Manual Geant4 Installation Guide

Prathamesh Changde
GitHub: pc2468

Contents

1 Introduction **3**

2 Basic Terminal Commands **3**
2.1 Terminal Shortcuts . . . 3

3 Prerequisites **4**

4 Installation Steps **4**
4.1 Distribution-Specific Dependencies . . . 4
4.1.1 Ubuntu, Debian, Mint / WSL . . . 4
4.1.1.1 Update Your System . . . 4
4.1.1.2 Install Required Dependencies . . . 5
4.1.2 Arch Linux . . . 5
4.1.2.1 Update Your System . . . 5
4.1.2.2 Install Required Dependencies . . . 5
4.1.3 Fedora . . . 5
4.1.3.1 Update Your System . . . 5
4.1.3.2 Install Required Dependencies . . . 5
4.1.4 openSUSE . . . 6
4.1.4.1 Update Your System . . . 6
4.1.4.2 Install Required Dependencies . . . 6
4.1.5 Rocky Linux, RHEL . . . 6
4.1.5.1 Update Your System . . . 6
4.1.5.2 Install Required Dependencies . . . 6
4.2 Create Geant4 Installation Folders . . . 6
4.3 Download Geant4 Source File . . . 7
4.4 Extract the Source Archive . . . 7
4.5 Configure the Build . . . 7
4.5.1 Configuration: . . . 7
4.5.2 Exiting Configuration: . . . 8
4.5.3 Build Options: . . . 8

4.5.4 Setting Installation Path and Enabling Options: 9
4.5.5 Final Configuration and Generation: 10
4.6 Generate Build Files: 10
4.7 Build and Install Geant4 10

5 Setup Environment **10**

6 Verify Installation with Basic Example **11**

7 Conclusion **12**

1 Introduction

This guide provides a step-by-step process for manually installing the Geant4 simulation toolkit on a Linux system using command-line instructions. Yes, command-line — that magical interface where real work happens, not just clicking shiny buttons.

This method gives you more control over the installation — which is great if you enjoy knowing what your machine is doing instead of hoping some script doesn't eat your system alive. It's ideal for those brave enough to step beyond the comfort of package managers.

Familiarity with basic Linux terminal commands is beneficial — or in other words, if you don't know how to use 'cd', maybe start there.

For comprehensive information, refer to the official Geant4 documentation — you know, the place where all the serious people hang out: `https://geant4.web.cern.ch/`.

2 Basic Terminal Commands

Here are the essential terminal commands used in this guide — basics every Linux user should know:

- `mkdir`: Create a directory.
- `cd`: Change your current directory.
- `rmdir`: Remove an empty directory.
- `ls`: List files in the current directory.
- `touch`: Create an empty file.
- `rm`: Delete files or directories — no undo button here.
- `kill` / `Ctrl+C`: Stop a running process.
- `wget`: Download a file from a URL.
- `git clone`: Copy a repository to your machine.

Note: Always start in the right working directory — saves you from cleaning up later.

2.1 Terminal Shortcuts

- Copy: `Ctrl+Shift+C`
- Paste: `Ctrl+Shift+V`

3 Prerequisites

Before you start, make sure your system meets these requirements — the basics, nothing too fancy:

- **Operating System**: Linux or Windows Subsystem for Linux (WSL).
- **C++ Compiler**: A compatible compiler (e.g., GNU GCC) and build tools (including **CMake**).
- **CMake**: Version 3.16 or later recommended.
- **Git**: Needed to clone the Geant4 source code.
- **Optional**: Python and Qt libraries for better visualization.
- **System Privileges**: Admin or `sudo` rights to install packages.
- **Command-Line Proficiency**: A basic grasp of the Linux terminal.
- **Disk Space**: At least 10 GB of free space for Geant4.
- **Internet Connection**: For downloading files and dependencies.

4 Installation Steps

Let's get started with installing Geant4 — it's not going to install itself!

4.1 Distribution-Specific Dependencies

Depending on your Linux distro, you'll need to install specific development packages. These packages might change with new Geant4 versions or updates to your distro. So, it's on you to keep an eye on the changes and make sure you're installing the right versions.
Follow the instructions for your system — it's not rocket science, but it's close.

4.1.1 Ubuntu, Debian, Mint / WSL

4.1.1.1 Update Your System

```
sudo apt update && sudo apt upgrade -y
```

4.1.1.2 Install Required Dependencies

```
sudo apt install -y \
cmake-curses-gui cmake g++ gcc binutils \
libx11-dev libxpm-dev libxft-dev libxext-dev \
libglew-dev libjpeg-dev libpng-dev libtiff-dev libgif-dev \
libxml2-dev libssl-dev libfftw3-dev \
qtbase5-dev qtchooser qttools5-dev-tools qt3d5-dev \
libgl1-mesa-dev libglu1-mesa-dev libxmu-dev
```

4.1.2 Arch Linux

4.1.2.1 Update Your System

```
sudo pacman -Syu --noconfirm
```

4.1.2.2 Install Required Dependencies

```
sudo pacman -S --noconfirm \
cmake gcc binutils \
libx11 libxpm libxft libxext glew \
libjpeg-turbo libpng libtiff giflib \
libxml2 openssl fftw \
qt5-base qt5-tools mesa glu libxmu
```

4.1.3 Fedora

4.1.3.1 Update Your System

```
sudo dnf upgrade --refresh -y
```

4.1.3.2 Install Required Dependencies

```
sudo dnf install -y \
cmake cmake-curses-gui cmake-gui gcc gcc-c++ binutils \
qt5-qtbase-devel qt5-qttools-devel qt5-qt3d-devel \
glew-devel libjpeg-turbo-devel libpng-devel libtiff-devel giflib-devel \
libxml2-devel openssl-devel fftw-devel \
mesa-libGL-devel mesa-libGLU-devel libXmu-devel
```

4.1.4 openSUSE

4.1.4.1 Update Your System

```
sudo zypper refresh
sudo zypper update -y
```

4.1.4.2 Install Required Dependencies

```
sudo zypper install -y \
cmake cmake-curses-gui cmake-gui gcc gcc-c++ \
libX11-devel libXpm-devel libXft-devel libXext-devel \
glew-devel libjpeg-devel libpng-devel libtiff-devel giflib-devel \
libxml2-devel libopenssl-devel fftw3-devel \
libqt5-qtbase-devel libqt5-qttools-devel libqt5-qt3d-devel \
Mesa-libGL-devel Mesa-libGLU-devel libXmu-devel
```

4.1.5 Rocky Linux, RHEL

4.1.5.1 Update Your System

```
sudo dnf upgrade --refresh -y
```

4.1.5.2 Install Required Dependencies

```
sudo dnf install -y \
cmake cmake-curses-gui cmake-gui gcc gcc-c++ binutils \
libX11-devel libXpm-devel libXft-devel libXext-devel \
glew-devel libjpeg-turbo-devel libpng-devel libtiff-devel giflib-devel \
libxml2-devel openssl-devel fftw-devel \
qt5-qtbase-devel qt5-qttools-devel qt5-qt3d-devel \
mesa-libGL-devel mesa-libGLU-devel libXmu-devel
```

4.2 Create Geant4 Installation Folders

To keep things neat and tidy, let's create dedicated folders for the Geant4 source and build files. It's all about organization — trust me, you'll thank yourself later:

```
mkdir Software
cd Software
mkdir Geant4
cd Geant4
```

4.3 Download Geant4 Source File

The latest stable version is 11.3.1. Get it here: `https://geant4.web.cern.ch/download/11.3.1.html`. Or just run:

```
wget https://gitlab.cern.ch/geant4/geant4/-/archive/v11.3.1/geant4-v11.3.1.tar.gz/
```

This magically downloads the Geant4 source to your directory.

4.4 Extract the Source Archive

Now, unzip the archive:

```
tar -xzf geant4-v11.3.1.tar.gz
```

Where:

- `-x`: Extracts files. Simple.
- `-z`: Uncompresses with gzip. Because who doesn't love uncompressed files?
- `-f`: Specifies the filename. You knew that, right?

4.5 Configure the Build

First, create a build directory and step into it:

```
mkdir geant4-v11.3.1-build
cd geant4-v11.3.1-build
```

Next, run the CMake tool like a pro:

```
ccmake ../geant4-v11.3.1
```

A fancy graphical interface will pop up. Enjoy the show.

4.5.1 Configuration:

At the bottom of the ‘ccmake‘ window, press the `c` key to configure the build. Easy, right?

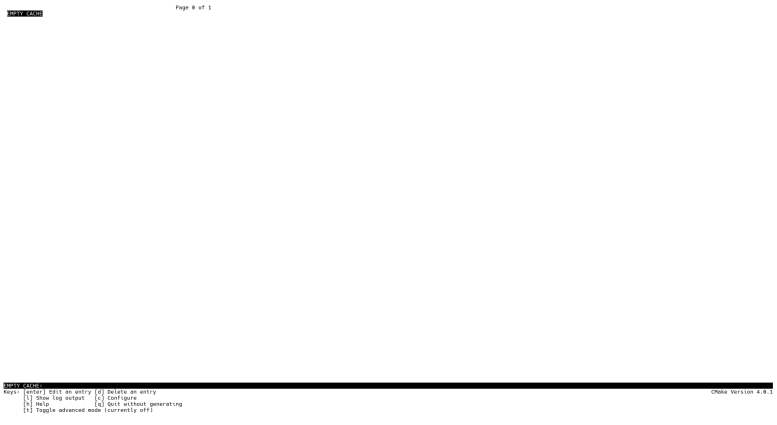

Figure 1: ccmake Initial Configuration Screen (Example)

4.5.2 Exiting Configuration:

Once the initial config is done, a summary will pop up. Press the `e` key to exit. Don't worry, it's not going to bite.

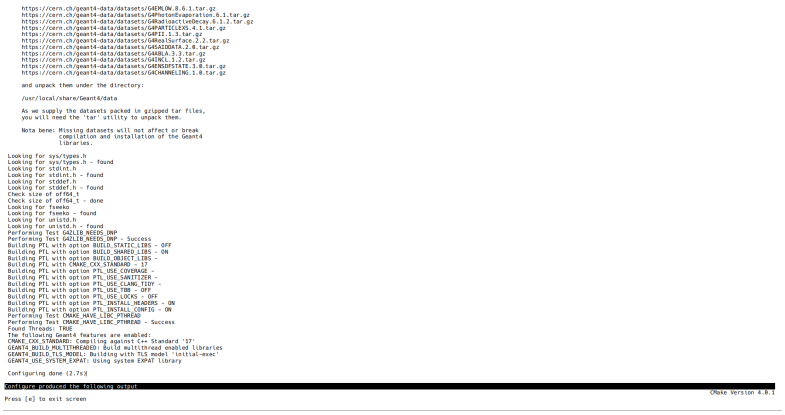

Figure 2: ccmake Configuration Summary (Example)

4.5.3 Build Options:

You'll be back at the main 'ccmake' screen. Don't panic — yes, there are a lot of options, but you'll manage.

Figure 3: ccmake Main Build Options (Example)

4.5.4 Setting Installation Path and Enabling Options:

First, navigate to the `CMAKE_INSTALL_PREFIX` option (it's usually the second one) with your trusty up and down arrows. Press `Enter` to edit it.
Enter your desired installation path (e.g., `/home/your_user/Geant4/geant4-v11.3.1-install`) and hit `Enter` to confirm. The path should magically update.
Now, enable these options by arrowing over to them and pressing `Enter` to toggle them to `ON`:

- `GEANT4_INSTALL_DATA`
- `GEANT4_USE_OPENGL_X11`
- `GEANT4_USE_QT`
- `GEANT4_USE_RAYTRACER_X11`

Figure 4: ccmake Setting Installation Prefix and Options (Example)

4.5.5 Final Configuration and Generation:

Press the `c` key again to apply the new settings. You might need to press it a few times until it stops nagging you about configuration changes. Once that's done, you'll see the magical `g` option at the bottom to generate the build files.

Figure 5: ccmake Final Configuration and Generate (Example)

4.6 Generate Build Files:

When you finally see the option to press `g` (generate), go ahead and do it. This will create the build files in your 'geant4-v11.3.1-build' directory. Simple, right?

4.7 Build and Install Geant4

Time to compile Geant4. Use the `make` command with the `-j12` option to use 12 CPU cores. If you have fewer (or more), adjust accordingly. Or, if you're not sure how many cores you have, just type `nproc` and hit enter—whatever number pops up, use that. Simple, right?

```
make -j12
```

After the build finishes successfully, install Geant4 with:

```
make install
```

5 Setup Environment

To make your life easier, add an alias in your '.bashrc' (or '.zshrc' if you're feeling fancy). Open the file with:

```
vim /home/your_user/.bashrc  # (Replace 'your_user' with your actual username)
```

Then, add this line (don't forget to adjust the path if you did something unusual with your install):

```
alias geant4make="source /home/your_user/Software/Geant4/geant4-v11.3.1-install
/share/Geant4/geant4make/geant4make.sh"
```

Save and close. Apply the changes by running:

```
source /home/your_user/.bashrc # (Replace 'your_user' with your actual username)
```

Or just restart your terminal. Test it by running:

```
geant4make
```

If you get no errors, congratulations, you're done. It works.

6 Verify Installation with Basic Example

Congrats, you made it! Let's check if Geant4 is actually working.
First, navigate back to 'Software/Geant4':

```
cd ../.. # Go up two levels
ls
```

You should see the 'geant4-v11.3.1-install' directory. Now, dive into the B1 example:

```
cd geant4-v11.3.1-install
cd share
cd Geant4
cd examples
cd basic
cd B1
```

Create a 'build' directory and go inside:

```
mkdir build
cd build
```

Configure and build the example:

```
cmake ..
make
```

If that goes without a hitch, list the contents of 'build':

```
ls
```

You should see an executable called 'exampleB1'. Run it:

```
./exampleB1
```

If you're greeted with a graphical output like the one below, your Geant4 installation is working fine.

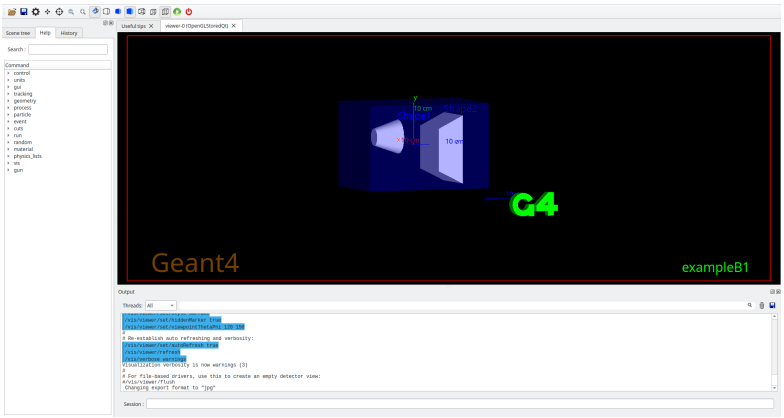

Figure 6: Example B1 Visualization

7 Conclusion

Well, look at you! You've successfully navigated through the manual installation of Geant4 on a Linux system. Just remember to tweak commands and paths to match your system. If things go sideways, don't panic—refer to the official Geant4 Installation Guide: `https://geant4.cern.ch/manuals/installationguide/`.
You're now ready to dive into the Geant4 toolkit and explore its overwhelming power!

Feedback and Contact: If you somehow have questions, suggestions, or if something in this guide didn't quite work out, feel free to drop me a line at `changdeprathemesh@gmail.com`.