Contents

CHAPTER 1 WHY CMAKE? 1

1.1 The History of CMake 3
1.2 Why Not Use Autoconf? 3
1.3 Why Not Use JAM, qmake, SCons, or ANT? 4
1.4 Why Not Script It Yourself? 4
1.5 On What Platforms Does CMake Run? 5

CHAPTER 2 GETTING STARTED 7

2.1 Getting and Installing CMake on Your Computer 7
UNIX and Mac Binary Installations 7
Windows Binary Installation 7
2.2 Building CMake Yourself 8
2.3 Basic CMake Usage and Syntax 8
2.4 Hello World for CMake 9
2.5 How to Run CMake? 10
Running CMake's Qt Interface 11
Running the ccmake Curses Interface 13
Running CMake from the Command Line 15
Specifying the Compiler to CMake 15
Dependency Analysis 16
2.6 Editing CMakeLists Files 17
2.7 Setting Initial Values for CMake 17
2.8 Building Your Project 19

CHAPTER 3 KEY CONCEPTS 21

3.1 Main Structures 21
3.2 Targets 24
3.3 Source Files 25
3.4 Directories, Generators, Tests, and Properties 26
3.5 Variables and Cache Entries 27
3.6 Build Configurations 32

CHAPTER 4 WRITING CMAKELISTS FILES 33

4.1 CMake Syntax 33
4.2 Basic Commands 34
4.3 Flow Control 35
4.4 Regular Expressions 42

4.5 Checking Versions of CMake 44
4.6 Using Modules 45
Using CMake with SWIG 48
Using CMake with Qt 49
Using CMake with FLTK 50
4.7 Policies 50
Updating a Project For a New Version of CMake 53
4.8 Linking Libraries 57
Specifying Optimized or Debug Libraries with a Target 59
4.9 Shared Libraries and Loadable Modules 59
4.10 Shared Library Versioning 64
4.11 Installing Files 66
Installing Prerequisite Shared Libraries 76
4.12 Advanced Commands 82

CHAPTER 5 SYSTEM INSPECTION 85

5.1 Using Header Files and Libraries 85
5.2 System Properties 87
5.3 Finding Packages 92
5.4 Built-in Find Modules 93
5.5 How to Pass Parameters to a Compilation? 95
5.6 How to Configure a Header File 97
5.7 Creating CMake Package Configuration Files 99

CHAPTER 6 CUSTOM COMMANDS AND TARGETS 103

6.1 Portable Custom Commands 103
6.2 Using add_custom_command on a Target 105
How to Copy an Executable Once it is Built? 106
6.3 Using add_custom_command to Generate a File 107
Using an Executable to Build a Source File 107
6.4 Adding a Custom Target 108
6.5 Specifying Dependencies and Outputs 111
6.6 When There Isn't One Rule For One Output 112
A Single Command Producing Multiple Outputs 112
Having One Output That Can Be Generated By Different Commands 112

CHAPTER 7 CONVERTING EXISTING SYSTEMS TO CMAKE 115

7.1 Source Code Directory Structures 115
7.2 Build Directories 117
7.3 Useful CMake Commands When Converting Projects 119
7.4 Converting UNIX Makefiles 120

7.5 Converting Autoconf Based Projects 121
7.6 Converting Windows Based Workspaces 123

CHAPTER 8 CROSS COMPILING WITH CMAKE 125

8.1 Toolchain Files 126
Finding External Libraries, Programs and Other Files 128
8.2 System Inspection 130
Using Compile Checks 131
8.3 Running Executables Built in the Project 133
8.4 Cross Compiling Hello World 136
8.5 Cross Compiling for a Microcontroller 140
8.6 Cross Compiling an Existing Project 143
8.7 Cross Compiling a Complex Project - VTK 145
8.8 Some Tips and Tricks 147

CHAPTER 9 PACKAGING WITH CPACK 149

9.1 CPack Basics 149
Simple Example 150
What Happens When CPack.cmake Is Included? 151
Adding Custom CPack Options 152
Options Added by CPack 153
9.2 CPack Source Packages 154
9.3 CPack Installer Commands 154
9.4 CPack for Windows Installer NSIS 156
CPack Variables Used by CMake for NSIS 156
Creating Windows Short Cuts in the Start Menu 161
Advanced NSIS CPack Options 161
Setting File Extension Associations With NSIS 162
Installing Microsoft Run Time Libraries 163
CPack Component Install Support 163
9.5 CPack for Cygwin Setup 173
9.6 CPack for Mac OS X PackageMaker 176
9.7 CPack for Mac OS X Drag and Drop 178
9.8 CPack for Mac OS X X11 Applications 180
9.9 CPack for Debian Packages 182
9.10 CPack for RPM 183
9.11 CPack Files 183

CHAPTER 10 AUTOMATION & TESTING WITH CMAKE 185

10.1 Testing with CMake, CTest, and CDash 185
10.2 How Does CMake Facilitate Testing? 186

10.3 Additional Test Properties 187
10.4 Testing Using CTest 189
10.5 Using CTest to Drive Complex Tests 191
10.6 Handling a Large Number of Tests 192
10.7 Producing Test Dashboards 194
Adding CDash Dashboard Support to a Project *196*
Client Setup *199*
10.8 Customizing Dashboards for a Project 202
Dashboard Submissions Settings *202*
Filtering Errors and Warnings *203*
Adding Notes to a Dashboard *205*
10.9 Setting up Automated Dashboard Clients 206
Settings for Continuous Dashboards *210*
Variables Available in CTest Scripts *212*
10.10 Advanced CTest Scripting 212
Limitations of Traditional CTest Scripting *213*
Extended CTest Scripting *213*
10.11 Setting up a Dashboard Server 218
CDash Server *218*
Advanced Server Management *220*
Build Groups *223*
Email *225*
Sites *226*
Graphs *227*
Adding Notes to a Build *228*
Logging *229*
Test Timing *229*
Mobile Support *230*
Backing up CDash *230*
Upgrading CDash *231*
CDash Maintenance *232*
10.12 Subprojects 233
Using ctest_submit with PARTS and FILES *236*
Splitting Your Project into Multiple Subprojects *237*

CHAPTER 11 PORTING CMAKE TO NEW PLATFORMS AND LANGUAGES241

11.1 The Determine System Process 241
11.2 The Enable Language Process 242
11.3 Porting to a New Platform 244
11.4 Adding a New Language 246
11.5 Rule Variable Listing 247
General Tag Variables *247*
Language Specific Information *248*

11.6 Compiler and Platform Examples 248
Como Compiler 248
Borland Compiler 249
11.7 Extending CMake 250
Creating a Loaded Command 250
Using a Loaded Command 251

CHAPTER 12 TUTORIALS 255

12.1 A Basic Starting Point (Step 1) 255
Adding a Version Number and Configured Header File 256
12.2 Adding a Library (Step 2) 258
12.3 Installing and Testing (Step 3) 260
12.4 Adding System Introspection (Step 4) 262
12.5 Adding a Generated File and Generator (Step 5) 263
12.6 Building an Installer (Step 6) 267
12.7 Adding Support for a Dashboard (Step 7) 268

APPENDIX A - VARIABLES 269

Variables That Change Behavior 269
Variables That Describe the System 272
Variables for Languages 274
Variables That Control the Build 278
Variables That Provide Information 280

APPENDIX B – COMMAND LINE REFERENCE 287

CMake Command Line Options 287
CMake Generators 292
CTest Command Line Options 294
CPack Command Line Options 298
CPack Generators 299

APPENDIX C – LISTFILE COMMANDS 301

Current Commands 301
Compatibility Commands 366

APPENDIX D – SELECTED MODULES 373

CMake Modules 373

APPENDIX E - PROPERTIES 411

Properties of Global Scope 411
Properties on Directories 414
Properties on Targets 417
Properties on Tests 431
Properties on Source Files 431
Properties on Cache Entries 434

APPENDIX F – CMAKE POLICIES 437

INDEX 447

Index

A

ALL_BUILD target, 13, 109
ANT, 4
assert, 41
autoconf, 3
  converting to CMake, 121

B

batch commands, 103
binary packages, 150
Borland compiler, 5, 21, 23, 249
build
  configurations, 32, 67
  in-source. *See* in-source build
  out-of-source. *See* out-of-source build
build platform, 125
building your project, 19

C

cache
  advanced entries, 18, 30
  documentation, 30
  entries, 27, 29, 434
  find_* commands, 86
  FORCE option, 18, 32
  intializing, 17
  predefined options, 30
  purpose, 30
  set command, 29
  variable behavior, 31
case sensitivity, 9, 38
ccmake. *See* running CMake
CDash, 195
  adding notes, 205
  automatic submissions, 206
  backup, 230
  client setup, 199
  creating a new project, 196
  email, 225

expected builds, 224
filtering errors/warnings, 203
logging, 229
mobile support, 230
server setup, 218
sites, 226
specifying the server, 202
submitting results, 196
subprojects, 233
timing, 229
tutorial, 268
upgrading, 231

CMake
benefits, 1
command line, 287
compiler selection, 15
extending, 250
history, 3
plugin, 250
porting, 241
Structure, 21
syntax, 8, 33
tutorial, 255
versions, 43

CMake Commands, 8, 9, 34, 301
add_custom_command, 62, 105–8, 109, 110, 111, 112, 123, 301
add_custom_target, 24, 108, 109, 110, 111, 123, 303
add_definitions, 88, 95, 96, 304
add_dependencies, 81, 304
add_executable, 9, 10, 24, 25, 34, 121, 193, 305
add_library, 24, 25, 34, 121, 306
add_subdirectory, 28, 34, 115, 116, 122, 307
add_test, 38, 39, 186, 187, 188, 192, 193, 307
aux_source_directory, 308
break, 42, 309
build_command, 309
cmake_minimum_required, 44, 51, 214, 309
cmake_policy, 51, 310
configure_file, 97, 98, 122, 311
create_test_sourcelist, 192, 193, 311
define_property, 312
else, 10, 35
elseif, 36
enable_language, 313
enable_testing, 313
endif, 10, 35
exec_program, 120, 217
execute_process, 313
export, 81, 314
file, 89, 116, 119, 120, 315
find_file, 29, 85, 318
find_library, 10, 41, 46, 47, 85, 86, 320
find_package, 48, 85, 92, 95, 323
find_path, 46, 47, 85, 86, 329
find_program, 85, 105, 332
fltk_wrap_ui, 50, 335
foreach, 29, 35, 38, 39, 335
function, 38, 39, 336
get_cmake_property, 336
get_directory_property, 337
get_filename_component, 193, 337
get_property, 25, 26, 337
get_source_file_property, 25, 338
get_target_property, 24, 105, 106, 108, 339
get_test_property, 339
if, 10, 27, 35, 36, 37, 38, 40, 42, 339
include, 28, 45, 46, 95, 343
include_directories, 86, 99, 252, 344
include_external_msproject, 344
include_regular_expression, 82, 344
install, 66, 68, 345
link_directories, 350
link_libraries, 25, 62
list, 351
load_cache, 352
load_command, 252, 352
macro, 35, 38, 39, 40, 41, 217, 352
make_directory, 120
mark_as_advanced, 18, 30, 119, 353
math, 353
message, 29, 122, 353
obsolete/deprecated, 366
option, 29, 95, 121, 354
output_required_files, 82, 354
project, 9, 10, 34, 82, 105, 119, 123, 242, 354
qt_wrap_cpp, 355
qt_wrap_ui, 355
remove, 34, 193

remove_definitions, 355
return, 42, 355
separate_arguments, 34, 356
set, 9, 10, 18, 24, 27, 28, 29, 31, 32, 34, 35, 46, 356
set_directory_properties, 357
set_property, 25, 26, 357
set_source_file_properties, 358
set_source_files_properties, 25, 48, 62
set_target_properties, 24, 65, 359
set_tests_properties, 187, 188, 359
site_name, 359
source_group, 123, 359
string, 42, 62, 360
subdirs, 189
target_link_libraries, 25, 59, 60, 362
try_compile, 85, 88, 89, 92, 122, 131, 363
try_run, 85, 88, 89, 92, 364
unset, 365
variable_watch, 365
while, 35, 38, 39, 365
CMake Modules. *See* modules
CMake Policies. *See* policies
CMake Properties, 26, 411
ABSTRACT, 26
ADDITIONAL_MAKE_CLEAN_FILES, 26
COMPILE_DEFINITIONS, 96
COMPILE_DEFINITIONS_<CONFIG>, 97
COMPILE_FLAGS, 25
EXCLUDE_FROM_ALL, 26
GENERATED, 25
LABELS, 240
LINK_FLAGS, 25
list of, 27
LISTFILE_STACK, 26
OBJECT_DEPENDS, 26
WRAP_EXCLUDE, 26
CMake Variables
BUILD_SHARED_LIBS, 24
CMAKE_BUILD_TYPE, 32
CMAKE_BUILD_TYPE, 67
CMAKE_C_COMPILER, 16, 249
CMAKE_C_COMPILER_WORKS, 89
CMAKE_C_CREATE_SHARED_LIBRARY, 249
CMAKE_C_FLAGS, 16, 252
CMAKE_C_LINK_EXECUTABLE, 82
CMAKE_CFG_INTDIR, 32, 62, 104
CMAKE_COMMAND, 104, 106, 107, 206
CMAKE_CONFIGURATION_TYPES, 32
CMAKE_CROSSCOMPILING, 127
CMAKE_CTEST_COMMAND, 192
CMAKE_CURRENT_BINARY_DIR, 105, 267
CMAKE_CURRENT_SOURCE_DIR, 105
CMAKE_CXX_COMPILE_OBJECT, 250
CMAKE_CXX_COMPILER, 16, 244, 245, 248, 249, 250
CMAKE_CXX_CREATE_SHARED_LIBRARY, 244, 248, 249
CMAKE_CXX_CREATE_SHARED_MODULE, 249
CMAKE_CXX_CREATE_STATIC_LIBRARY, 245, 249, 250
CMAKE_CXX_FLAGS, 16, 32, 251
CMAKE_CXX_FLAGS_DEBUG, 245
CMAKE_CXX_FLAGS_DEBUG_INIT, 245
CMAKE_CXX_FLAGS_INIT, 245
CMAKE_CXX_FLAGS_MINSIZEREL, 245
CMAKE_CXX_FLAGS_MINSIZEREL_INIT, 245
CMAKE_CXX_FLAGS_RELEASE, 245
CMAKE_CXX_FLAGS_RELEASE_INIT, 245
CMAKE_CXX_FLAGS_RELWITHDEBINFO, 246
CMAKE_CXX_FLAGS_RELWITHDEBINFO_INIT, 246
CMAKE_CXX_LINK_EXECUTABLE, 82, 249
CMAKE_DL_LIBS, 245
CMAKE_END_TEMP_FILE, 249, 250
CMAKE_FIND_ROOT_PATH, 129
CMAKE_GENERATOR, 192, 207
CMAKE_HOST_*, 130
CMAKE_INSTALL_COMPONENT, 75
CMAKE_INSTALL_PREFIX, 15, 67, 75
CMAKE_INSTALL_RPATH, 148
CMAKE_LIBRARY_PREFIX, 105

CMAKE_MAJOR_VERSION, 44
CMAKE_MAKE_PROGRAM, 192
CMAKE_MODULE_PATH, 45
CMAKE_RANLIB, 248, 249
CMAKE_ROOT, 95, 252, 265
CMAKE_SHARED_LIBRARY_C_FLAGS, 249
CMAKE_SHARED_LIBRARY_CREATE_C_FLAGS, 245
CMAKE_SHARED_LIBRARY_CREATE_CXX_FLAGS, 244, 245, 248
CMAKE_SHARED_LIBRARY_RUNTIME_C_FLAG, 245
CMAKE_SHARED_LIBRARY_RUNTIME_C_FLAG_SEP, 245
CMAKE_SHARED_MODULE_PREFIX, 105
CMAKE_SKIP_RPATH, 64
CMAKE_START_TEMP_FILE, 249, 250
CMAKE_SYSTEM_NAME, 127, 241, 242, 245
CMAKE_SYSTEM_PROCESSOR, 127, 142
CMAKE_SYSTEM_VERSION, 127, 241
CMAKE_TOOLCHAIN_FILE, 126
CMAKE_VERSION, 44
EXECUTABLE_OUTPUT_PATH, 105, 118, 119
LIBRARY_EXPORT, 61
LIBRARY_OUTPUT_PATH, 105, 118, 119
PROJECT_BINARY_DIR, 105, 119, 252
PROJECT_SOURCE_DIR, 37, 105, 109, 252, 253, 258
WIN32, 10, 24, 61, 123
cmake-gui, 11, 13, 15, 30
pulldown option, 30
CMakeSetup. *See* cmake-gui
cmMakefile, 23, 24, 31
code coverage, 196, 200
comments, 10, 33
comparison, 37
compile flags, 25
compiler specification, 15
compilers, 5, 243
compiling CMake, 8
conditional statements, 35
configured header, 256, 260
converting autoconf projects to CMake, 121
converting Makefiles to CMake, 120
converting Visual Studio projects, 123
copying files. *See* file
CPack, 149
command line, 298
components, 163
DESTDIR, 155
escape characters, 152
generators, 153
tutorial, 267
variables, 151
cross compiling, 125
hello world, 136
system introspection, 130
CTest, 189
advanced, 212
building tests, 191
command line, 294
options, 190
properties, 187
regular expressions, 188
running the tests, 189
tutorial, 260
variables, 212
CTest Commands
ctest_build, 214
ctest_build_command, 213
ctest_configure, 214
ctest_empty_binary_directory, 213
ctest_start, 214
ctest_submit, 214
ctest_test, 214
ctest_update, 214
custom commands, 103, 105
custom targets, 108
CVS, 209
cygwin, 149, 173

D

DART. *See* CDash
dashboards, 17, 185, 194, 195
Debian, 150, 182

declspec, 61
dependencies, 82
dependency analysis, 16, 26
DESTDIR, 75, 149, 155
directory installation, 72
directory structures, 10, 115
 creating, 120
 properties, 26

E

editing CMakeLists files, 17
Emacs modes, 17
embedded devices, 125
environment variables, 2, 9, 16, 64, 217
escape character, 33
executing programs, 120

F

false, 38
file
 COMPONENT, 67
 copying, 104
 exclude, 74
 existance, 36
 globbing, 119
 installing, 66
 OPTIONAL, 68
 permissions, 67
 properties, 25
 regular expressions, 74
 remove, 104
 rename, 71
filtering errors and warnings, 203
final pass, 24
find modules. *See* modules
flags, 16, 25, 64
flow control, 34
FLTK, 50, 143
fortran, 242, 278
function, 35
functions, 39

G

generator, 15, 21, 23, 32, 207, 263
graphviz, 289

H

header files, configured, 97
hello world, 9, 136
history of CMake, 3

I

initial pass, 23
in-source build, 10, 117
installation scripts, 75
installer, 267
installing CMake
 Mac OS X, 7
 UNIX, 7
 Windows, 7
installing files. *See* files

J

JAM, 4

K

KDE, 3

L

languages, 34, 242, 247
LaTeX custom target, 109

libraries, 59
  API changes, 64
  exporting, 80
  importing, 78
  installing prerequisites, 76
  modules, 24
  pitfalls, 63
  rpath, 64
  shared, 24
  shared vs static, 59
  soname, 64
  static, 24, 25
  versions, 64
library configurations, 59
linking
  libraries, 56
  system libraries, 57
  Windows, 59
list. *See* variables
loaded commands, 26, 250, 254
looping constructs, 35

M

Mac OS X
  drag and drop installer, 178
  frameworks, 319, 322, 327
  Package Maker, 149, 176
  relocatable applications, 76
  Xcode. *See* Xcode
macros, 39, 40
Makefiles, 21, 23, 32
  build configurations, 32
microcontroller, 140
MinGW, 13, 143
modules, 44, 373
  documentation, 47
  find, 45, 93
  find <Package>Config.cmake, 93
  find conventions, 93
  path, 45
  system introspection, 47, 87
  utility, 47
modules, loadable. *See* libraries

N

NMake, 23
not found, 86
NSIS, 149, 153, 156
Nullsoft Installer, 149
numeric comparison, 37

O

operator precedence, 38
operators, 36
out-of-source build, 10, 117

P

package configuration file, 99
platforms, 4, 242
PLEASE_FILL_OUT, 131
plugins, 59, 250, 251
policies, 50, 437
  scope, 52
  setting, 51
  support multiple versions, 56
portability issues, 103
portability of code, 87
porting CMake, 241
POSIX systems, 241
project
  languages, 34
  name, 34
properties. *See* CMake Properties
proxies, 203
purify, 82, 200, 203

Q

qmake, 4

Qt, 2, 49
Qt CMake user interface. *See* cmake-gui
quoting, 9

R

regression testing, 185
regular expressions, 37, 42–43, 82, 190
relocatable packages, 76, 150
removing files. *See* file
required package, 101
RPATH, 64, 148
RPM, 150, 183
rule variables, 247
running CMake, 10
    ccmake, 13
    cmake-gui, 11
    command line, 15

S

SCons, 4
shared library. *See* libraries
shell commands, 103
source files, 25, 431
    generated, 107
source packages, 150, 154
subprojects, 237
SWIG, 2, 47–49
syntax of CMake, 8, 33
system introspection, 262, *See* modules

T

target platform, 126
targets, 24
    custom, 108
    custom commands, 105
    exporting, 79
    importing, 78
    properties, 25, 417
testing, 185
toolchain file, 126
true, 38
tutorial, 255

U

unit testing, 186
UNIX
    libraries, 63
    Makefiles. *See* Makefiles
    symbols, 61
utility modules. *See* modules

V

valgrind, 200, 203
variable argument lists, 41
variable scope, 28
variables, 9, 27, 269
    cache, 29
    interaction with cache, 31
    list, 9, 29, 34
    PARENT_SCOPE, 28
    quoting, 9
    remove from list, 34
    rule, 247
    scope, 40
    types, 30
versions, 43
Vim mode, 17
Visual Studio, 3, 5, 13, 15, 17, 21, 23, 32, 34, 66, 104, 105, 107, 109, 112, 117
    converting to CMake, 123
    file groups, 123
    header files, 123

W

white space, 33
Windows

build directory, 119
CDash, 208
dllexport, 61
executable, 24
executables, 123
installers. *See* NSIS
linking, 59
manifest files, 163
registry entries, 2, 86
registry values, 9
relocatable applications, 76
run time libraries, 163
symbols, 61
WinMain, 24, 123
wrapping C/C++. *See* SWIG

X

Xcode, 3, 66, 274