



An Introduction to CMake/CTest

Jan Engels

Advanced Programming Concepts Workshop 2012

Desy, 2012-10-11

What is CMake?

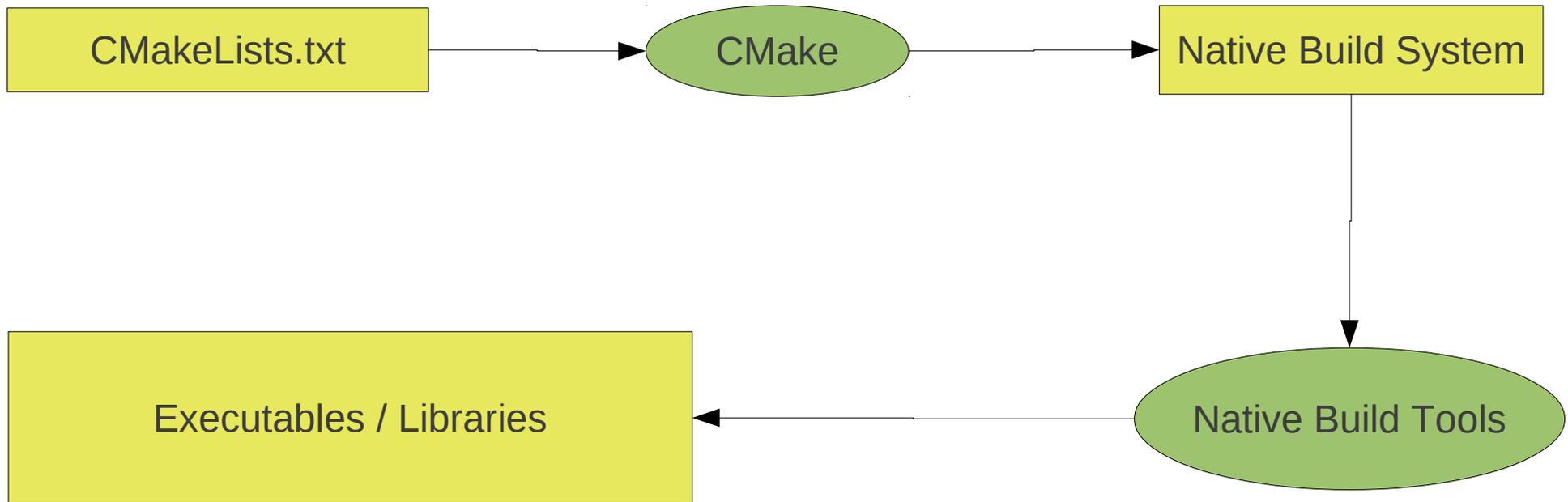


- CMake
 - is a tool for generating native build environments
 - UNIX/Linux -> Makefiles
 - Windows -> VS Projects/Workspaces
 - Apple -> XCode
 - Open-Source :)
 - Cross-Platform



- CMake's nice features:
 - Manage complex, large build environments (KDE4)
 - Simple, intuitive syntax
 - Flexible
 - Support for Macros
 - So called **CMake modules** are available for finding existing software
 - FindGSL.cmake, FindOpenGL.cmake, FindZLib.cmake ...
 - Can be customized, e.g. FindROOT.cmake
 - Create custom targets/commands
 - Support for regular expressions (*nix style)
 - Support for **out-of-source** builds
 - Cross Compiling
 - Integrated Testing & Packaging (CTest, CPack)

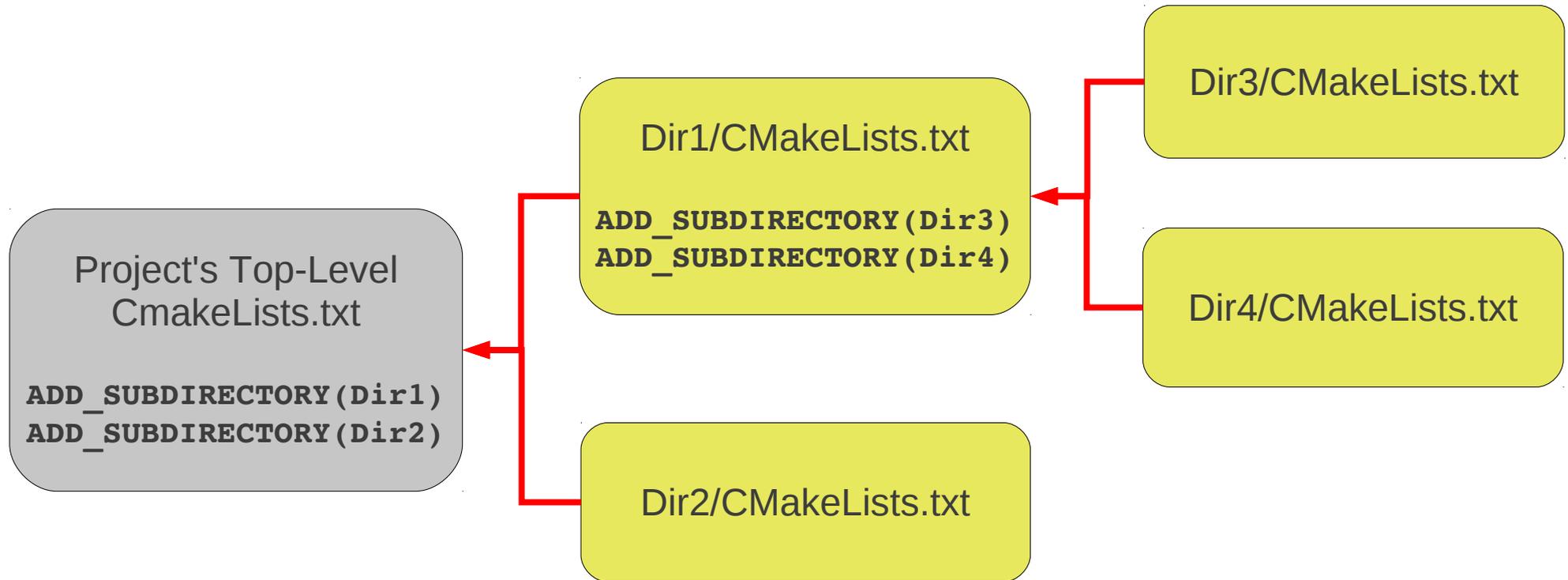
CMake's build system generator





- **CMakeLists.txt**
 - Input text files that contain the project parameters and describe the flow control of the build process in simple CMake language.
- **Source Tree** (where source files are located)
 - CmakeLists.txt
 - Source files (hello.cc)
 - Header files (hello.h)
- **Binary Tree** (where build files are located)
 - Native build system files (Makefiles)
 - Output from build process:
 - Libraries, executables
 - Any other build generated files
- Source and Binary trees may be:
 - In the same directory (**in-source build**)
 - In different directories (**out-of-source build**)

- Subdirectories added with **ADD_SUBDIRECTORY**
- Child inherits from parent (lacking feature in traditional Makefiles)
- Order of processing: Dir1;Dir3;Dir4;Dir2
 - When CMake finds an ADD_SUBDIRECTORY command it stops processing the current file and goes down the tree branch





- CMake variables
 - CMAKE_INSTALL_PREFIX
 - Where to put files when calling 'make install'
 - CMAKE_BUILD_TYPE
 - Type of build (Debug, Release, ...)
 - BUILD_SHARED_LIBS
 - Switch between shared and static libraries
 - CMAKE_MODULE_PATH
 - Path to find own written cmake modules (e.g. FindMyPackage.cmake)

- CMake variables can be set in the CMakeLists.txt and dynamically changed on the command line as follows:
 - `cmake -DBUILD_SHARED_LIBS=OFF`
 - GUI also available: `ccmake`



- **Create a build directory** (“out-of-source-build” concept)
 - `mkdir build ; cd build`
- **Configure**
 - `cmake [options] <source_tree>`
- **Build**
 - `make`
- **Install**
 - `make install`
- The Build step is automatically called with 'make install'

Similar to Auto-tools

Hello World project using CMake (1)



- Top-level project directory:
 - CMakeLists.txt
 - Sub-directory `hello_lib`:
 - CMakeLists.txt
 - `hello.h`
 - `hello.cc`
 - Sub-directory `hello_app`:
 - CMakeLists.txt
 - `main.cc`

```
/*hello.h*/  
#ifndef _hello_h  
#define _hello_h  
  
class Hello {  
public:  
    void Print();  
};  
  
#endif
```

```
/*hello.cc*/  
#include "hello.h"  
#include <iostream>  
using namespace std;  
  
void Hello::Print() {  
    cout<<"Hello,  
World!"<<endl;  
}
```

Hello Library

```
/*main.cc*/  
#include <iostream>  
#include "hello.h"  
  
int main() {  
    Hello().Print();  
    return 0;  
}
```

HelloWorld Application

Hello World project using CMake (2)



```
# Top-Level CMakeLists.txt
```

```
PROJECT( HelloWorld )
```

```
ADD_SUBDIRECTORY( hello_lib )
```

```
ADD_SUBDIRECTORY( hello_app )
```

```
# CMakeLists.txt in hello_lib dir
```

```
# Adds a library called Hello (libHello.a under Linux) from the source file hello.cc
```

```
ADD_LIBRARY( Hello hello )
```

```
# CMakeLists.txt in hello_app dir
```

```
# Make sure the compiler can find include files from our Hello library.
```

```
INCLUDE_DIRECTORIES(${PROJECT_SOURCE_DIR}/Hello)
```

```
# Add binary called "helloWorld" that is built from the source file "main.cc".
```

```
# The extension is automatically found.
```

```
ADD_EXECUTABLE(helloWorld main)
```

```
# Link the executable to the Hello library.
```

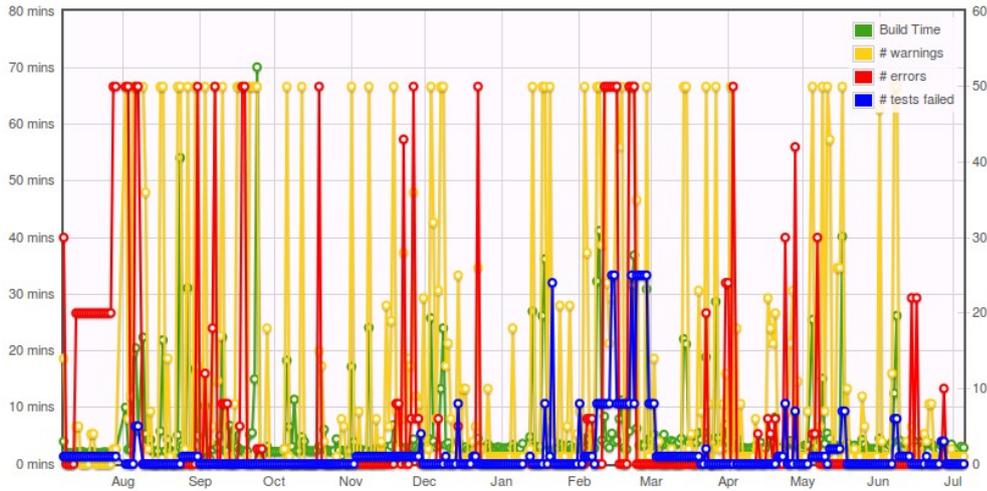
```
TARGET_LINK_LIBRARIES(helloWorld Hello)
```



- CTest
 - Integrated testing in CMake
 - Uses flexible CMake syntax
 - Support for regular expressions (*nix style)
 - Support for running custom commands
 - Dashboard (CDash) for displaying test results in web page

 - More on live demo..

- CDash



Warnings	Configure Errors	Configure Warnings	Build Errors	Build Warnings	Tests Failed
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	2	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	10	1	3
0	0	0	0	1	3
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	8	0
0	0	0	0	8	0
0	0	0	0	3	1
0	0	0	0	4	0
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	22	1	0
0	0	0	0	2	0
0	0	0	22	5	0
0	0	0	0	1	1
0	0	0	0	3	1
0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	50	6
1	0	0	0	50	6
0	0	0	0	12	0
0	0	0	0	1	0
2012-06-23 02:10:23	0	0	0	1	0
2012-06-22 02:10:22	0	0	0	8	0
2012-06-21 02:10:21	0	0	0	8	0
2012-06-20 02:10:20	0	0	0	3	1
2012-06-19 02:10:19	0	0	0	4	0
2012-06-18 02:10:18	0	0	0	1	0
2012-06-17 02:10:17	0	0	0	1	0
2012-06-16 02:10:16	0	0	0	1	0
2012-06-15 02:10:15	0	0	0	2	0
2012-06-14 02:10:14	0	0	0	5	0
2012-06-13 02:10:13	0	0	0	1	1
2012-06-12 02:10:12	0	0	0	3	1
2012-06-11 02:10:11	0	0	0	1	1
2012-06-10 02:10:10	0	0	0	1	1
2012-06-09 02:10:09	0	0	0	1	1
2012-06-08 02:10:08	0	0	0	50	6
2012-06-07 02:10:07	1	0	0	50	6
2012-06-06 02:10:06	0	0	0	12	0
2012-06-05 04:24:05	0	0	0	1	0



- CDash

Nightly										
Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
it-xenvm027	Linux-c++	0	0	0	0	1	0	1	28	13 hours ago
grid-ilc-pa0	Linux-c++	0	0	0	0	1	0	0	29	13 hours ago

My Projects						
Project Name	Actions	Buils	Buils per day	Success Last 24h	Errors Last 24h	Warnings Last 24h
CED		1134	2	2	0	0
CEDViewer		1122	2	2	0	0
ForwardTracking		263	2	2	0	0
GEAR		1127	2	2	0	0
iLCTest		1700	2	0	0	2
KalDet		1118	2	2	0	0
KalTest		1122	2	2	0	0
LCCD		1126	2	2	0	0
LCIO		1122	2	2	0	0
Marlin		1126	2	2	0	0
MarlinReco		1110	2	2	0	0
MarlinTPC		984	2	2	0	0
MarlinTrk		928	2	2	0	0
MarlinUtil		1123	2	2	0	0
Overlay		1110	2	2	0	0
RAIDA		1130	2	2	0	0



- References:
 - <http://www.cmake.org>
 - <http://www.cdash.org>
 - **Mastering CMake**
 - Ken Martin, Bill Hoffman
 - Published by Kitware, Inc.
 - ISBN: 1-930934-16-5

