Setup instructions for Geant 4.9.5.p02 for the usage of the PanEDM UCN-simulation on Ubuntu 18.04

Leonard Romano

17.12.2019

Zusammenfassung

Here I show the exact steps I needed to install a running version of GEANT4 that is compatible with the PanEDM-UCN simulation. All information provided is up to my best knowledge and it is possible and in fact quite likely that easier ways to achieve this exist.

Inhaltsverzeichnis

1 Prerequisites

As mentioned in the title it is assumed you are using Ubuntu 18.04. For different operating systems other rules might apply, but the general procedure should be quite similar. Unfortunately I did not find a way to write the code in a way such that line breaks in this tutorial are not also directly translated to line breaks in the code, so it is recommended to remove line breaks whenever it is necessary when copying the code.

1.1 Needed standard packages

A handful of standard packages are needed for the installation of the required software. Pasting the following command in the terminal and pressing enter should take care of these dependencies:

```
sudo apt-get install -y g++ cmake qt4-default libsoqt4-dev
libsoqt-dev-common libqt4-dev libmotif-dev libxmu-dev
libxpm-dev libinventor1 csh autoconf automake tk gedit
```

After these packages have successfully been installed you can continue with the next steps.

1.2 Additional prerequisites

After the prerequisites for the prerequisites have been installed, we can start installing the prerequisites.

1.2.1 CLHEP

CLHEP [1] is a physics library that is needed for running Geant4. A detailed documentation is given on the official CLHEP webpage. For our purposes we will need Version 2.0.4.2. which can be downloaded from

```
http://proj-clhep.web.cern.ch/proj-clhep/DISTRIBUTION/clhep.html
```

Or is also available on my Git-repository[5], where I collect all needed files.
After you unzip the compressed version, autoconfig should set up everything, so we can immediately start with the configuration.
For this the following steps need to be followed:

- Create a build directory outside the source directory (here we assume the source was unpacked in the directory /home/<username>/CLHEP). Here <username> corresponds to your username on Ubuntu. So open a terminal in /home/<username>/CLHEP and type:

```
mkdir CLHEP-build
```

- Enter the build directory and run the configure script by typing:

```
cd CLHEP_build
/home/<username>/CLHEP/clhep-2.0.4.2/2.0.4.2/CLHEP/configure
--prefix=<path/to/install/directory>
```

  The path to the install directory has to be explicit. I recommend to put it in the same directory as the build directory e.g. substitute /home/username/CLHEP/CLHEP-install into <path/to/install/directory>

- Then "make", check and install:

```
make
make check
make install
```

  It is possible that make install needs root permissions. In that case simply replace the last line with:

```
sudo make install
```

- Finally we only need to export CLHEP in bashrc. To do so go to your home directory and type in the terminal:

```
gedit .bashrc
```

  A text editor opens. Include the following line in there and save:

```
export CLHEP_BASE_DIR=/home/<username>/CLHEP/CLHEP-install/
```

Now CLHEP should have been installed successfully.

1.2.2 libsoxt

Libsoxt is another package that needs to be installed. It can be downloaded from

```
https://bitbucket.org/Coin3D/coin/downloads/SoXt-1.3.0.tar.gz
```

Or from my Git-repository[5]. After upacking it similarly to CLHEP, autoconfig will create the configure file needed for configuration. For this packacge the following sequence of commands performed in the source directory will be sufficient to install it:

```
./configure
make
make check
make install
make clean
```

It is again possible that you need root permissions. In this case a ßudo"will do the trick.

1.2.3 DAWN

Another package we need is DAWN[2], which is a software package for visualization with GEANT4. A version of DAWN that works with Ubuntu 18.04 is DAWN 3.91a, which can be downloaded from

```
https://geant4.kek.jp/~tanaka/DAWN/About_DAWN.html
```

or from my Git-repository[5]. To install it, unzip it in a directory of your choice where autoconfig will create the needed configuration files. In the terminal go to the source directory and run the following commands:

```
make clean
make guiclean
./configure
make
sudo make install
```

After executing "./configure"you will be prompted to answer some questions. For my setup the answers (g++, -O0, wish, /usr/local/bin) worked but ultimately it depends on what you are using. After installation I recommend you test DAWN by executing:

```
dawn primitives.prim
```

in your source folder. First a menu should open where you can specify some visualization options, after you close it a picture should show up. For me this didn't work as ImageMagick 6 doesn't give permission to show them. This can be fixed easily however by going to /etc/ImageMagick-6 and commenting out the last few lines in "policy.xml":

```
  <!-- disable ghostscript format types
<policy domain="coder" rights="none" pattern="PS" />
<policy domain="coder" rights="none" pattern="PS2" />
<policy domain="coder" rights="none" pattern="PS3" />
<policy domain="coder" rights="none" pattern="EPS" />
<policy domain="coder" rights="none" pattern="PDF" />
<policy domain="coder" rights="none" pattern="XPS" /> -->
```

Afterwards the visualization with DAWN should work. For me however it still did not work, as it did not find the command 'dawn'. I fixed this by running "DAWN_GUI" separately to set all the options and then including the following alias in ".bashrc":

```
alias dawn='~/DAWN/dawn_3_91a/dawn_3_91a/dawn -d'
```

Where the '-d' option forces it to use the settings from the last session.

1.2.4 xerces-c

The last prerequisite you will need is xerces-c-3.1.1[3]. It is a validating C++ XML parser, that we will need to build GEANT4. It can be downloaded from the following website:

```
https://launchpad.net/ubuntu/+source/xerces-c
```

Or from my Git-repository[5]. To install it you need to unpack it in a directory of your choice, and again autoconfig will setup all the needed configuration files. As it happens to be GEANT4 requires a very specific path for the installation of xerces-c so you will need to create the following directory in the terminal:

```
cd /opt
sudo mkdir xerces-c-3.1.1
```

after that change to the source directory, and execute the configure file, make, check and install it:

```
./configure prefix=/opt/xerces-c-3.1.1
make
make check
sudo make install
```

This concludes the installation of prerequisites and we can finally move on to install GEANT4.

2 Installation of GEANT4

You have managed to install all the prerequisites and now you are finally ready to install GEANT4. First of all you will need a copy of the source directory, which you can get from my Git-repository[5]. In the examples some absolute paths will already have been set. In the case you want to compile the example programs, you will need to change the corresponding paths in the make and gnumake files of each example. For installation first unzip the source directory in a directory of your choice and create a build directory outside of the source:

```
mkdir GEANT4-build
cd GEANT4-build
```

Next we need to use cmake to configure the build (Here the source is located in /home/<username>/GEANT4/geant4.9.5.p02):

```
cmake -DCMAKE_INSTALL_PREFIX=/home/<username>/GEANT4/GEANT4-install
-DGEANT4_USE_XM=ON -DGEANT4_BUILD_EXAMPLES=OFF -DBUILD_SHARED_LIB=ON
-DGEANT4_USE_OPENGL_X11=ON -DXERCESC_ROOT_DIR=/opt/xerces-c-3.1.1/
-DGEANT4_USE_GDML=ON /home/<username>/GEANT4/geant4.9.5.p02
```

It will run for some time and if it succeeds it should end with the following lines (there can be some junk in between):

```
-- Configuring done
-- Generating done
-- Build files have been written to: /home/<username>/GEANT4/GEANT4-build
```

If there are any error messages at this point carefully check them. After this step succeeded you can make in the build directory:

```
make -jN
```

Where N is the number of parallel jobs you require (e.g. up to the number of processor cores of your machine).
After this has been completed simply execute in the build directory:

```
make install
```

Now in principle GEANT4 has been successfully installed but you first need to make your libraries available. To do so change your directory in the terminal to your install directory and run the following command:

```
cd /home/<username>/GEANT4/GEANT4-install/bin
. geant4.sh
```

Also you need to include the following lines in your .bashrc:

```
source /home/<username>/GEANT4/GEANT4-build/geant4make.sh
G4INCLUDE=/home/<username>/GEANT4/GEANT4-install/include/Geant4
export G4INCLUDE
LD_LIBRARY_PATH=/home/<username>/GEANT4/GEANT4-install/lib/
LD_LIBRARY_PATH=/usr/local/lib/:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=/home/<username>/geant4_workdir/tmp/Linux-g++/UCNgdml/:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

Where the UCNgdml one is specific for the UCNgdml project. Now you can try to run an example project, from the source directory or try to run the UCNgdml project which is available on my Git-repository[5].
More information on the installation of GEANT4 can be found on the official website[4].

3 Troubleshooting

In this section I will list some errors that occured and how we fixed them.
1. Problem with files in shared libraries

```
./UCNgdml: error while loading shared libraries: libxerces-c-3.1.so:
cannot open shared object file: No such file or directory
```

This could simply be fixed by copying the file /opt/xerces-c-3.1.1/lib/libxerces-c-3.1.so"to the directory /usr/lib".

Literatur

[1] http://proj-clhep.web.cern.ch/proj-clhep/

[2] https://geant4.kek.jp/ tanaka/

[3] file:///home/universe/xerces/xerces-c_3.1.1.orig/xerces-c-3.1.1/doc/html/index.html

[4] http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/InstallationGuide/html/gettingstarted.html

[5] https://github.com/leonardromano/PanEDM_GEANT4