

CMake Internals

Roger Leigh

Tuesday 28th October 2014
University of Dundee



wellcometrust
Strategic Award

Overview

Overview

Build systems

- cmake introduction

cmake basics

- Simple program

- Simple library

- Language features

Available build systems

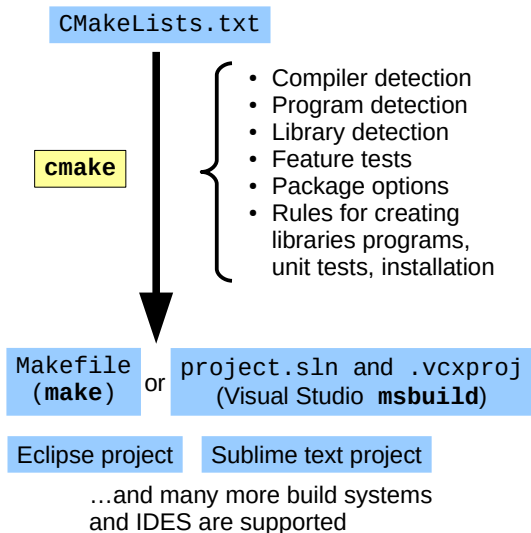
There are many available build systems, which include:

- ▶ Make and GNU Make
- ▶ GNU Autotools
- ▶ CMake
- ▶ Qt qmake
- ▶ SCons
- ▶ Jam / BJam
- ▶ Ant / Maven / Gradle

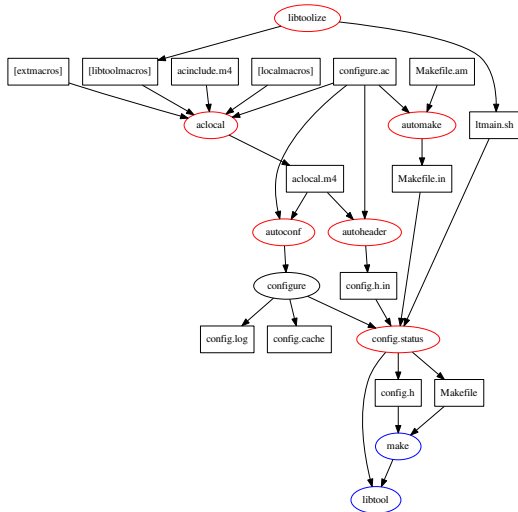
cmake features

- ▶ `cmake` is a generic cross-platform build system
- ▶ `cmake` generates build files for a large number of common build systems
- ▶ On FreeBSD, Linux and MacOS X, `make` Makefiles will be used
- ▶ On Windows with Visual Studio, `msbuild .sln` solution files will be used
- ▶ Eclipse, Sublime Text, Kate, Code::Blocks or several other IDEs or build systems may be used instead, if desired

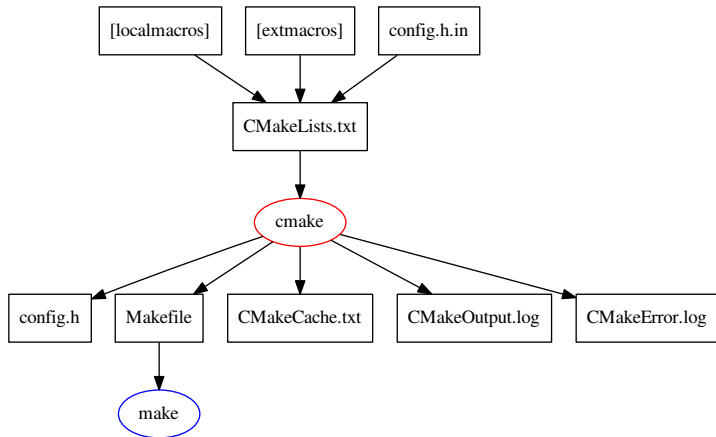
cmake overview



Autotools overview



cmake overview



A simple program

```
cmake_minimum_required(VERSION 2.8)
cmake_policy(VERSION 2.8.0)

project("test-program")

enable_language(CXX)

add_executable(test-program test.cpp)
```

- ▶ Compiles test.cpp into the program test-program

A simple library

```
project("test-library")

enable_language(CXX)

add_library(test-library SHARED test.cpp test.h)
set_target_properties(test-library PROPERTIES VERSION
    "1.0.0")

add_executable(test-program main.cpp)

target_link_libraries(test-program test-library)
```

- ▶ Compiles `test.cpp` into the library `test-library`
- ▶ Adds ABI version number to `test-library`
- ▶ Compiles `main.cpp` into the program `test-program`
- ▶ Links `test-program` with `test-library`

Variables, conditionals, loops

```
set(var value)
if(var)
  message(STATUS "The value of var is ${var}")
else()
  message(WARNING "var is unset")
endif()

set(var "a;b;c")
list(APPEND var d e)
foreach(v ${var})
  message(STATUS "List item value is ${v}")
endforeach()
```

- ▶ Variables are lists of strings of 0, 1 or multiple values

Feature tests

```
find_program(DOXYGEN_EXECUTABLE doxygen DOC "Doxygen  
API documentation tool")  
  
find_path(BOOST_INCLUDE_DIR  
          NAMES "boost/filesystem.hpp"  
          DOC "Boost.Filesystem header directory")  
  
find_library(XERCES_LIBRARY xerces-c  
            DOC "Xerces-C shared library")
```

- ▶ Feature tests probe system capabilities and adapt the build to the system

Built-in feature tests

```
find_package(Doxygen "1.7.0")  
find_package(Boost REQUIRED COMPONENTS filesystem  
            system)  
find_package(Xerces REQUIRED)
```

- ▶ CMake provides an extensive set of feature tests
- ▶ Custom feature tests can be written if needed

Advanced feature tests

```
include(CheckCXXSourceCompiles)

check_cxx_source_compiles("
void foo() noexcept{}
int main() { foo(); }
" HAVE_NOEXCEPT)

check_cxx_source_compiles("
#include <array>
int main() { std::array<int,3> a; a[0] = 5; }
" HAVE_ARRAY)
```

- ▶ Checking for header or library existence is sometimes insufficient
- ▶ Checks can compile, link and execute test code
- ▶ Test specific implementation details

Package options

```
option(test "Enable unit tests (requires gtest)" ON)
set(BUILD_TESTS ${test})
option(extended-tests "Enable extended tests (more
    comprehensive, longer run time)" ON)
set(EXTENDED_TESTS ${extended-tests})

message(STATUS "Build tests: ${BUILD_TESTS}")
message(STATUS "Extended tests: ${EXTENDED_TESTS}")
```

- ▶ User-configurable options
- ▶ Customise any aspect of CMake operation

Unit tests

```
enable_testing()

enable_language(CXX)

add_executable(test-program test.cpp)

add_test(simple-test/test-program test-program)
```

- ▶ Unit tests are simple programs

Installation

```
include(GNUInstallDirs)

install(TARGETS test-library LIBRARY
        DESTINATION "${CMAKE_INSTALL_FULL_LIBDIR}")

install(FILES test.h
        DESTINATION "${CMAKE_INSTALL_FULL_INCLUDEDIR}")

install(TARGETS test-program RUNTIME
        DESTINATION "${CMAKE_INSTALL_FULL_BINDIR}")
```

- ▶ Any target or file can be installed
- ▶ Installation prefix or any install category may be customised

Acknowledgements

- ▶ OME Team, Dundee
 - ▶ Jason Swedlow
 - ▶ Jean-Marie Burel
 - ▶ Mark Carroll
 - ▶ Andrew Patterson
 - ▶ ...and the rest of the team
- ▶ Micron, Oxford
 - ▶ Douglas Russell
- ▶ Glencoe Software
 - ▶ Melissa Linkert
 - ▶ Josh Moore



wellcometrust

Strategic Award