4

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. The details of how to implement source code for an application are described in detail in the Geant4 User's Guide for Application Developers. Here, we describe how you can build your sources into an application and compile and link it against Geant4.

We provide three main tools to help with building applications: a CMake "Geant4Config.cmake" config file, a GNUMake module "binmake.gmk" and a UNIX-only command line program "geant4-config". The following sections give an overview of each tool and how to use them to build a simple application.

# 3.2.1. Using CMake to build Applications: Geant4Config.cmake

Geant4 installs a file named Geant4Config.cmake located in:

+- CMAKE\_INSTALL\_PREFIX +- lib/ +- Geant4-10.0.0/ +- Geant4Config.cmake

which is designed for use with the CMake scripting language find\_package command. Building a Geant4 application using CMake therefore involves writing a CMake script CMakeLists.txt using this and other CMake commands to locate Geant4 and describe the build of your application against it. Whilst it requires a bit of effort to write the script, CMake provides a very powerful and flexible tool, especially if you are working on multiple platforms. It is therefore the method we recommend for building Geant4 applications.

We'll use Basic Example B1, which you may find in the Geant4 source directory under examples/basic/B1, to demonstrate the use of CMake to build a Geant4 application. You'll find links to the latest CMake documentation for the commands used throughout, so please follow these for further information. The application sources and scripts are arranged in the following directory structure:

```
+- 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. This arrangement of source files is not mandatory when building with CMake, apart from the location of the CMakeLists.txt file in the root directory of the application.

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

```
# (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})
# (б)
```

```
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)
```

For clarity, the above listing has stripped out the main comments (CMake comments begin with a "#") you'll find in the actual file to highlight each distinct task:

1. Basic Configuration

The cmake\_minimum\_required command simply ensures we're using a suitable version of CMake. The project command sets the name of the project and enables and configures C and C++ compilers.

2. Find and Configure Geant4

The aforementioned find\_package command is used to locate and configure Geant4 (we'll see how to specify the location later when we run CMake), the REQUIRED argument being supplied so that CMake will fail with an error if it cannot find Geant4. The option command specifies a boolean variable which defaults to ON, and which can be set when running CMake via a -D command line argument, or toggled in the CMake GUI interfaces. We wrap the calls to find\_package in a conditional block on the option value. This allows us to configure the use of Geant4 UI and Visualization drivers by exampleB1 via the ui\_all vis\_all "component" arguments to find\_package . These components and their usage is described later.

3. Configure the Project to Use Geant4 and B1 Headers

To automatically configure the header path, compiler flags and compiler definitions needed for linking to Geant4, we use the include command to load a CMake script supplied by Geant4. The CMake variable named Geant4\_USE\_FILE is set to the path to this module when Geant4 is located by find\_package . We use the include\_directories command to add the B1 header directory to the compiler's header search path. The CMake variable PROJECT\_SOURCE\_DIR points to the top level directory of the project and is set by the earlier call to the project command.

4. List the Sources to Build the Application

Use the globbing functionality of the file command to prepare lists of the B1 source and header files.

Note however that CMake globbing *is only used here as a convenience*. The expansion of the glob only happens when CMake is run, so if you later add or remove files, the generated build scripts will not know a change has taken place. Kitware strongly recommend listing sources explicitly as CMake automatically makes the build depend on the CMakeLists.txt file. This means that if you explicitly list the sources in CMakeLists.txt, any changes you make will be automatically picked when you rebuild. This is most useful when you are working on a project with sources under version control and multiple contributors.

5. Define and Link the Executable

The add\_executable command defines the build of an application, outputting an executable named by its first argument, with the sources following. Note that we add the headers to the list of sources so that they will appear in IDEs like Xcode.

After adding the executable, we use the target\_link\_libraries command to link it with the Geant4 libraries. The Geant4\_LIBRARIES variable is set by find\_package when Geant4 is located, and is a list of all the libraries needed to link against to use Geant4.

6. Copy any Runtime Scripts to the Build Directory

Because we want to support out of source builds so that we won't mix CMake generated files with our actual sources, we copy any scripts used by the B1 application to the build directory. We use foreach to loop over the list of scripts we constructed, and configure\_file to perform the actual copy.

Here, the CMake variable **PROJECT\_BINARY\_DIR** is set by the earlier call to the project command and points to the directory where we run CMake to configure the build.

7. If Required, Install the Executable

Use the install command to create an install target that will install the executable to a bin directory under CMAKE\_INSTALL\_PREFIX.

If you don't intend your application to be installable, i.e. you only want to use it locally when built, you can leave this out.

This sequence of commands is the most basic needed to compile and link an application with Geant4, and is easily extendable to more involved use cases such as using other third party packages (via find\_package) or platform specific configuration.

With the CMake script in place, using it to build an application is a two step process. First CMake is run to generate buildscripts to describe the build. By default, these will be Makefiles on Unix platforms, and Visual Studio solutions on Windows, but you can generate scripts for other tools like Xcode and Eclipse if you wish. Second, the buildscripts are run by the chosen build tool to compile and link the application.

A key concept with CMake is that we generate the buildscripts and run the build in a separate directory, the socalled *build directory*, from the directory in which the sources reside, the so-called *source directory*. This is the exact same technique we used when building Geant4 itself. Whilst this may seem awkward to begin with, it is a very useful technique to employ. It prevents mixing of CMake generated files with those of your application, and allows you to have multiple builds against a single source without having to clean up, reconfigure and rebuild.

We'll illustrate this configure and build process on Linux/Mac using Makefiles, and on Windows using Visual Studio. The example script and Geant4's Geant4Config.cmake script are vanilla CMake, so you should be able to use other Generators (such as Xcode and Eclipse) without issue.

#### 3.2.1.1. Building ExampleB1 with CMake on Unix with Makefiles

We'll assume, *for illustration only*, that you've copied the exampleB1 sources into a directory under your home area so that we have

```
+- /home/you/B1/
+- CMakeLists.txt
+- exampleB1.cc
+- include/
+- src/
+- ...
```

Here, our *source directory* is /home/you/B1, in other words the directory holding the CMakeLists.txt file.

Let's also assume that you have already installed Geant4 in your home area under, *for illustration only*, /home/you/geant4-install.

Our first step is to create a *build directory* in which build the example. We will create this alongside our B1 *source directory* as follows:

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

We now change to this *build directory* and run CMake to generate the Makefiles needed to build the B1 application. We pass CMake two arguments:

```
$ cd $HOME/B1-build
$ cmake -DGeant4_DIR=/home/you/geant4-install/lib64/Geant4-10.0.0 $HOME/B1
```

Here, the first argument points CMake to our install of Geant4. Specifically, it is the directory holding the Geant4Config.cmake file that Geant4 installs to help CMake find and use Geant4. You should of course adapt the value of this variable to the location of your actual Geant4 install.

The second argument is the path to the *source directory* of the application we want to build. Here it's just the B1 directory as discussed earlier. You should of course adapt the value of that variable to where you copied the B1 source directory.

CMake will now run to configure the build and generate Makefiles. On Linux, you will see the output

```
$ 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++
-- Detecting CXX compiler ABI info
-- Detecting done
-- Generating done
-- Build files have been written to: /home/you/B1-build
```

On Mac OS X, you will see slightly different output, but the last three lines should be identical.

If you now list the contents of you build directory, you can see the files generated:

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

Note the Makefile and that all the scripts for running the exampleB1 application we're about to build have been copied across. With the Makefile available, we can now build by simply running make:

\$ make -jN

CMake generated Makefiles support parallel builds, so can set N suitable for the number of cores on your machine (e.g. on a dual core processor, you could set N to 2). When make runs, you should see the output

```
$ make
Scanning dependencies of target exampleB1
[ 16%] Building CXX object CMakeFiles/exampleB1.dir/exampleB1.cc.o
[ 33%] Building CXX object CMakeFiles/exampleB1.dir/src/B1PrimaryGeneratorAction.cc.o
[ 56%] 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/B1DetectorConstruction.cc.o
[ 100%] Building CXX object CMakeFiles/exampleB1.dir/src/B1DetectorConstruction.cc.o
[ 100%] Building CXX object CMakeFiles/exampleB1.dir/src/B1SteppingAction.cc.o
```

CMake Unix Makefiles are quite terse, but you can make them more verbose by adding the VERBOSE argument to make:

\$ make VERBOSE=1

If you now list the contents of your *build directory* you will see the exampleB1 application executable has been created:

\$ 1sCMakeCache.txtexampleB1init.macrun1.macCMakeFilesexampleB1.ininit\_vis.macrun2.maccmake\_install.cmakeexampleB1.outMakefilevis.mac

You can now run the application in place:

\$ ./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 <<< Reference Physics List OBBC Checking overlaps for volume Envelope ... OK! Checking overlaps for volume Shape1 ... OK! Checking overlaps for volume Shape2 ... OK! User-derived detector construction class does not implement ConstructSDandFiled method: i.e. workers will not have SD and fields! The user can safely ignore this message if (s)he has no sensitive detector or field in her/his application. ### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0 Visualization Manager instantiating with verbosity "warnings (3)"... Visualization Manager initialising.. Registering graphics systems ...

Note that the exact output shown will depend on how both Geant4 and your application were configured. Further output and behaviour beyond the Registering graphics systems... line will depend on what UI and Visualization drivers your Geant4 install supports. If you recall the use of the ui\_all vis\_all in the find\_package command, this results in all available UI and Visualization drivers being activated in your application. If you didn't want any UI or Visualization, you could rerun CMake as:

\$ cmake -DWITH\_GEANT4\_UIVIS=OFF -DGeant4\_DIR=/home/you/geant4-install/lib64/Geant4-10.0.0 \$HOME/B1

This would switch the option we set up to false, and result in find\_package not activating any UI or Visualization for the application. You can easily adapt this pattern to provide options for your application such as additional components or features.

Once the build is configured, you can edit code for the application in its *source directory*. You only need to rerun make in the corresponding *build directory* to pick up and compile the changes. However, note that due to the use of CMake globbing to create the source file list, if you add or remove files, you need to rerun CMake to pick up the changes! This is another reason why Kitware recommend listing the sources explicitly.

## **3.2.1.2.** Building ExampleB1 with CMake on Windows with Visual Studio

We'll assume, for illustration only, that you've copied the exampleB1 sources into a directory

```
C:\Users\Ben\Documents\examples\basic\B1
```

Let's now use the CMake GUI to configure the build and generate a Visual Studio solution for it, and then use Visual Studio to build the application from this solution. This process involves several steps, which we break down as follows.

1. If you browse the contents of the source directory, you can see that it contains the CMakeLists.txt file. So our *source directory* is C:\Users\Ben\Documents\examples\basic\B1, in other words the directory holding the CMakeLists.txt file.

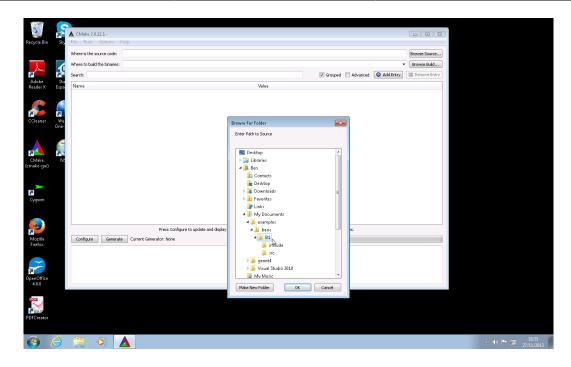
le Edit View Tools Help					
Organize 👻 Share with 👻	New folder				II • 🔟 (
🔆 Favorites	Documents library				Arrange by: Folder -
🧮 Desktop	81				Arrange by: Folder *
〕 Downloads	Name	Date modified	Туре	Size	
🔤 Recent Places				0440	
	🎥 include	27/11/2013 15:54	File folder		
🗃 Libraries	鷆 src	27/11/2013 15:54	File folder		
a 📑 Documents	.README	26/11/2013 11:28	README File	8 KB	
My Documents	📄 CMakeLists	26/11/2013 11:28	Text Document	3 KB	
4 🎍 examples	🐏 exampleB1	26/11/2013 11:28	C++ Source	5 KB	
🚛 🚵 basic	exampleB1.in	13/11/2013 09:45	IN File	1 KB	
4 🍌 Bj.,	exampleB1.out	13/11/2013 09:45	OUT File	30 KB	
mclude	GNUmakefile	26/11/2013 11:28	File	1 KB	
src	History	13/11/2013 09:45	File	3 KB	
> 🏭 geant4	init.mac	05/11/2012 11:37	MAC File	1 KB	
Visual Studio 2010	init_vis.mac	05/11/2012 11:37	MAC File	1 KB	
Public Documents	README	26/11/2013 11:28	File	8 KB	
Music	run1.mac	05/11/2012 11:37	MAC File	1 KB	
Fictures	run2.mac	13/11/2013 09:45	MAC File	1 KB	
Fictures Videos	vis.mac	05/11/2012 11:37	MAC File	4 KB	
P 🛅 Videos					
💺 Computer					
😂 🚣 Local Disk (C:)					
🙀 Network					
15 items					

2. Open the CMake (cmake-gui) executable, and click on the *Browse Source*... button in the top right hand corner of the window.

Recycle Bin Sky	File Tools Options Help			
	Where is the source code:	Browse Source		
1		Browse Build		
Adobe Stu	Search: Grouped 🗌 Advanced 🔂 Add Entry	Remove Entry		
Adobe Stu Reader X Expa	Name Value			
CCleaner We One- CMake (cmske-gui)				
Cygwin				
	Press Configure to update and display new values in red, then press Generate to generate selected build files.			
Mozilla Firefox	Configure Generate Current Generator: None			
OpenOffice 4.0.0				
PDFCrestor				
🕘 🌔	🚆 🧿 🛕		▲ (*) I* II 16:33 27/11/2	

3. Use the file browser popup to locate the B1 *source directory*, and click *OK*.

Setting Up and Using an Install of Geant4



4. Now we create the *build directory* in which to create the Visual Studio project files and hold the build products. This directory should not be the same as, or inside, the *source directory*. Therefore, let's create this *build directory* alongside our source directory.

Click on the *Browse Build*... button in the top right hand side of the CMake GUI window. Use the file browser popup to browse back to C:\Users\Ben\Documents\examples\basic, and click on the *Make New Folder* button. Rename the created folder to *B1-build*, and click on the *OK* button. The two text entries at the top of the GUI should now contain C:/Users/Ben/Documents/examples/basic/B1 and C:/Users/Ben/Documents/examples/basic/B1-build respectively (*Note*: CMake always represents Windows paths with forward slashes).

Recycle Bin Sty Adobe Reader X	Criste 2.8.12.1-  File Tools Options Help  Where is the source code: C.j.Kers/Ben/Documents/examples/basic/B1  Where to build the binaries:  Search:  Name Value	Browse Source; Browse Source; Browse Bald 2 St Remove Entry	
CCleaner We One CMake (cmake-gui)	Browse For Folder		
Mozilla Firefox OpenOffice 4.0.0	Press Configure to update and depty Configure Generator: None Generator Current Generator: None Main New Folde Main New Folde K. Cancel		
PDFCrestor			_ () <b>P</b> (3) 16:36 27/11/29

5. Before we can configure the project, we need to tell CMake where to find Geant4. To do this, click on the *Add Entry* button in the top right corner of the CMake GUI.

5		rs/Ben/Documents/examples/basic/B1-build		
	File Tools Options H			
	Where is the source code:	C:/Users/Ben/Documents/examples/basic/B1	Browse Source	
	Where to build the binaries:	C:/Users/Ben/Documents/examples/basic/81-build	Browse Build	
	Search:	🖉 Grouped 🔲 Advanced	Add Entry 🗱 Remove Entry	
Adobe Stu Reader X Expa	Name	Value	Add New Entry	
CCleaner We One-				
CMake N: (cmake-gui)				
Cygwin				
		Press Configure to update and display new values in red, then press Generate to generate selected build files.		
Mozilla Firefox	Configure Generate	e Current Generator: None		
Firefox				
OpenOffice 4.0.0				
PDFCrestor				
📀 <i>(</i>		Δ		▲ ● ► 16:37 27/11/2013

6. In the *Add Cache Entry* window that pops up, set the *Name* of the entry to *Geant4\_DIR*. Set the *Type* to *PATH*.

Recycle Bin		CMake 2.8.12.1 - C:/Use File Tools Options I	ers/Ben/Documents/examples/ Help	'basic/B1-build							X		
5			C:/Users/Ben/Documents/exam C:/Users/Ben/Documents/exam		id				•	Browse Source Browse Build			
<b>7</b>		Search:					Grouped	Advanced	💠 Add Entry	S Remove Enl	ry		
Adobe Reader X	Stu Expa	Name				Value							
CCleaner CCleaner CMake (cmake-gui)	We One- NS			Value:	Geant4_DIR BOOL BOOL PATH ETI EPATH	ß	•						
Cygwin						ок	Cancel						
Mozilla Firefox				gure to update ar	nd display new values in	red, then press Generate b	o generate selected bu	ild files.					
Mozilla Firefox		Configure Generab	e Current Generator: None								- 1		
OpenOffice 4.0.0													
PDFCrestor													
<b>(</b>	3		<u>A</u>									- • P 1	16:38 27/11/2013

7. In the *Add Cache Entry* window, click the browse button on the right hand side of the *Value* to pop up the *Browse For Folder* window. Browse to find your install of Geant4, and select the folder which contains the Geant4Config.cmake file. This is generally located in the lib/Geant4-X.Y.Z (where X.Y.Z is the Geant4 version number) folder inside the directory in which you installed Geant4.

#### Setting Up and Using an Install of Geant4

Rec	ycle Bin	Sky	CMske 28.12.1 - C/User/Ben/Documents/examples/basic/B1-build CMske 28.12.1 - C/User/Ben/Docum	
	X.	1	miners of source does - cyberge outcomine system projects, par where to build the binaries: Cyberge outcomine system projects, par where to build the binaries: Cyberge outcoments/example/basige/build Search: Ø Grouped @ Advanced @ Advanced @ Advanced @ Advanced @ Advanced for an outcomentary	
A	idobe ader X	Stu Expa		
~	<b>S</b> leaner	<b>We</b>		
		One-		
C (cm	Make ake-gui)	N	Description:	
0	ygwin		A grant4.11.0-initall bin bin bin bin bin bin bin bin	
	e lozilla irefox		Press Configure to update and dopley Configure Generate Current Generator None	
Ope	enOffice		My Music My Pictures My Videos	
2	Creator			
6	9	Ø		▲ ●》 ●● 16:40 27/11/2013

8. With *Geant4\_DIR* set, click on the *Configure* button in the bottom left hand corner of the GUI.

5	9					
	7		s/Ben/Documents/examples/basic/B1-build			
Recycle Bin	Sky	File Tools Options I	elp			
		Where is the source code:	C:/Users/Ben/Documents/examples/basic/B1		Browse Source	
Å	6	Where to build the binaries:	C:/Users/Ben/Documents/examples/basic/B1-build	-	Browse Build	
	d'	Search:		🕼 Grouped 📃 Advanced 🗳 Add Entry	38 Remove Entry	
Adobe Reader X	Stu Expa	Name	Value			
	e-p-v	Geant4_DIR	C:/Users/I	len/Documents/geant4/geant4.10.0-install/lib/Geant4-10.0.0		
CCleaner	We One-					
CMake						
(cmake-gui)						
Cygwin						
			Press Configure to update and display new values in red, then pres	s Generate to generate selected build files.		
Mozilla Firefox		Configure Generab	Current Generator: None			
Firefox						
OpenOffice 4.0.0						
PDFCreator						
<b>1</b>	8	📋 O 🛛				▲ ◆ ► 16:41 27/11/2013

9. In the pop up window, select *Visual Studio 10* (or 9 if you have that version installed) and ensure the *Use default native compilers* radio button is ticked. Click on the *Finish* button. CMake will run to check that your system can perform the build and that Geant4 can be found. It will output information on its status and any errors in the logging window at the bottom of the GUI.

Clearer Construction   Construction Construction	Recycle Bin Sky	CMske 2.8.12.1 - C/User/Ben/Documents/examples/basic// File Tools Options Help Where is the source code: C:/Users/Ben/Documents/examples/bas Where to build the binaries: C:/Users/Ben/Documents/examples/ba	sic/B1	Brows Sauro Rows Sauro	
Center   Construction	Adobe Stu	Name			
CMARE       N         CMARE       N         CMARE       N         Constrained       Stacky native completes         Stacky native completes       Stacky native completes         Press Configure to up       Press Configure to up         Configure       Generate         Connext       Generate         Conne       Generate<	CCleaner We		Specify the generator for this project	9	
Mootilis Friefox     Configure     Connect Generator: None       Cppuroffice     Image: Connect Generator: None	(cmake-gui)		Use default native complers     Specify native complers     Specify toolchain file for cross-compling		
	Mozilla Firefox		ur		
	OpenOffice 4.0.0				

10. After a successful configuration, you will see *Configuring done* in the logging window at the bottom, and some red highlighted entries in the main window, including one for *CMake*. The red color does not always mean there has been an error, but denotes that CMake needs to rerun to resolve options it has found in the build. Simply reclick the *Configure* button to rerun CMake.

						_	
🥶 💦		ers/Ben/Documents/examples/basic/B1-build					
cycle Bin Sky	File Tools Options	Help					
	Where is the source code:	C:/Users/Ben/Documents/examples/basic/B1				Browse Source	
1	Where to build the binaries:	C:/Users/Ben/Documents/examples/basic/B1-build			•	Browse Build	
	Search:			Grouped 📃 Advanced	💠 Add Entry	38 Remove Entry	
Adobe Stu eaderX Expa			Value				
	<ul> <li>Ungrouped Entries</li> <li>CMAKE</li> </ul>						
	Ungrouped Entries						
<b>1</b>							
Cleaner We One-							
:Make N: iake-gui)							
g,							
_							
4							
ygwin							
9		Press Configure to update and display new	values in red, then press Generate to gener	ate selected build files.			
Aozilla irefox	Configure Generat	e Current Generator: Visual Studio 10					
Inclux.	Detecting C compile Detecting C compile	er ABI info er ABI info - done				*	
~	Check for working (	XXX compiler using: Visual Studio 10 XXX compiler using: Visual Studio 10 works					
	Detecting CXX compi	iler ABI info				E	
enOffice 4.0.0	Configuring done	iler ABI info - done				-	
4.0.0							
Ciou ta							
FCreator							
		A					- () 🕨 🎁 16:
9 😏							- 10 - 10 27/11/

11. After a further successful configuration, you will see *Configuring done* in the logging window at the bottom, and there should be no red colored entries in the main window. Now click on the *Generate* button to create the Visual Studio solution.

		n/Documents/examples/basic/B1-build				
Recycle Bin Sky		v/bocuments/examples/basic/b1-build				
,		sers/Ben/Documents/examples/basic/B1			Browse Source	
		Isers/Ben/Documents/examples/basic/81-build			Browse Build	
		sers/beil/bocomencs/examples/basic/b1-bolio		🗸 Grouped 📃 Advanced 🙀		
Adobe Stu	Search:			Grouped C Advanced	Add Entry 36 Remove Entry	
Reader X Expa	Name  Ungrouped Entries		Value			
	CMAKE					
2						
CCleaner We						
One-						
CMake N: (cmake-gui)						
(cmake-gui)						
Cygwin						
Mozilla Firefox		Press Configure to update and disp	play new values in red, then press Generate to g	enerate selected build files.		
Mozilla	Configure Generate G	Eurrent Generator: Visual Studio 10				
FILETOX	Configuring done					
OpenOffice 4.0.0						
4.0.0						
1000						
PDFCreator						
	📋 🖸 🔺					- <b>↓</b> ≫ 📭 🎁 16 27/11
						2//11

12. After generation, you should see *Configuring done, Generating done* in the logging window at the bottom of the GUI. CMake has run successfully, and a Visual Studio solution has been created. You can now exit the CMake GUI.

	9														_										
Recycle Bin		A CMake 2.8.12.1 - C:/Use File Tools Options			ocument	s/examples,	/basic/B1	1-build											E						
Recycle Bin	- 245																			- )					
		Where is the source code:																		rowse Source					
Å.	-0	Where to build the binaries:	a C:/U	/User	s/Ben/Docu	ments/exam	aples/basic	.c/81-build	d						_				C	Browse Build					
Adobe Reader X	Stu	Search:													Groupe	ed 🔳	Advanced	骨 Add Entry		Remove Entry					
Reader X	Expa	Name Ungrouped Entries									Va	alue													
		CMAKE																							
CCleaner	<b>₹</b> We																								
	One-																								
	N																								
	7																								
CMake (cmake-gui)	N																								
(entake gai)																									
_																									
3																									
Cygwin																									
Mozilla Firefox		Configure Generat	** (	0.00		Press Confi ator: Visual S			id display n	iew value	ts in red, t	then press	Generate	to general	ce selecced b	iulia rile	s.								
Firefox			ו•	Cun		con visual s	000010																		
		Configuring done Generating done																							
OpenOffice 4.0.0																									
4.0.0		-				_	_	_																	
POR																									
Tiou ta																									
PDFCreator																									
<b>@</b>	e		A																			▲ 10	P (1)	16	i:4
	9																								

13. Now start up Visual Studio and choose *Open Project*. This guide is somewhat specific to Visual Studio 2010 Express, but the solution files should appear with the same names in both 2008 and 2010. Browse to your build directory and open the B1.sln Microsoft Visual Studio Solution file. It may take some time for Visual Studio to fully open the project and parse all the sources.

Setting Up and Using an Install of Geant4

😫 Start Page - Microsoft Visual C++ 2010 Express							- 🗗 🛙
File Edit View Debug Tools Window Help							
🖓 • 🕫 📓 📓 📓 👗 🛍 • 🕅 • 🖗 •	× -	- G4ParticleTable	• 🗠 🕾 🕾 🚺	÷			
四 出 👗 🖕							
🔣 Solution Explorer 🛛 👻 🕂 Start Page 😒	×						- 🎽
8 🖻							• Toolbox
	Visual C++ 2010 Express						b.
Document Outline	VISUAL C++ 2010 Express						
tline							
		Cat Charted Latent Nouro				)	
	(a) Open Project				×		
ේ ම	😋 🔾 🛡 📕 « Documents 🕨 My Doc	cuments + examples + basic + B1-build +	<ul> <li>✓ </li> <li>✓ </li></ul>	h B1-build	٩		
	Organize 👻 New folder			)III 🔹 🛅	0		
Recent	Microsoft Visual C++ 2010 Expre	Documents library				velopers quickly create exciting interactive express development environment, improved	
	Projects	B1-build		Arrange by: Folder -		idea to great application has never been easier.	
Close p		Name	Date modified	Type	Size	Center, or find the latest and coolest projects on	
Show :	🔶 Favorites	CMakeFiles	27/11/2013 16:44	File folder			
Output	Desktop	ALL_BUILD	27/11/2013 16:44	VC++ Project			<b>-</b> 4 ×
Show output	Downloads Recent Places	1 B1	27/11/2013 16:44	Microsoft Visual S			
	E	<b>3</b> 81	27/11/2013 16:44	VC++ Project			_
	🥽 Libraries	🞇 exampleB1	27/11/2013 16:44	VC++ Project			
	Documents	🐺 INSTALL 🎇 ZERO_CHECK	27/11/2013 16:44 27/11/2013 16:44	VC + + Project VC + + Project			
	My Documents	an zero_check	27/11/2013 10:44	VC++ Project			
	examples						
	B1						
	B1-build						
	🔑 CMakeFiles						
	🎉 geant4						
	Visual Studio 2010 Public Documents						
	-h Murie	-					
	* man	٠ ( m					
	File name: B1		→ All Proj	ject Files (*.sln;*.dsw;*.vc	•		
			Op	en 🔽 Cancel			
				ML.	d		
🚱 🦂 🚞 🖸 💽						- 🕪 🏴 📁	17:02

14. In the Solution Explorer, you can click on the exampleB1 to view the sources for the project.

B1 - Microsoft Visual C++ 2010 Express			- 8 🕰
File Edit View Project Debug To			
🔂 • 🔤 • 🥁 属 🥔 🔺 🕰 🔛			
🕮 🛲   🚠 🚽 🖾 🗞 🌭 🔺 🗏	9		
🔙 Solution Explorer 🛛 🔫 🛠	exampleB1.cc ×	-	Properties 👻 👻 🏏
ş 🗟 🔄 🗉	(Global Scope) •		- Te
Solution Eglorer 4 2 × Solution Bill Grapicsti Solution Bill Grapicsti Edit Source File Bill Source File Bill ActionInitialization Bill Bill Source File Bill Source File Bi	(dlobalScop)       *         79       //         71       //         72       //         73       //         74       //         75       //         74       //         75       //         76       //         77       //         78       //         79       //         70       //         71       //         72       //         73       //         74       //         75       /         76       /         77       GWModularPhysicsList*         78       //         79       /         70       /         71       //         71       //         72       /         73       //         70       /         71       /         71       /         71       /         71       /         72       /         73       /         74       /         70       /<		Properties • 4 × 27 Too Bay
K			
Ready		Ln1 Col1	
🚳 🥝 📜 🖸			<ul> <li>▲ ● ■ 17:15</li> <li>■ 27/11/2013</li> </ul>

15. Click on the Solution Configuration drop down, and change the configuration to *Release*. In general, it's best to build your application with the same configuration as your Geant4 installation. In this example, we built Geant4 in Release mode, so we build the B1 example in the same mode.

😫 B1 - Microsoft Visual C++ 2010 Express		- # <b>×</b>
File Edit View Project Debug Tools Window Help		
👔 - 🛅 - 😂 👦 🧭 🙏 🖕 九 - 🗠 - 🕨 Debug 🔹 Win32 🔹 🧭 G4ParticleTable 🔹 🔍 🕾 💥 💀 🔍 =		
御 告 A - I C Na Na A - I C 2 - Debug		
MinsizeRei		
Solution Explorer     * 1 ×     Relate     Relate       Completion Str (5 project)     (Global Scope Configuration Manager)     * Writing Manager)       J Subion Str (5 project)     70     //       J Alt BUID     71     //       J Subion Str (5 project)     72     runWanager->SetUserInitialization(new SlDetectorConstruction());       J Subion Str (5 project)     74     //		Properties 👻 🥆
See	-	• 10 10 10 10 10
Solution 'B1' (5 projects) 70 //	<b>±</b>	19 AL 🗉
Image: Second structure         71         // Detector construction           Image: Second structure         72         runManager->SetUserInitialization(new B1DetectorConstruction());	A	
P     Image: SetUserInitialization(new B1DetectorConstruction());       Image: SetUserInitialization(new B1DetectorConstruction());		
75 GevenourarPhysicsList PhysicsList = new QBBC;		
<pre>A Comparison of Compariso</pre>		
BIDetectorConstruct 78	12	
BiEventAction.hh 79 // User action initialization	5	
B1PrimaryGeneratory 80 runManager->SetUserInitialization(new B1ActionInitialization());		
BIRunAction.hh 82 ⊟ // Initialize 64 kernel BISteopingAction.hh 83 //		
<pre>84 runManager-&gt;Initialize();</pre>		
BIActionInitialization	*	
🕶 B1DetectorConstruct 100 % 👻 4	• •	
BIEventAction.cc Output	<b>-</b> ₽ ×	
🔁 B1PrimaryGenerator/ Show output from: Build - 🖓 🕼 🕸 👒 🖃		
C BIRunce Show output Home Build	*	
e Bikupacon.cc		
CMakeLists.tet		
INSTALL		
▶ 📓 ZERO_CHECK		
	*	
Ready	Ln 111 Col 7	Ch 7 INS
		47.20
		▲ (i) → 17:39 27/11/2013

16. To build the project, right click on ALL\_BUILD in the Solution Explorer and click Build in the dialog box.

	fisual C++ 2010 Express		- # <b>*</b>
	Project Debug Tools Window Help		
	🛃 🥥   🔏 🖧   10 - (14 -   🕨 Release	- Win32 - 🧭 G4ParticleTable - 🖓 🕾 🎘 🖬 🗟 🗸	
🖽 📇   🕹 🚽 l (	13 % % A*   I 2 2 .		
Solution Explore	er 👻 🕈 🗙 exampleB1.cc 🗙		👻 Properties 🔍 👻 🔆
g 🛅 🔯	(Global Scope)		ALL_BUILD Project Properties
Solution Export Carl Carl Carl Carl Carl Carl Carl Carl	B1 (5 project)         70         7/           UP         Build         7/           B2         Rebuild         7/           If and the second secon	or construction ->SetUserInitialization(new #IDetectorConstruction()); = list =Physicalist physicalist = new QBBC; =>SetUserInitialization(physicalist); ttion initialization =>SetUserInitialization(new #IActionInitialization()); lize 64 kernel =>Initialize(); = refe	Lagund Project Properties     Winney     Marcia     Marcia
Ready			• (1) (10 - 10 - 10 - 10 - 10 - 10 - 10 - 10

17. Visual Studio will perform the build, and report on progress in *Output*.

Setting Up and Using an Install of Geant4

B1 - Microsoft Visual C++ 2010 Express			- # <b>*</b>
File Edit View Project Debug To			
i 🛅 - 🖽 - 🧀 属 🕼 👗 🐚 选	🖅 - 🔍 - 🕨 Release - Win32 - 🧭 G4ParticleTable - 💀 🕾 🛃 🚽		
出 出 ] 三 シレ - 三	92		
Solution Explorer - 4 ×	exampleBLcc ×	+ Pro	nnerties 🔹 🗸 🗙
2 🔚 🐚	(Global Scope) - <sup>10</sup> main(int argc, char ** argv)		L_BUILD Project Properties
Solution Explorer + 4 × Solution 'B1' (5 projects) Solution 'B1' (5 projects) MALLBUID MA	(violad scope) • • • • main(int argc, cnar • argv)		
ALL BUILD	71 // Detector construction	A	
2 ▷ 38 B1	72 runManager->SetUserInitialization(new B1DetectorConstruction());	111	Misc
🗧 🍙 🎇 exampleB1	73 74 // Physics list		(Name) ALL_BUILD
External Dependencies	74 // mysics ist 75 G4WModularPhysicsList* physicsList = new OBBC:		Project Deper Project File C:\Users\Ben\Docu
Header Files	76 physicsList->SetVerboseLevel(1);		Root Namesp
<ul> <li>Source Files</li> <li>CMakeLists.bdt</li> </ul>	77 runManager->SetUserInitialization(physicsList);		Roocivarriesp
Distance Instance	78 79 // User action initialization	E	
ZERO CHECK	<pre>runWanger-&gt;SetUserInitialization(new BlActionInitialization());</pre>		
	81		
	82 🖻 // Initialize G4 kernel 83   //		
	<pre>03 // // 84 runManager-&gt;Initialize():</pre>		
	85		
	10% - 4		
	Output	• # ×	
	Show output from: Build - 3 2 2 3 3 2	* * ^	
	show output from: Build Project: ZERO CHECK, Configuration: Release Win32		
	1> Checking Build System	<u> </u>	
	<ol> <li>CMake does not need to re-run because C:/Users/Ben/Documents/examples/basic/B1-build/CMakeFiles/generate.stamp is up-to-date.</li> <li>2&gt; Build started: Project: example81, Configuration: Release Win32</li> </ol>		
	2> BULIG started: Project: examplesi, Configuration: Release Win32 2> Buliding Custom Rule C:/Desr/Ben/Documents/examples/Dasic/B1/CHakeLists.txt		
	2> CMake does not need to re-run because C:\Users\Ben\Documents\examples\basic\B1-build\CMakeFiles\generate.stamp is up-to-date.		
	2> exampleB1.cc 2> B1ActionInitialization.cc	1	
	2> BibetetorConstruction.cc		
	2> B1EventAction.cc		
	2> B1PrimaryGeneratorAction.cc 2> B1Run.cc		
	2> BiRunAction.cc		
	2> BISteppingAction.cc		
	2> Generating Code 2> exampleB1.vcxprof -> C:\Users\Ben\Documents\examples\basic\B1-build\Release\exampleB1.exe		
	3> Build started: Project: ALL_BUILD, Configuration: Release Win32		
	3> Building Custom Rule C:/Users/Ben/Documents/examples/basic/B1/CMakeLists.txt		
	3> CMake does not need to re-run because C:\Users\Ben\Documents\examples\basic\B1-build\CMakeFiles\generate.stamp is up-to-date. 3> Build all projects		
	======================================		lame) pecifies the project name.
		↓ sp	recines the project name.
Build succeeded			
🕘 🧁 📜 🖸			▲ (*) ■ 17:41 27/11/2013

18. Whilst you can run the built application in Visual Studio, it can be tricky to set up all the paths correctly (see below). It's therefore easiest to open a *cmd.exe* window and *cd* to the *build directory* (the directory where the Visual Studio solution was generated). You can then run the built application directly from the command line. Note that because Visual Studio supports multiple configurations and we've built in *Release* mode, the application is located at *Release*\*exampleB1.exe*.

🔩 B1 - Microsoft Visual C++ 2010 Express	- # X
File Edit View Project Debug Tools Window Help	
[ 🔂 = 🖄 = 🕼 🚚 🕼 🖕 그 🕲 9 - 안 = 🕨 Relesse - Win32 - 🔯 9 G4ParticleTable - 직 3 3 3 3 2 3 3	
西西  ふ, 口公知本  三台。	
	→ Properties → 4 × X
(Global Scope) - <sup>14</sup> main(int argc, char** argv)	ALL_BUILD Project Properties
Image: Set of	All BUILD     Project Deer     Project Deer     Root Namesp
100 de la carre une 100 de la carre une 100 de la carre de	•
	<b>_</b>
Nicrosoft Windows (Uersion 6.1.7601) Copyright (c) 2009 Microsoft Corporation. All rights reserved.	<u>^</u>
Copyright Co. 2009 Microsoft Corporation. All rights reserved. C:NJsers/Ben/Cd Documents C:NJsers/Ben/Documents/cd axanples/hasic/BI-build C:NJsers/Ben/Documents/caxanples/hasic/BI-build/Release/exampleBI.exc_ k	E
d.Nelass/texampleB1.exe 32	(Name) * Specifies the project name.
Build succeeded	▲ (*) → 17:45 27/11/2013

19. On execution, the *exampleB1* application should pop up a UI window, and a visualization window if your install of Geant4 was installed with OpenGL visualization.

	dicrosoft Visual C++ 2010 Expres		viewer-0 (OpenGLStoredWin32)	- • •
	t View Project Debug To 🗊 = 📂 🎜 🐊 🐊 👗 骗 🎇			
	A. 15 & L + 1			
🔣 Solut	ion Explorer 🛛 👻 🖡 🗙	exampleB1.cc ×		
S 🔛		(Global Scope) - 🌳 main(int argc, cha		
△ △ ■ △	Solution 181: Gproject) 3 ALL OUD 3 DL 5 DL 5 DL 5 DL 5 DL 6 Cample 3 6 DL 1 Charles 1 1 Charles 1	<pre>70 /// 71 /// Detector construction 72 rundmanger-&gt;SetUserInitialization(new BlDetectorConstruction()); 73 // Physics list 74 GWHOOLaPHysicsList physicsList = new (BBC; 75 physicsList-&gt;SetUserDoteLevel(1); 75 rundmanger&gt;-SetUserInitialization(physicsList); 76 rundmanger&gt;-SetUserInitialization(new BlActionInitialization()); 78 // User setUserInitialization(new BlActionInitialization()); 79 // User setUserInitialization(new BlActionInitialization()); 70 // User setUserInitialization(new BlActionInitialization()); 71 // UserInitialization(new BlActionInitialization()); 72 // UserInitialization(new BlActionInitialization()); 73 // UserInitialization(new BlActionInitialization()); 74 // UserInitialization(new BlActionInitialization()); 75 // UserInitialization(new BlActionInitialization()); 76 // UserInitialization(new BlActionInitialization()); 77 // UserInitialization(new BlActionInitialization()); 78 // UserInitialization(new BlActionInitialization()); 79 // UserInitialization(new BlActionInitialization()); 70 // UserInitialization(new BlActionInitialization(new BlActionInitialization()); 71 // UserInitialization(new BlActionInitialization(new BlAc</pre>		<b>c4</b>
	/vis/viewer/set/styl /vis/viewer/set/hidd /vis/viewer/set/view # # Re-establish auto			
	G401sHttr linestyle: solid, drawing style: not line segments per time range: (-1.79 G4AttValue pointer /vis/verbose warning: Visualization verbos	Dattes: visible, Augures visible, colour: (0.1.0.1) (green ino vidh: 1 Torend, auxiliar, edge visibility: not forced 600-3081.7/97650-308) is prov. GdHtDef pointer is zero d(Releast)cian exp, use this to create an empty detector view: 2) Build Historical States 2) Build His	leðl.exe ild/CMakeffles\generate.stamp is up-to-date. NAme	
		<pre>====================================</pre>		es the project name.
Build succ	reeded	*L		
<b>6</b>	6 📋 0			▲ ♦> ► 17:46 27/11/2013

20. The *exampleB1* can be controlled from the Geant4 CLI as normal, here we have run 10 events.

😫 B1 - Microsoft Visual C++ :		Wewer-U (UpenGLStoredWin32)	)	- 8 %
File Edit View Project [	~			
Solution Explorer	- 4 ×			
		(Global Scope) • 10 main(int argc, char		
Comment Outline Statution B3: (5 project ALE STATE ALE STATE A	endencies	70 // 71 // Detector construction 72 runWaager->SetUserInitialization(new BiDetectorConstruction()); 73 // University		
		Outout		
		Show output from: Build 🔹 🖓 💭 🦓 🖃		
Voxelisation Percent To 0.00 Voxelisation	: top CF tal CPU 0.00 : top me	10 0.00 Ok Envelope zamples/basic/81-bu		
Percent 1 108.00 ris/sconorotify ### Run Ø starts: Run Stomary Runber of even User=0.062s Res The run consist Dose in scoring	Menory Øk yHandlen • ts proce al=0s Sy End s of 10 volume	Heads Nodes Pointers Total CPU Unlune : 1 3 4 0.00 Envelope rs scene=0	mp is up-to-date.	(Name) Specifies the project name.
Ready			Ln 22 Col 1	Ch 1 INS
🚱 🏉 🚞				▲ (>) IT:50 27/11/2013

As noted above, you can also run the application through the Visual Studio debugger, although we have seen issues with setting the directory to run incorrectly. The executable locates the macro files by assuming they are in the current working directory, so the executable must be run from a directory containing the macros.

One key CMake related item to note goes back to our listing of the *headers* for the application in the call to add\_executable. Whilst CMake will naturally ignore these for configuring compilation of the application, it will add them to the Visual Studio Solution. If you do not list them, they will not be editable in the Solution.

#### 3.2.1.3. Usage of Geant4Config.cmake

Geant4Config.cmake is designed to be used with CMake's find\_package command. When found, it sets several CMake variables and provides a mechanism for checking and activating optional features of Geant4. This allows you to use it in many ways in your CMake project to configure Geant4 for use by your application.

The most basic usage of Geant4Config.cmake in a CMakeLists.txt file is just to locate Geant4 with no requirements on its existence, version number or components:

find\_package(Geant4)

If you must find Geant4, then you can use

find\_package(Geant4 REQUIRED)

This will cause CMake to fail should an install of Geant4 not be located.

When an install of Geant4 is found, the module sets a sequence of CMake variables that can be used elsewhere in the project:

• Geant4\_FOUND

Set to CMake boolean true if an install of Geant4 was found.

• Geant4\_INCLUDE\_DIRS

Set to a list of directories containing headers needed by Geant4. May contain paths to third party headers if these appear in the public interface of Geant4.

• Geant4\_LIBRARIES

Set to the list of libraries that need to be linked to an application using Geant4.

• Geant4\_DEFINITIONS

The list of compile definitions needed to compile an application using Geant4. This is most typically used to correctly activate UI and Visualization drivers.

• Geant4\_CXX\_FLAGS

The compiler flags used to build this install of Geant4. Usually most important on Windows platforms.

• Geant4\_USE\_FILE

A CMake script which can be included to handle certain CMake steps automatically. Most useful for very basic applications.

• Geant4\_builtin\_clhep\_FOUND

A CMake boolean which is set to true if this install of Geant4 was built using the internal CLHEP.

• Geant4\_builtin\_expat\_FOUND

A CMake boolean which is set to true if this install of Geant4 was built using the internal Expat.

• Geant4\_DATASETS

A CMake list of the names of the physics datasets used by physics models in Geant4. It is provided to help iterate over the Geant4\_DATASET\_XXX\_YYY variables documented below.

• Geant4\_DATASET\_<NAME>\_ENVVAR

The name of the environment variable used by Geant4 to locate the dataset with name <NAME>.

• Geant4\_DATASET\_<NAME>\_PATH

The absolute path to the dataset with name <NAME>. Note that the setting of this variable does not guarantee the existence of the dataset, and no checking of the path is performed. This checking is not provided because the action you take on non-existing data will be application dependent.

You can access the Geant4\_DATASET\_XXX\_YYY variables in a CMake script in the following way:

```
find_package(Geant4_REQUIRED)  # Find Geant4
foreach(dsname ${Geant4_DATASETS})  # Iterate over dataset names
    if(NOT EXISTS ${Geant4_DATASET_${dsname}_PATH})  # Check existence
    message(WARNING "${dsname} not located at ${Geant4_DATASET_${dsname}_PATH}")
    endif()
endforeach()
```

A typical use case for these variables is to automatically set the dataset environment variables for your application without the use of the shell scripts described in Section 3.1. This could typically be via a shell script wrapper around your application, or runtime configuration of the application environment via the relevant C/ C++ API for your system.

The typical usage of find\_package and these variables to configure a build requiring Geant4 is thus:

<pre>find_package(Geant4 REQUIRED) include_directories(\${Geant4_INCLUDE_DIRS}) add_definitions(\${Geant4_DEFINITIONS}) set(CMAKE_CXX_FLAGS \${Geant4_CXX_FLAGS})</pre>	# 2 # 2	Find Geant4 Add -I type paths Add -D type defs Optional
add_executable(myg4app myg4app.cc) target_link_libraries(myg4app \${Geant4_LIBRARIES})		Compile application Link it to Geant4

Alternatively, the CMake script pointed to by Geant4\_USE\_FILE may be included:

<pre>find_package(Geant4 REQUIRED) include(\${Geant4_USE_FILE})</pre>	<pre># Find Geant4 # Auto configure includes/flags</pre>
<pre>add_executable(myg4app myg4app.cc) target_link_libraries(myg4app \${Geant4_LIBRARIES})</pre>	# Compile application # Link it to Geant4

This is very useful for basic applications, but if you need fine control, you should use the variables directly.

By default, CMake will look in several platform dependent locations for the Geant4Config.cmake file (see find\_package for listings). You can also specify the location yourself when running CMake by setting the Geant4\_DIR variable to the path of the directory holding Geant4Config.cmake. It may be set on the command line via a -D option, or by adding an entry to the CMake GUI. For example, if we have an install of Geant4 located in

```
+- opt/
+- Geant4/
+- lib/
+- libG4global.so
+- ...
+- Geant4-10.0.0/
+- Geant4Config.cmake
```

then we would pass the argument -DGeant4\_DIR=/opt/Geant4/lib/Geant4-10.0.0 to CMake.

You can also, if you wish, build an application against a build of Geant4 without installing it. If you look in the directory where you built Geant4 itself (e.g. on UNIX, where you ran make), you see there is a

Geant4Config.cmake file. This is a perfectly valid file, so you can also point CMake to this file when building your application. Simply set Geant4\_DIR to the directory where you built Geant4. This feature is most useful for Geant4 developers, but it can be useful if you cannot, or do not want to, install Geant4.

A version number may also be supplied to search for a Geant4 install *greater than or equal to* the supplied version, e.g.

find\_package(Geant4 9.6.0 REQUIRED)

would make CMake search for a Geant4 install whose version number is greater than or equal to 9.6.0. An exact version number may also be specified:

find\_package(Geant4 10.0.0 EXACT REQUIRED)

In both cases, CMake will fail with an error if a Geant4 install meeting these version requirements is not located.

Geant4 can be built with many optional components, and the presence of these can also be required by passing extra "component" arguments. For example, to require that Geant4 is found *and* that it support Qt UI and visualization, we can do

find\_package(Geant4 REQUIRED qt)

In this case, if CMake finds a Geant4 install that does *not* support Qt, it will fail with an error. Multiple component arguments can be supplied, for example

find\_package(Geant4 REQUIRED qt gdml)

requires that we find a Geant4 install that supports both Qt and GDML. If the component(s) is(are) found, any needed header paths, libraries and compile definitions are appended to the variables Geant\_INCLUDE\_DIRS, Geant4\_LIBRARIES and Geant4\_DEFINITIONS respectively. Variables Geant4\_<COMPONENTNAME>\_FOUND are set to TRUE if component COMPONENTNAME is supported by the installation.

If you want to activate options only if they exist, you can use the pattern

find\_package(Geant4 REQUIRED)
find\_package(Geant4 QUIET COMPONENTS qt)

which will require CMake to locate a core install of Geant4, and then check for and activate Qt support if the install provides it, continuing without error otherwise. A key thing to note here is that you can call find\_package multiple times to append configuration of components. If you use this pattern and need to check if a component was found, you can use the Geant4\_<COMPONENTNAME>\_FOUND variables described earlier to check the support.

The components which can be supplied to find\_package for Geant4 are as follows:

multithreaded

Geant4\_multithreaded\_FOUND is TRUE if the install of Geant4 was built with multithreading support.

Note that this only indicates availability of multithreading support. Multithreading in an application requires creation and usage of the appropriate C++ objects and interfaces as described in the Application Developers Guide.

• gdml

Geant4\_gdml\_FOUND is TRUE if the install of Geant4 was built with GDML support.

• g3tog4

Geant4\_g3tog4\_FOUND is TRUE if the install of Geant4 provides the G3ToG4 library.

• ui\_tcsh

Geant4\_ui\_tcsh\_FOUND is TRUE if the install of Geant4 provides the TCsh command line User Interface.

• ui\_win32

Geant4\_ui\_win32\_FOUND is TRUE if the install of Geant4 provides the Win32 command line User Interface.

• motif

 ${\tt Geant4\_motif\_FOUND}\ is\ {\tt TRUE}\ if\ the\ install\ of\ Geant4\ provides\ the\ Motif(Xm)\ User\ Interface\ and\ Visualization\ driver.$ 

• qt

Geant4\_qt\_FOUND is TRUE if the install of Geant4 provides the Qt4 User Interface and Visualization driver.

• wt

Geant4\_wt\_FOUND is TRUE if the install of Geant4 provides the Wt Web User Interface and Visualization driver.

• vis\_network\_dawn

Geant4\_vis\_network\_dawn\_FOUND is TRUE if the install of Geant4 provides the Client/Server network interface to DAWN visualization.

• vis\_network\_vrml

Geant4\_vis\_network\_vrml\_FOUND is TRUE if the install of Geant4 provides the Client/Server network interface to VRML visualization.

• vis\_opengl\_x11

Geant4\_vis\_opengl\_x11\_FOUND is TRUE if the install of Geant4 provides the X11 interface to the OpenGL Visualization driver.

• vis\_opengl\_win32

Geant4\_vis\_opengl\_win32\_FOUND is TRUE if the install of Geant4 provides the Win32 interface to the OpenGL Visualization driver.

• vis\_openinventor

Geant4\_vis\_openinventor\_FOUND is TRUE if the install of Geant4 provides the OpenInventor Visualization driver.

• ui\_all

Activates all available UI drivers. Does not set any variables, and never causes CMake to fail.

• vis\_all

Activates all available Visualization drivers. Does not set any variables, and never causes CMake to fail.

Please note that whilst the above aims to give a complete summary of the functionality of Geant4Config.cmake, it only gives a sampling of the ways in which you may use it, and other CMake func-

tionality, to configure your application. We also welcome feedback, suggestions for improvement and bug reports on Geant4Config.cmake.

# **3.2.2. Using Geant4Make to build Applications:** binmake.gmk

Geant4Make is the Geant4 GNU Make toolchain formerly used to build the toolkit and applications. It is installed on UNIX systems (except for Cygwin) for backwards compatibility with the Geant4 Examples and your existing applications which use a GNUmakefile and the Geant4Make binmake.gmk file. *However, please note that the system is now deprecated, meaning that it is no longer supported and may be removed in future releases without warning. You should migrate your application to be built using CMake via the* Geant4Config.cmake script, *or any other build tool of your choice, using the* geant4-config program to query the relevant compiler/linker flags.

The files for Geant4Make are installed under:

```
+- CMAKE_INSTALL_PREFIX/
+- share/
+- geant4make/
+- geant4make.sh
+- geant4make.csh
+- config/
+- binmake.gmk
+- ...
```

The system is designed to form a self-contained GNUMake system which is configured primarily by environment variables (though you may manually replace these with Make variables if you prefer). Building a Geant4 application using Geant4Make therefore involves configuring your environment followed by writing a GNUmakefile using the Geant4Make variables and GNUMake modules.

To configure your environment, simply source the relevant configuration script CMAKE\_INSTALL\_PREFIX/ share/Geant4-10.0.0/geant4make/geant4make.(c)sh for your shell. Whilst both scripts can be sourced interactively, if you are using the C shell and need to source the script inside another script, you must use the commands:

cd CMAKE\_INSTALL\_PREFIX/share/Geant4-10.0.0/geant4make source geant4make.csh

or alternatively

source CMAKE\_INSTALL\_PREFIX/share/Geant4-10.0.0/geant4make/geant4make.csh \\
CMAKE\_INSTALL\_PREFIX/share/Geant4-10.0.0/geant4make

In both cases, you should replace CMAKE\_INSTALL\_PREFIX with the actual prefix you installed Geant4 under. Both of these commands work around a limitation in the C shell which prevents the script locating itself.

Please also note that due to limitations of Geant4Make, you *should not* rely on the environment variables it sets for paths into Geant4 itself. In particular, note that the G4INSTALL variable *is not equivalent to CMAKE\_INSTALL\_PREFIX*.

Once you have configured your environment, you can start building your application. Geant4Make enforces a specific organization and naming of your sources in order to simplify the build. We'll use Basic Example B1, which you may find in the Geant4 source directory under examples/basic/B1, as the canonical example again. Here, the sources are arranged as follows

+- B1/

+- GNUmakefile

+- exampleB1.cc

```
+- include/
| ... headers.hh ...
+- src/
... sources.cc ...
```

As before, exampleB1.cc contains main() for the application, with include/ and src/ containing the implementation class headers and sources respectively. You must organise your sources in this structure with these filename extensions to use Geant4Make as it will expect this structure when it tries to build the application.

With this structure in place, the GNUmakefile for exampleB1 is very simple:

```
name := exampleB1
G4TARGET := $(name)
G4EXLIB := true
.PHONY: all
all: lib bin
include $(G4INSTALL)/config/binmake.gmk
```

Here, name is set to the application to be built, and it must match the name of the file containing the main() program without the .cc extension. The rest of the variables are structural to prepare the build, and finally the core Geant4Make module is included. The G4INSTALL variable is set in the environment by the geant4make script to point to the root of the Geant4Make directory structure.

With this structure in place, simply run make to build your application:

```
$ make
```

If you need extra detail on the build, you append CPPVERBOSE=1 to the make command to see a detailed log of the command executed.

The application executable will be output to \$(G4WORKDIR)/bin/\$(G4SYSTEM)/exampleB1, where \$(G4SYSTEM) is the system and compiler combination you are running on, e.g. Linux-g++. By default, \$(G4WORKDIR) is set by the geant4make scripts to \$(HOME)/geant4\_workdir, and also prepends this directory to your PATH. You can therefore run the application directly once it's built:

```
$ exampleB1
```

If you prefer to keep your application builds separate, then you can set G4WORKDIR in the GNUmakefile before including binmake.gmk. In this case you would have to run the executable by supplying the full path.

Further documentation of the usage of Geant4Make and syntax and extensions for the GNUMakefile is described in the FAQ and Appendices of the Geant4 User's Guide for Application Developers.

Please note that the Geant4Make toolchain is provided purely for conveniance and backwards compatibility. We encourage you to use and migrate your applications to the new CMake and geant4-config tools. Geant4Make is deprecated in Geant4 10.0 and later.

### 3.2.3. Other Unix Build Systems: geant4-config

If you wish to write your own Makefiles or use a completely different buildsystem for your application, a simple command line program named geant4-config is installed on Unix systems to help you query the Geant4 installation. It is located in

+- CMAKE\_INSTALL\_PREFIX +- bin/ +- geant4-config It may be run using either a full or relative path, or directly if CMAKE\_INSTALL\_PREFIX/bin is in your PATH.

This program provides the following command line interface for querying various parameters of the Geant4 installation:

\$ geant4-confighelp	
Usage: geant4-config [OPI	
prefix	output installation prefix of Geant4
version	output version for Geant4
libs	output all linker flags
cflags	output all preprocessor
	and compiler flags
libs-without-gui	output linker flags without
5	GUI components
cflags-without-gui	output preprocessor and compiler
	flags without GUI components
has-feature FEATURE	output yes if FEATURE is supported,
	or no if not supported
datasets	output dataset name, environment variable
	and path, with one line per dataset
check-datasets	output dataset name, installation status and
	expected installation location, with one line
	per dataset
install-datasets	download and install any missing datasets,
install datasets	requires a network connection and for the dataset
	path to be user writable
Known Features:	
multithreading[no]	
clhep[yes]	
expat[no]	
gdml[no]	
usolids[no]	
g3tog4[no]	
qt[no] motif[no]	
raytracer-x11[no]	
opengl-x11[no]	
openinventor[no]	
openinveneor(no)	
Help options:	
-?,help	show this help message
usage	display brief usage message

You are completely free to organise your application sources as you wish and to use any buildsystem that can interface with the output of geant4-config.

The --cflags argument will print the required compile definitions and include paths (in -I<path> format) to use Geant4 to stdout. Note that default header search paths for the compiler Geant4 was built with are filtered out of the output of --cflags. The --libs argument will print the libraries (in -L<path> -lname1 ... -lnameN format) required to link with Geant4 to stdout. Note that this may include libraries for third party packages and may not be reliable for static builds. By default, all the flags and Geant4 libraries needed to activate all installed UI and Visualization drivers are provided in these outputs, but you may use the -without-gui variants of these arguments to suppress this.

You may also check the availability of features supported by the install of Geant4 with the --has-feature argument. If the argument to --has-feature is known to Geant4 *and* enabled in the installation, yes will be printed to stdout, otherwise no will be printed.

The --datasets argument may be used to print out a table of dataset names, environment variables and paths. No checking of the existence of the paths is performed, as the action to take on a non-existing dataset will depend on your use case. The table is printed with one row per dataset, with space separated columns for the dataset

name, environment variable name and path. As with Geant4Config.cmake, this information is provided to help you configure your application environment to locate the Geant4 datasets without a preexisting setup, if your use case demands this.

The --check-datasets argument may be used to check whether the datasets are installed in the location expected (as set by the configuration of Geant4). A table is printed with one row per dataset, with space separated columns for the dataset name, installation status and expected path. If the expected path is found, the status column will contain INSTALLED, otherwise it will contain NOTFOUND. Note that this check only verifies the existence of the dataset path. It does not validate that the dataset files are all present nor that the relevant environment variables are set.

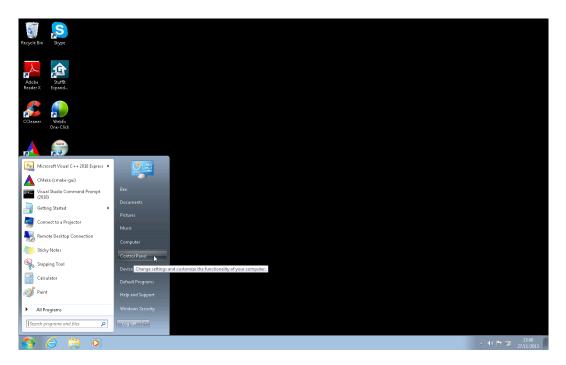
If you did not use the GEANT4\_INSTALL\_DATA option to install data when Geant4 itself was installed, you can use the --install-datasets argument to perform this task at a later time. Running geant4-config with this argument will download, unpack and install each dataset to the location expected by the Geant4 installation. These steps require a working network connection, the local dataset installation path to be writable by the user running geant4-config and the presence of the curl, openssl and tar programs. Note that no changes to the environment are made by the data installation, so you may need to update this using the relevant scripts documented in Section 3.1.

Due to the wide range of possible use cases, we do not provide an example of using geant4-config to build an application. However, it should not require more than appending the output of --cflags to your compiler flags and that of --libs to the list of libraries to link to. We welcome feedback, suggestions for improvement and bug reports on geant4-config.

## 3.3. Note on Geant4 Datasets

If you built and installed Geant4 configured with the option GEANT4\_INSTALL\_DATA set, then the Geant4 datasets will have been downloaded and installed automatically.

In this case, the geant4. (c) sh and geant4make. (c) sh scripts will set up the needed environment variables required by Geant4 to locate these datasets. On Windows, you will need to set the needed environment variables listed below by hand. This can be done as follows



1. Open the Windows Control Panel.

2. Open the *System* item in the Control Panel.

dobe Stuffit ader X Expand	Control Panel + All Con	trol Panel Items 🕨		
<b>\$</b>	Adjust your computer's settings		View by: Small icons 🔻	
leaner WebEx One-Click	Action Center	administrative Tools	T AutoPlay	
-	Backup and Restore	RitLocker Drive Encryption	Color Management	
Naid	Credential Manager	Pate and Time	Default Programs	
lake NSIS	Desktop Gadgets	Bevice Manager	R Devices and Printers	
iake ivuus (e-gui)	E Display	S Ease of Access Center	Flash Player (32-bit)	
	Folder Options	K Fonts	Getting Started	
	HomeGroup	A Indexing Options	C Internet Options	
	🛃 Java (32-bit)	a Keyboard	Location and Other Sensors	
ywin	Mail (32-bit)	I Mouse	Vetwork and Sharing Center	
	Notification Area Icons	NVIDIA Control Panel	A Parental Controls	
	Performance Information and Tools	Personalization	Phone and Modern	
-	Power Options	Programs and Features	P Recovery	
zilla	Region and Language	RemoteApp and Desktop Connections	Sound	
fox	Speech Recognition	Sync Center	j≅ Syster]⊾	
	Taskbar and Start Menu	Troubleshooting	& User A System	
	Windows CardSpace	Windows Defender	Windd View information about your	
	Windows Update		computer, and change settings for hardware, performance, and remote	
Office 0.0	windows opdate		connections.	
			connections.	

3. Click on the *Advanced system settings* link on the System window (on *Windows XP*, click on the *Advanced* tab).

Recycle Bin	Skype						
Adobe Reader X	Stuffit Expand		🕒 🔵 🗢 📴 🕨 Control Pa	nel → All Control Panel Rems → System	Search Control Panel	×	
CCleaner	Web£x One-Click		Control Panel Home Device Manager Remote settings System protection Advanced system settings	View basic information about your computer Windows edition Windows 7 Enterprise Copyright © 2009 Microsoft Corporation. All rights reserved. Service Pack 1	<b>~</b>		
CMake (cmake-gui) Cygwin	NIIS			System		E	
Mozilla Firefox			See also Action Center Windows Update	Pen and Touch: Computer name, domain, and workgroup settings Full computer name Full computer name Computer description: Workgroup:	😵 Change settings		
OpenOffice 4.0.0			Performance Information Tools	Windows activation	an tor . die	*	
<b>(</b>	6	0					▲ (1) (1) 15:13 27/11/2013

4. Click on the *Environment Variables* button in the System Properties window.

#### Setting Up and Using an Install of Geant4

Recycle Bin Stype		
Adobe Spand. Control Fund	All Control Panel Items + System + System Panel Control Panel Panel	
Cottanter WithEx Orse-Click Provide Manager Provide Manager Provide Manager Provide Manager Provide Manager	View basic information about your computer Windows ef Windows ef Window Copyrig Copyrig Language Name Hadware Advanced System Protection Remote Copyrig	
CMuke (cmixke-gui)	Service You must be kgged on as an Administrator to make most of these charges.	
Cygwin	Rating: Use Profiles = = = = = = = = = = = = = = = = = = =	
Mozila Firefox See also Action Center	Pen and Stantag and Recovery Computer System stantag, system fakee, and debugging information Comput Full com Comput Comp	
Action Control Windows Update Performance Information and Tools	Windows # DK Cancel Activ	
Policestor		
🚱 🤌 🚞 💽 🐺 🥾		▲ ● ● ● 15:19 27/11/2013

5. For each dataset, click on the *New* button to create a new entry

Recycle Bin	Skype									
Adobe Reader X	Stuffit Expand	<b>G</b> • <b>E</b>	<ul> <li>Control Panel          <ul> <li>All Cor</li> </ul> </li> </ul>	ntrol Panel Ite	ems 🕨 System	• 47	Search Control Panel			
CCleaner	WebEx One-Click	Control Pa 😵 Device Ma 😵 Remote se 😵 System pro	Vie nager Wir ttings	Windo		dvanced System Protection Remote				
CMake (cmake-gui)		😵 Advanced	Syst	Service	TEMP %JUS	a sers\Ben\Documents\geant4\geant ERPROFILE%\AppData\Local\Temp				
Cygwin				Proces Installe System Pen an	System variables	ERPROFILE%(AppData)Local(Temp		E		
Mozilla Firefox		See also Action Cer Windows I	iter	mputer Compi Full co Compi Workg	FP_NO_HOST_C NO NUMBER_OF_P 2 OS Wind			hange settings		
OpenOffice 4.0.0			ce Information and Wir	ndows a Windows		OK Cance		asktar. ২৮		
PDFCreator		0 📴 🧏							<ul> <li>↓&gt;  ■&lt; (□</li> </ul>	15:58 27/11/2013

6. Enter the Variable name and Variable value using the names and paths described below.

#### Setting Up and Using an Install of Geant4

Recycle Bin	Skype				
Adobe Reader X	Stuffit Expand	🚱 🔍 🖷 🕨 Control Panel 🕨	All Control Par	el Bernis > System • 4	
CCleaner	WebEx One-Click	Control Panel Home           Ouvice Manager           Remote settings           System protection           Advanced system settings	6	ic information about your computer System Properties Compute Name Hardware Advanced System Protection Remote Environment Variables	
CMake (cmake-gui)	NSIS		System — Rating: Proces Installe	User variables for Ben Veriable Value PATH Value Colluers/SeniDocument/gantHigant TOPP VulsePArnable Veriable name: 64LEDATA	
Mozilla Firefox		See also Action Center	Systen Pen an Computer Comp Full co Comp	Variable value: share/Geam+10.0.0/data/G49N.0W6.3] Cancel FP_INO_HOST_C NO NUMBER_OF_F 2 OS Windows_IVT	
OpenOffice 4.0.0		Windows Update Performance Information and Tools	Workg Windows & Window	New Edt Delete	
PDFCrestor	6	<b>9</b>			- () <b>►</b> ( <b>)</b> 1617 27/11/2013

If you chose not to install the datasets, but require them later, the geant4-config program may be used to install them on Unix platforms via its --install-datasets argument. This is documented in Section 3.2.3.

On Windows platforms, you will need to download and unpack them by hand to the location you specified for GEANT4\_INSTALL\_DATADIR. If you did not set this, it defaults to CMAKE\_INSTALL\_PREFIX/share/Geant4-10.0.0/data. Unpacking the datasets in this location will result in Geant4 automatically locating them once geant4.(c)sh or geant4make.(c)sh have been sourced.

If you have datasets in a different location, then you will need to manually set the following environment variables:

• G4LEDATA

Set to the path to the G4EMLOW6.35 dataset.

• G4LEVELGAMMADATA

Set to the path to the PhotonEvaporation3.0 dataset.

• G4NEUTRONHPDATA

Set to the path to the G4NDL4.4 dataset.

• G4NEUTRONXSDATA

Set to the path to the G4NEUTRONXS1.4 dataset.

• G4PIIDATA

Set to the path to the G4PII1.3 dataset.

• G4RADIOACTIVEDATA

Set to the path to the RadioactiveDecay4.0 dataset.

• G4REALSURFACEDATA

Set to the path to the RealSurface1.0 dataset.

• G4SAIDXSDATA

Set to the path to the G4SAIDDATA1.1 dataset.

• G4ABLADATA

Set to the path to the G4ABLA3.0 dataset.

In Geant4 10.0, a new dataset G4ENSDFSTATE1.0 has been introduced. Whilst this dataset is downloaded and installed for convenience, it is only required in a few special use cases described in the Geant4 Guide for Toolkit Developers. It is therefore not configured by default (meaning that the environment variable pointing to the data is not set) and you should not do so unless your application has the specific use cases discussed in that guide. Please see the Application Developer's Guide for information about these use cases and how to configure the use of G4ENSDFSTATE1.0.

# Chapter 4. CMake and Build Tools For Geant4 Developers

Geant4 developers can make use of several powerful features of CMake to help with their work. The key concept and working practice is the separation of the *source directory*, which is where the sources you edit reside, and the *build directory*, where the buildscripts and compiled products reside. The reason for enforcing this separation is twofold:

- It provides separation of CMake generated files (e.g. Makefiles) from the Geant4 sources under revision control.
- It allows multiple builds against a single source directory, giving fast incremental builds without having to reconfigure.

# 4.1. Developing Geant4 using Make, Xcode, Visual Studio and Eclipse

CMake is a *buildsystem generator* that can create scripts for many buildsystems including Make, Xcode, Visual Studio and Eclipse, among others. To find out which systems your install of CMake can generate scripts for, consult the "GENERATORS" section of the CMake man page, or click on the "Generate" button in the CMake GUI. The resulting scripts can be used within the buildsystem of choice to perform the actual build, install and packaging.

Whilst we only support Make and Visual Studio for Unix and Windows user builds respectively, Geant4 developers are welcome, and encouraged, to use their buildsystem of choice. Scripts for developing Geant4 using these systems can be generated by choosing the CMake generator when running CMake for the first time.

On the command line, one can select the tool using the –G argument of CMake. For example, to generate an Xcode project for Geant4 using the example from Section 2.1:

```
$ mkdir -p /path/to/geant4.10.0-build-xcode
$ cd /path/to/geant4.10.0-build-xcode
$ cmake -G Xcode -DCMAKE_INSTALL_PREFIX=/path/to/geant4.10.0-install /path/to/geant4.10.0
```

The resulting /path/to/geant4.10.0-build-xcode/Geant4.xcodeproj project may be opened with Xcode.

In the CMake GUI, the generator will be asked for the first time you click on *Configure* button (see Section 2.2), where it can be selected from a drop down list.

Note that in all cases, you can only have one buildtool configuration in a given build directory (e.g. you cannot have Unix Makefiles alongside an Xcode project).

Support for these buildtools is still preliminary, so feedback is welcome, whether bug reports, guides or general comments.

### 4.1.1. Using the Eclipse IDE

Eclipse projects using Makefiles can be generated via the command:

\$ cmake -G"Eclipse CDT4 - <TYPE> Makefiles" <otherargs>

where <TYPE> is platform dependent and one of Unix, MinGW or NMake. Note that only a single build mode is supported here because the projects are Makefile based. This means that you will need to supply CMake with the command line argument -DCMAKE\_BUILD\_TYPE=<MODE>, where <MODE>, is the required mode if you want to change the default mode. By default, Geant4 is built in "Release" mode.

With out-of-source builds enforced in Geant4, there are two issues that need to be worked around due to the way Eclipse expects project directories to be organised. These are to do with the integration of version control support and code editing/navigation/autocompletion. Both issues, together with their resolution are described in the CMake Wiki entry on Eclipse CDT4 under the "Out-ofSource Builds" section.

### 4.2. Command Line Help with Make

If you develop using the command line and Make, you can get information on the targets available by "building" the help target in your build directory:

```
$ make help
The following are some of the valid targets for this Makefile:
... all (the default if no target is provided)
... clean
... depend
.furthertargets.
```

You may build any target individually, and it will be built together with all of its dependencies. CMake's generated makefiles only output minimal information by default, so if you need to see the full commands used, you can run make with the extra VERBOSE argument:

```
$ make VERBOSE=1
```

to output every command in full.

If you want to quickly check that your target compiles, *without* checking and rebuilding any of its dependencies, you can append /fast to the target name, e.g.

\$ make G4run/fast

This will finish with an error if any dependents of the target do not exist, but can be useful for rapidly checking that your sources simply compile.

# 4.3. Building Quickly and Efficiently with Multiple Build Directories

The many ways in which Geant4 can be configured with optional components can make compilation and testing under different configurations time consuming.

As the CMake generated scripts live in a *build directory* isolated from the *source directory*, one can create several build directories configured against the same source directory. Each directory can have a different configuration, for example

```
$ cd /path/to
$ ls
geant4.10.0
$ mkdir geant4.10.0-build-release geant4.10.0-build-debug
$ cd geant4.10.0-build-release
$ cmake -DCMAKE_BUILD_TYPE=Release ../geant4.10.0
...output...
$ make -j8
...output...
$ cd ../geant4-build-debug
$ cmake -DCMAKE_BUILD_TYPE=Debug ../geant4.10.0
...output...
```

\$ make -j8

The above example uses Unix Makefiles, but the same technique works on all platforms. It may not seem to have gained you much, but when you edit and develop code that is living under /path/to/geant4.10.0, you only need to rebuild in each directory:

```
... work on code ...
$ cd /path/to/geant4.10.0-build-release
$ make -j8
... incremental build ...
$ cd /path/geant4.10.0-build-debug
$ make -j8
... incremental build ...
```

The builds pick up the changes you make to the source and build separately *without needing reconfiguration*. This is particularly powerful if the different configurations you need to test (for example, different versions of an external package) would require significant recompilation if the configuration were changed. Naturally, this power comes at the cost of some disk space, so may not be ideal in all cases.

Note that whilst this technique works on all platforms and buildtools, some IDEs (Xcode or Visual Studio for example) automatically support multiple build modes such as Release and Debug. In this case, you do not need separate build directories as the IDE handles this for you. However, you would still need two separate build directories if you, for example, wanted to develop and test against two versions of an external package such as Xerces-C.

### 4.4. Building Test Applications Against Your Development Build

A key feature of Geant4's CMake scripts is that you *do not* need to install your current build to be able to use it. A typical use case here is that you have a simple testing application which you want to build against your latest development build of Geant4.

Versions of the geant4cmake. (c) sh (described in Section 3.2.2), Geant4Config.cmake (described in Section 3.2.1) and geant4-config (described in Section 3.2.3) scripts are created in the build directory. These versions are all configured to use the libraries as they exist in the build directory, and headers from the source directory, without installation.

You can therefore use these scripts as described earlier in Chapter 3 to build your test applications *against a specific build tree*. You therefore don't need to install Geant4 everytime you make a small update.

# Chapter 5. Help And Support

## 5.1. Getting Help

Whilst every effort has been made to make the build of Geant4 robust and reliable, the multitude of platforms and system configurations mean we cannot guarantee that problems will not be encountered on platforms other than those listed in Section 1.1.

In case of issues with building and installing Geant4, we welcome questions as well as feedback via our HyperNews Forum. To help us deal with your problem as quickly as possible, please include as much detail as possible on the problem you have encountered. At minimum, you should let us know the platform and operating system version, your C++ compiler type and version, CMake version and any error messages. It also helps to have the sequence of commands you used so we can try and reproduce the issue.

Please note that as discussed earlier we can only support user installs on Unix via CMake and Unix Makefiles, and on Windows via CMake and Visual Studio. Developers however are welcome to try CMake and other buildtools like Xcode and Eclipse, and we welcome your feedback here.

If you feel you have found a genuine bug in the Geant4 CMake build, please report in to the CMake category on our Bugzilla. As with reports to HyperNews, please include as much information as possible so that we can triage the bug and track it down quickly. We also welcome general feature requests and feedback on the system.

## 5.2. Further Information

- CMake Documentation
- CMake Wiki
- CMake Tutorial