

Geant4-V11 を Mac にインストールする手順書

(概要)

Homebrew の環境下に Geant4 version11 をインストールする手順を説明します。Geant4 に必要なコンパイラは Xcode を用います。その他のライブラリ関係は Homebrew を使って導入します。

(作業の流れ)

- 1) Xcode をインストールする
- 2) Geant4 に必要なライブラリ等を Homebrew でインストールして環境設定する
- 3) Geant4 のインストール

(設定手順)

1) Xcode をインストールする

- ・ App Store で Xcode を検索してインストールしてください。
- ・ Xcode インストール後に、Xcode コマンドラインツールを導入します。ターミナルを起動して、次のコマンドを入力します。表示されたダイアログで「install」をクリックします。

```
$ xcode-select --install
```

- ・ xcode 使用のライセンス許諾を行います。表示にしたがって「agree」を入力します。

```
$ sudo xcodebuild -license
```

2) Geant4 に必要なライブラリ等を Homebrew でインストールして環境設定する

- ・ Homebrew をインストールします。Homebrew のホームページ(<https://brew.sh>)に表示されている次のコマンドをコピペして、ターミナルで実行します。コマンドは1行で入力します。

```
$ /bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- ・ インストール後に、ターミナルで次のコマンドを入力します。

```
$ export PATH=/opt/homebrew/bin:$PATH
```

- ・ cmake をインストールします。

```
$ brew install cmake
```

- ・ Qt5 をインストールします。

```
$ brew install qt@5
```

*現状の Geant4 では、qt5 のバージョンまでのサポートです。qt6 がインストールされていると Geant4 のコンパイルがエラーで失敗します。qt5 以外のバージョンはアンインストールしてください。

- Xerces-C++をインストールします。(GDML のためのオプションです)

```
$ brew install xerces-c
```

- CERN ROOT をインストールします。

```
$ brew install root
```

- Emacs エディタをインストールします。(他のエディタを利用できる方は不要です)

```
$ brew install emacs
```

- XQuartz をインストールします。

ダウンロード URL: <https://www.xquartz.org>

X Quartz-2.8.5.pkg をダウンロードしてインストールします。

- ここまでにインストールしたソフト等のパスを.zshrc や.bashrc に記載してログイン時に自動設定される様にします。

```
$ emacs ~/.zshrc (自分でシェルを変更していなければ、こちらになります。)
```

または

```
$ emacs ~/.bashrc (自分でシェルを bash に変更している場合)
```

- ファイルの末尾に次の行を追加してください。

(* インストール先を確認してください。Finder を開いて、[Command]+[SHIFT]+[G] を押して表示されるテキストボックスに /opt と入力すると、/opt フォルダのファイルや下層にあるフォルダを確認することができます。下記の例と異なる場合がありますので確認することを勧めます。)

```
export PATH=/opt/homebrew/bin:$PATH
export PATH=/opt/homebrew/opt/qt@5/bin:$PATH
#
. /opt/homebrew/bin/thisroot.sh
#
```

- 編集したら、Ctrl-x, Ctrl-s で保存します。その後に Ctrl-x, Ctrl-c でエディタを終了します。

- 新しいターミナルを開きます。このとき、エラーが出ていないか確認してください。

エラーが出ていた場合は、パスの記載が間違っている可能性が高いです。再確認しましょう。

3) Geant4 のインストール

- ・ソースコードをダウンロードしてします。Windows のブラウザでダウンロードしましょう。 ([geant4-v11.3.0.tar.gz](https://geant4.org/download/))

ダウンロード: <https://geant4.org/download/>

「Source code」の「Download tar.gz」のボタンをクリックしてダウンロードします。

- ・ファイルを展開します。今回は、ホームディレクトリに geant4 というフォルダを作成して、その中に展開します。 (*){user}の部分は自分のユーザ名です。

```
$ cd
$ mkdir geant4
$ cd geant4
$ tar zxvf /Users/{user}/Downloads/geant4-v11.3.0.tar.gz
(または $ tar xvf /Users/{user}/Downloads/geant4-v11.3.0.tar )
```

- ・cmake を実行します。下記のコマンドで¥マークがついている部分は、¥マークや改行を入れずに 1 行で続けて入力してくださいという意味です。

```
$ pwd
/Users/{user}/geant4 と表示されることを確認してください。
$ mkdir geant4-v11.3.0-build
$ mkdir geant4-v11.3.0-install
$ cd geant4-v11.0.3-build
$ cmake -DCMAKE_INSTALL_PREFIX=../geant4-v11.3.0-install ¥
        -DGEANT4_INSTALL_DATA=ON ¥
        -DGEANT4_USE_OPENGL_X11=ON ¥
        -DGEANT4_USE_QT=ON ¥
        -DGEANT4_USE_GDML=ON ¥
        ../geant4-v11.3.0
```

*cmake を実行し終えた時に、エラーや Not found の様な出力表示が出ていないか確認してください。もしもエラーなどが出ていた場合には、コマンドの打ち間違いの可能性が高いです。その場合には、フォルダ内のファイルを消したのちに cmake を再実行してください。

```
$ pwd
/home/{user}/geant4/geant4-v11.1.2-build であることを確認する。
$ rm -rf * (ファイルを削除)
$ cmake -DCMAKE_INSTALL_PREFIX=.... (再確認して再実行)
```

- ・ビルドします。

```
$ make -j 4    (4 はスレッド数なので変更可)
```

*make-j で指定するスレッド数は、コンピュータの性能で決めてください。但し、10 などを指定した場合は、ファイルの順番が崩れるのかエラーになる場合がある様です。その場合には、スレッド数を減らして再度、make-j を実行してください。

*ビルドのエラー等がないかは確認してください。

- ・インストールします。

```
$ make install
```

- ・ログイン時の設定ファイルに環境変数を設定します。emacs エディタで編集します。

```
$ emacs ~/.zshrc
```

または

```
$ emacs ~/.bashrc
```

- ・開いたファイルの末尾に次の行を追加して保存します。

```
. /Users/{user}/geant4/geant4-v11.3.0-install/bin/geant4.sh
```

Ctrl キーを押しながら、s キーを押して保存します。その後に、Ctrl キーを押しながら、x キー、c キーを続けて押して emacs エディタを終了します。

- ・ターミナルを閉じます。

```
$ exit
```

- ・ターミナルを開き直します

*ログイン時にエラーが表示されないかを確認します。エラーがあれば、.bashrc の編集ミスと思われます。綴り間違いや、geant4 のパスに間違いがある可能性が高いのでエディタで確認して修正してください。

- ・設定された環境変数を確認します。

```
$ printenv | grep -i Geant4
```

...

```
GEANT4_DATA_DIR=/Users/.....
```

*GEANT4_DATA_PATH が出力されていれば OK です。

4) Geant4 サンプルコードの実行確認

サンプルコードをビルドして実行できることを確認します。ホームディレクトリに work というフォルダを作って、その中で作業を行います。

```
$ cd
$ mkdir work
$ cd work
$ cp -r ~/geant4/geant4-v11.3.0/examples/basic/B4/B4c .
$ mkdir B4c-build
$ cd B4c-build
$ cmake ../B4c
$ make -j 4
$ ./exampleB4c
```

・QT ウィンドウの「Session:」に次の入力を行って飛跡を確認してみます。

```
/run/beamOn 10
(飛跡が表示されるか確認しましょう)
exit (終了します)
```

・スレッドモードでのマクロファイル実行も試してみましょう。-t で4 スレッドを指定して、マクロファイル run2.mac を実行します。

```
$ ./exampleB4c -t 4 -m run2.mac
```

画面表示の中に、G4WT0～G4WT3 までの表示があり、4 スレッドで実行していると確認できます。

・実行後の出力ファイル名を確認します。

```
$ ls -l
B4.root というファイルが、実行後の時間にできたことが確認できます。
```

補足 Homebrew のコマンド

・バージョンを確認する

```
$ brew --version
```

・Homebrew をアップデート

```
$ brew update
```

・インストール済みのパッケージの更新

```
$ brew upgrade
```