

---

# cmakeと演習プログラムのビルド

---

Geant4 10.3.P3準拠

Geant4 HEP/Space/Medicine 講習会資料



大学共同利用機関法人  
高エネルギー加速器研究機構

# 本資料に関する注意

---

- 本資料の知的所有権は、高エネルギー加速器研究機構およびGeant4 collaborationが有します
- 以下のすべての条件を満たす場合に限り無料で利用することを許諾します
  - 学校、大学、公的研究機関等における教育および非軍事目的の研究開発のための利用であること
    - ・ Geant4の開発者はいかなる軍事関連目的へのGeant4の利用を拒否します
  - このページを含むすべてのページをオリジナルのまま利用すること
    - ・ 一部を抜き出して配布したり利用してはいけません
  - 誤字や間違いと疑われる点があれば報告する義務を負うこと
- 商業的な目的での利用、出版、電子ファイルの公開は許可なく行えません
- 本資料の最新版は以下からダウンロード可能です
  - <http://geant4.kek.jp/lecture/>
- 本資料に関する問い合わせ先は以下です
  - Email: [lecture-feedback@geant4.kek.jp](mailto:lecture-feedback@geant4.kek.jp)

# 目次

---

1. プログラム ビルド入門
2. Geant4でのcmakeの使い方
3. cmake使用のまとめ

# プログラム ビルド入門

---

# ビルド・ツールとは何か？

---

## ■ ビルドとはプログラムのソースコードから実行モジュールを作成すること

- 単純なプログラムではビルドは簡単

```
$ g++ helloWorld.cc  
$ a.out
```

- しかし、Geant4のアプリケーションのような膨大な数のソース・ファイルから成るプログラムではビルドは複雑で間違いを起きやすい

## ■ ビルド・ツールとは複雑なビルド手続きを自動化させるもの

- スクリプトに複雑な手続きを記述して、それを実行させることでビルドを実行

## ■ Linux環境でのビルドツール

- makeが標準的なツールでGNUmakeが最も一般的に使われる
- Makefileとよばれるファイルにmake言語を用いてビルドの手続きを書く
- Geant4もversion9.4以前ではGNUmakeを直接使っていた

## ■ Linux環境以外でのビルドツール

- Windows: Visual Studioが用意するビルド環境
- Mac OSX: Xcodeが用意するビルド環境
- etc

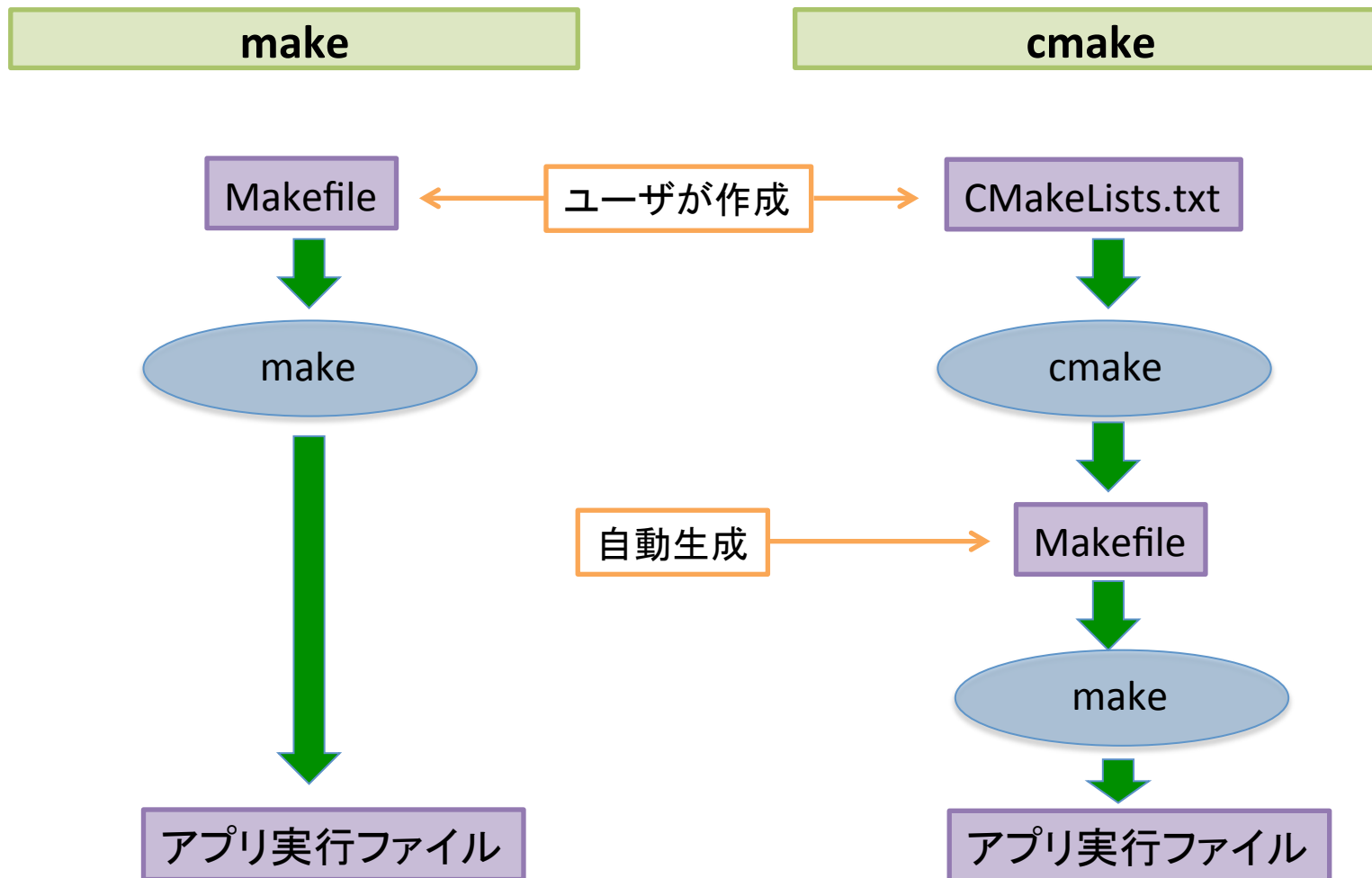
# makeとcmake

---

- **make**はプログラムのビルドを自動化してくれるが、異なる計算環境(linux/Unix, Windows, Mac OSX, etc)で統一的に使えるものではない
- **cmake**は異なる計算機環境でのビルドを一元的に管理するツール
  - cmakeを使えば、Geant4のビルド(ライブラリ、アプリケーションを含めて)を異なる計算機環境で統一的行える
  - cmakeではCMakeLists.txtとよばれるファイルにcmakeコマンドを用いてビルドの手続きを書く
  - Geant4はversion 9.5からcmakeを採用
  - GNUmakeを使ったビルド手法はversion 10.0以降はサポートしていないので必ずcmakeを使うこと
- cmakeはlinux/unix環境ではMakefileを自動生成する
  - Windows環境ではmakeが使えないので、cmakeコマンドがVisual Studioのビルド・コマンドにマッピングされる

## makeとcmake - 続き

### ■ Linux環境でのmakeとcmakeの使用法の違い



# Geant4でのcmakeの使い方

---



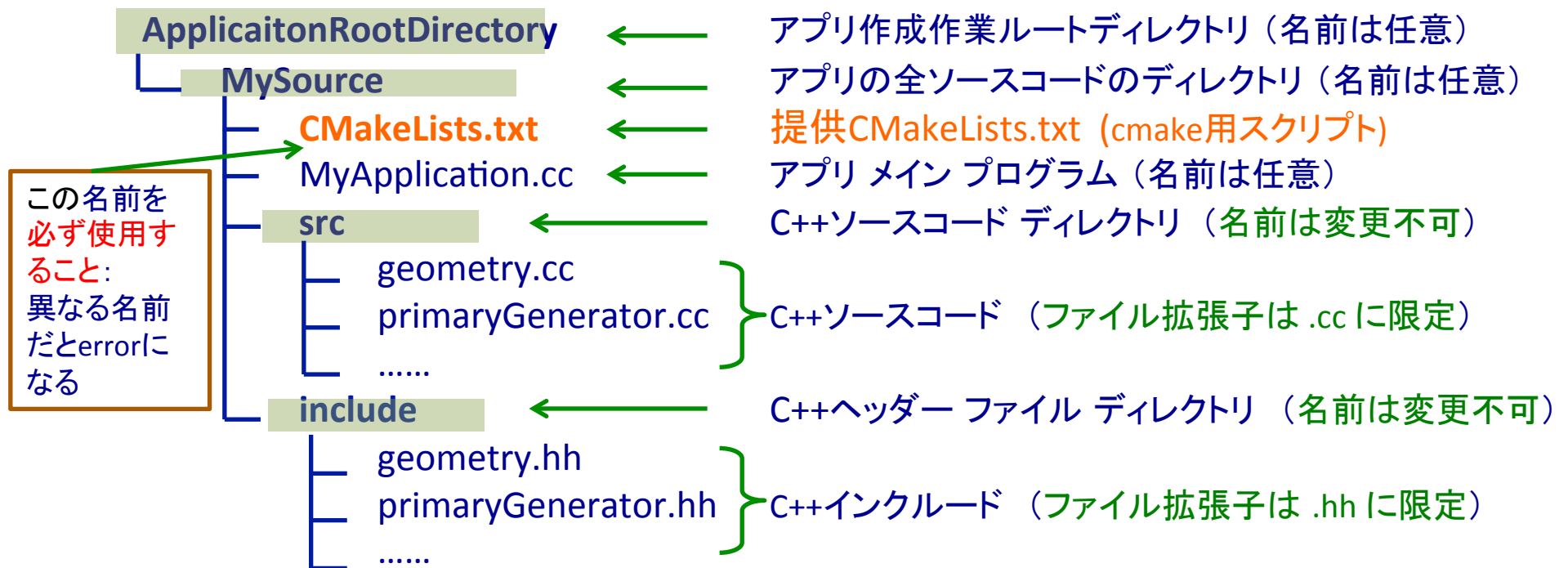
# はじめに

---

- 本講義ではcmakeおよびmakeの詳細には立ち入らない
  - cmake言語、make言語はそれ自体が複雑で習得には時間がかかる
  - 本講習会で提供するCMakeLists.txtは汎用性を考慮して作られているので、ユーザが今後、独自のGeant4アプリケーションを作るときにも使える
    - ・ 最低のパラメータ設定は必要であるが、これに関しては後述
  - まずはblackboxとして提供CMakeLists.txtを使い、必要に応じて内容を勉強すること勧める
  
- makeおよびcmakeの標準参考文献をあげておく
  - cmake: <http://www.cmake.org/documentation/>
  - make: <http://www.gnu.org/software/make/manual/make.html>

# 演習で提供されるCMakeLists.txtを使うためのファイル環境

- 演習で提供されるCMakeLists.txtは下図のファイル構造で使うことを前提としている
  - アプリを作る場合、初心者ユーザはこの構造に従うことを推奨する
  - 演習プログラムは全てこのファイル構造で構成されている
  - cmake言語をある程度学べば、独自のファイル構造で作業ができるようになる



[注] 提供演習環境では拡張子が'.cc'および'.hh'でないファイルをcmakeは無視(ビルドの対象外)

# 提供CMakeLists.txtの使用方法

- 提供されているCMakeLists.txtを自分のアプリ作成に使用するときに変更しなければならない部分は以下の二つのパラメータのみ

- 好みのeditorでCMakeLists.txtを開きこれらのパラメータを設定するだけ
- 他の部分は一切触れる必要なし

```
#+++++
# CMakeLists.txt: Geant4 Tutorial for Hep/Space Users
#+++++
cmake_minimum_required(VERSION 2.6 FATAL_ERROR)

#-----
# Setup Section: You have to set the following two parameters
#-----
# Set the name of your main program
set( NAME_MAIN_PROGRAM Application_Main)

# Set the path to Geant4 library
set( Geant4_DIR /opt/geant4/10.3.2/lib64/Geant4-10.3.2)

#-----
# Intact Section: You don't need to touch the following part
#-----
# Set the project name
project( ${NAME_MAIN_PROGRAM} )
.....
```

## ① ここに自分のアプリのmainプログラムの名前を設定する

- `Set()`はcmake言語で変数の設定に使うコマンド
- mainプログラム名がMyAppli.ccならMyAppliがプログラム名
- `NAME_MAIN_PROGRAM`はcmake変数名で変更不可

## ② 使用するGeant4ライブラリへのパス設定

- `Geant4_DIR`はcmake変数名で変更不可
- `.bashrc`などで'`geant4.sh`'を実行していれば、この設定は不要

[注: cmakeのシンタックス]

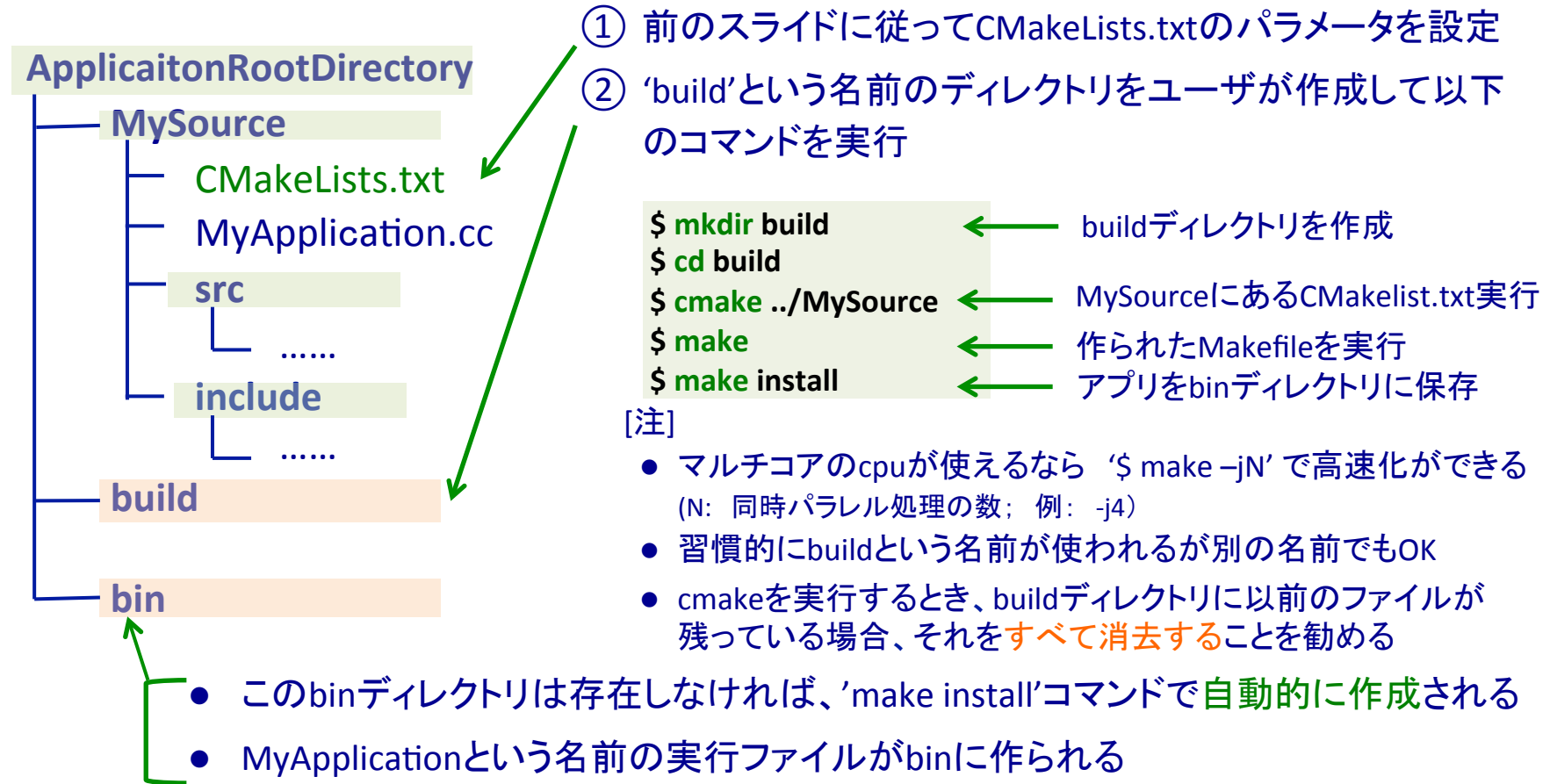
- コマンドは大小文字の区別なし(set=SET)
- 変数名、変数値は大小文字の区別あり

以降のコード部分は一切変更不要

→ **blackbox**として扱うのでOK

# 提供CMakeLists.txtを使ったビルドの手順

- ApplicationRootDirectoryのもとで以下の手順に従いビルドを実行すればよい



[注] このビルド手順はGeant4が公式提供しているExample Codesで使われているものとは異なる

← ここで採用している手順がより一般的で多くのユーザアプリに応用が可能

# cmake使用のまとめ

---

# cmakeを使用するための最低限知識まとめ

---

## ■ ソース ディレクトリ

- ビルドの対象となるソースファイルが置かれる
- ビルドの手続きを記述したCMakeLists.txtがディレクトリの直下に置かれる

## ■ ビルド ディレクトリ

- cmakeのワーク ディレクトリであり、ソース ディレクトリとは独立に作ることでお互いの干渉を排除できる
- ここに以下のようなファイルが自動的に作られ： ユーザは中身を知る必要なし
  - Makefile
  - ライブラリ
  - アプリ実行ファイル（make installでbinディレクトリにコピーされる）
  - 各種ビルド関連で作られるファイル
- cmakeを実行するとき、buildディレクトリに以前のファイルが残っている場合、それをすべて消去することを推奨

## ■ cmakeの使用手順

```
$ mkdir build; cd build  
$ cmake path_to_CMakeLists.txt  
$ make; make install
```

← 'make install'だけでもOK

# Geant4提供Examplesのビルド

- Geant4が提供するExample CodesをビルドするためのCMakeLists.txtは、このチュートリアルで使うものと異なる: Geant4公式提供の使い方は以下を参照

[http://geant4.web.cern.ch/geant4/UserDocumentation/Doxygen/examples\\_doc/html/README\\_HowToRun.html](http://geant4.web.cern.ch/geant4/UserDocumentation/Doxygen/examples_doc/html/README_HowToRun.html)

The screenshot shows the Geant4 website interface. At the top, there's a header with the Geant4 logo and navigation links: Download, User Forum, Gallery, and Contact Us. Below the header is a search bar. The main content area has a breadcrumb trail 'Home > Examples' and a navigation menu with tabs: Main Page, Related Pages (selected), Modules, Namespaces, Classes, and Files. The title of the page is 'How to build and run an example'. The text explains that novice, basic, and most extended examples have a similar structure. It mentions that the main() function is in exampleXYZ.cc, source code is in include and src subdirectories, and the executable takes the same name as the file without the .cc extension. It also notes that several macros are provided for running examples with various start-up conditions, and a macro exampleXYZ.in is used for testing and output. A link to the Geant4 Installation Guide is provided for more details. The page concludes with a section 'Compile and link to generate an executable' and a subsection 'With CMake' containing a code block with shell commands for building and installing an example.

```
% cd path_to_exampleXYZ      # go to directory which contains your example
% mkdir exampleXYZ_build
% cd exampleXYZ_build
% cmake -DGeant4_DIR=path_to_Geant4_installation/lib[64]/Geant4-10.1.0/ ../exampleXYZ
% make -j N exampleXYZ        # "N" is the number of processes
% make install                # this step is optional
```

[注]  
Geant4が提供する  
CMakeLists.txtは  
本講義で提供する  
ものに比べて汎用性  
が少ないことに注意