cmake\_minimum\_required(VERSION 3.0)

project(CatchSelfTest)

set\_property(GLOBAL PROPERTY USE\_FOLDERS ON)

# define some folders

set(CATCH\_DIR ${CMAKE\_CURRENT\_SOURCE\_DIR})

set(SELF\_TEST\_DIR ${CATCH\_DIR}/projects/SelfTest)

set(BENCHMARK\_DIR ${CATCH\_DIR}/projects/Benchmark)

set(HEADER\_DIR ${CATCH\_DIR}/include)

if(USE\_CPP11)

## We can't turn this on by default, since it breaks on travis

message(STATUS "Enabling C++11")

set(CMAKE\_CXX\_FLAGS "-std=c++11 ${CMAKE\_CXX\_FLAGS}")

elseif(USE\_CPP14)

message(STATUS "Enabling C++14")

set(CMAKE\_CXX\_FLAGS "-std=c++14 ${CMAKE\_CXX\_FLAGS}")

endif()

if(USE\_WMAIN)

set(CMAKE\_EXE\_LINKER\_FLAGS "${CMAKE\_EXE\_LINKER\_FLAGS} /ENTRY:wmainCRTStartup")

endif()

#checks that the given hard-coded list contains all headers + sources in the given folder

function(CheckFileList LIST\_VAR FOLDER)

set(MESSAGE " should be added to the variable ${LIST\_VAR}")

set(MESSAGE "${MESSAGE} in ${CMAKE\_CURRENT\_LIST\_FILE}\n")

file(GLOB GLOBBED\_LIST "${FOLDER}/\*.cpp"

"${FOLDER}/\*.hpp"

"${FOLDER}/\*.h")

list(REMOVE\_ITEM GLOBBED\_LIST ${${LIST\_VAR}})

foreach(EXTRA\_ITEM ${GLOBBED\_LIST})

string(REPLACE "${CATCH\_DIR}/" "" RELATIVE\_FILE\_NAME "${EXTRA\_ITEM}")

message(AUTHOR\_WARNING "The file \"${RELATIVE\_FILE\_NAME}\"${MESSAGE}")

endforeach()

endfunction()

function(CheckFileListRec LIST\_VAR FOLDER)

set(MESSAGE " should be added to the variable ${LIST\_VAR}")

set(MESSAGE "${MESSAGE} in ${CMAKE\_CURRENT\_LIST\_FILE}\n")

file(GLOB\_RECURSE GLOBBED\_LIST "${FOLDER}/\*.cpp"

"${FOLDER}/\*.hpp"

"${FOLDER}/\*.h")

list(REMOVE\_ITEM GLOBBED\_LIST ${${LIST\_VAR}})

foreach(EXTRA\_ITEM ${GLOBBED\_LIST})

string(REPLACE "${CATCH\_DIR}/" "" RELATIVE\_FILE\_NAME "${EXTRA\_ITEM}")

message(AUTHOR\_WARNING "The file \"${RELATIVE\_FILE\_NAME}\"${MESSAGE}")

endforeach()

endfunction()

# define the sources of the self test

# Please keep these ordered alphabetically

set(TEST\_SOURCES

${SELF\_TEST\_DIR}/ApproxTests.cpp

${SELF\_TEST\_DIR}/BDDTests.cpp

${SELF\_TEST\_DIR}/ClassTests.cpp

${SELF\_TEST\_DIR}/CmdLineTests.cpp

${SELF\_TEST\_DIR}/CompilationTests.cpp

${SELF\_TEST\_DIR}/ConditionTests.cpp

${SELF\_TEST\_DIR}/DecompositionTests.cpp

${SELF\_TEST\_DIR}/EnumToString.cpp

${SELF\_TEST\_DIR}/ExceptionTests.cpp

${SELF\_TEST\_DIR}/GeneratorTests.cpp

${SELF\_TEST\_DIR}/MessageTests.cpp

${SELF\_TEST\_DIR}/MiscTests.cpp

${SELF\_TEST\_DIR}/PartTrackerTests.cpp

${SELF\_TEST\_DIR}/TagAliasTests.cpp

${SELF\_TEST\_DIR}/TestMain.cpp

${SELF\_TEST\_DIR}/ToStringGeneralTests.cpp

${SELF\_TEST\_DIR}/ToStringPair.cpp

${SELF\_TEST\_DIR}/ToStringTuple.cpp

${SELF\_TEST\_DIR}/ToStringVector.cpp

${SELF\_TEST\_DIR}/ToStringWhich.cpp

${SELF\_TEST\_DIR}/TrickyTests.cpp

${SELF\_TEST\_DIR}/VariadicMacrosTests.cpp

${SELF\_TEST\_DIR}/MatchersTests.cpp

)

CheckFileList(TEST\_SOURCES ${SELF\_TEST\_DIR})

# A set of impl files that just #include a single header

# Please keep these ordered alphabetically

set(IMPL\_SOURCES

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_common.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_console\_colour.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_debugger.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_capture.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_config.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_exception.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_generators.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_registry\_hub.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_reporter.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_runner.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_interfaces\_testcase.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_message.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_option.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_ptr.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_stream.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_streambuf.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_test\_spec.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_xmlwriter.cpp

${SELF\_TEST\_DIR}/SurrogateCpps/catch\_test\_case\_tracker.cpp

)

CheckFileList(IMPL\_SOURCES ${SELF\_TEST\_DIR}/SurrogateCpps)

# Please keep these ordered alphabetically

set(TOP\_LEVEL\_HEADERS

${HEADER\_DIR}/catch.hpp

${HEADER\_DIR}/catch\_session.hpp

${HEADER\_DIR}/catch\_with\_main.hpp

)

CheckFileList(TOP\_LEVEL\_HEADERS ${HEADER\_DIR})

# Please keep these ordered alphabetically

set(EXTERNAL\_HEADERS

${HEADER\_DIR}/external/clara.h

${HEADER\_DIR}/external/tbc\_text\_format.h

)

CheckFileList(EXTERNAL\_HEADERS ${HEADER\_DIR}/external)

# Please keep these ordered alphabetically

set(INTERNAL\_HEADERS

${HEADER\_DIR}/internal/catch\_approx.hpp

${HEADER\_DIR}/internal/catch\_assertionresult.h

${HEADER\_DIR}/internal/catch\_assertionresult.hpp

${HEADER\_DIR}/internal/catch\_capture.hpp

${HEADER\_DIR}/internal/catch\_clara.h

${HEADER\_DIR}/internal/catch\_commandline.hpp

${HEADER\_DIR}/internal/catch\_common.h

${HEADER\_DIR}/internal/catch\_common.hpp

${HEADER\_DIR}/internal/catch\_compiler\_capabilities.h

${HEADER\_DIR}/internal/catch\_config.hpp

${HEADER\_DIR}/internal/catch\_console\_colour.hpp

${HEADER\_DIR}/internal/catch\_console\_colour\_impl.hpp

${HEADER\_DIR}/internal/catch\_context.h

${HEADER\_DIR}/internal/catch\_context\_impl.hpp

${HEADER\_DIR}/internal/catch\_debugger.h

${HEADER\_DIR}/internal/catch\_debugger.hpp

${HEADER\_DIR}/internal/catch\_default\_main.hpp

${HEADER\_DIR}/internal/catch\_errno\_guard.hpp

${HEADER\_DIR}/internal/catch\_evaluate.hpp

${HEADER\_DIR}/internal/catch\_exception\_translator\_registry.hpp

${HEADER\_DIR}/internal/catch\_expression\_lhs.hpp

${HEADER\_DIR}/internal/catch\_fatal\_condition.hpp

${HEADER\_DIR}/internal/catch\_generators.hpp

${HEADER\_DIR}/internal/catch\_generators\_impl.hpp

${HEADER\_DIR}/internal/catch\_impl.hpp

${HEADER\_DIR}/internal/catch\_interfaces\_capture.h

${HEADER\_DIR}/internal/catch\_interfaces\_config.h

${HEADER\_DIR}/internal/catch\_interfaces\_exception.h

${HEADER\_DIR}/internal/catch\_interfaces\_generators.h

${HEADER\_DIR}/internal/catch\_interfaces\_registry\_hub.h

${HEADER\_DIR}/internal/catch\_interfaces\_reporter.h

${HEADER\_DIR}/internal/catch\_interfaces\_runner.h

${HEADER\_DIR}/internal/catch\_interfaces\_tag\_alias\_registry.h

${HEADER\_DIR}/internal/catch\_interfaces\_testcase.h

${HEADER\_DIR}/internal/catch\_legacy\_reporter\_adapter.h

${HEADER\_DIR}/internal/catch\_legacy\_reporter\_adapter.hpp

${HEADER\_DIR}/internal/catch\_list.hpp

${HEADER\_DIR}/internal/catch\_matchers.hpp

${HEADER\_DIR}/internal/catch\_matchers\_string.h

${HEADER\_DIR}/internal/catch\_matchers\_string.hpp

${HEADER\_DIR}/internal/catch\_matchers\_vector.h

${HEADER\_DIR}/internal/catch\_message.h

${HEADER\_DIR}/internal/catch\_message.hpp

${HEADER\_DIR}/internal/catch\_notimplemented\_exception.h

${HEADER\_DIR}/internal/catch\_notimplemented\_exception.hpp

${HEADER\_DIR}/internal/catch\_objc.hpp

${HEADER\_DIR}/internal/catch\_objc\_arc.hpp

${HEADER\_DIR}/internal/catch\_option.hpp

${HEADER\_DIR}/internal/catch\_platform.h

${HEADER\_DIR}/internal/catch\_ptr.hpp

${HEADER\_DIR}/internal/catch\_reenable\_warnings.h

${HEADER\_DIR}/internal/catch\_registry\_hub.hpp

${HEADER\_DIR}/internal/catch\_reporter\_registrars.hpp

${HEADER\_DIR}/internal/catch\_reporter\_registry.hpp

${HEADER\_DIR}/internal/catch\_result\_builder.h

${HEADER\_DIR}/internal/catch\_result\_builder.hpp

${HEADER\_DIR}/internal/catch\_result\_type.h

${HEADER\_DIR}/internal/catch\_run\_context.hpp

${HEADER\_DIR}/internal/catch\_section.h

${HEADER\_DIR}/internal/catch\_section.hpp

${HEADER\_DIR}/internal/catch\_section\_info.h

${HEADER\_DIR}/internal/catch\_section\_info.hpp

${HEADER\_DIR}/internal/catch\_stream.h

${HEADER\_DIR}/internal/catch\_stream.hpp

${HEADER\_DIR}/internal/catch\_streambuf.h

${HEADER\_DIR}/internal/catch\_suppress\_warnings.h

${HEADER\_DIR}/internal/catch\_tag\_alias.h

${HEADER\_DIR}/internal/catch\_tag\_alias\_registry.h

${HEADER\_DIR}/internal/catch\_tag\_alias\_registry.hpp

${HEADER\_DIR}/internal/catch\_test\_case\_info.h

${HEADER\_DIR}/internal/catch\_test\_case\_info.hpp

${HEADER\_DIR}/internal/catch\_test\_case\_registry\_impl.hpp

${HEADER\_DIR}/internal/catch\_test\_case\_tracker.hpp

${HEADER\_DIR}/internal/catch\_test\_registry.hpp

${HEADER\_DIR}/internal/catch\_test\_spec.hpp

${HEADER\_DIR}/internal/catch\_test\_spec\_parser.hpp

${HEADER\_DIR}/internal/catch\_text.h

${HEADER\_DIR}/internal/catch\_timer.h

${HEADER\_DIR}/internal/catch\_timer.hpp

${HEADER\_DIR}/internal/catch\_tostring.h

${HEADER\_DIR}/internal/catch\_tostring.hpp

${HEADER\_DIR}/internal/catch\_totals.hpp

${HEADER\_DIR}/internal/catch\_type\_traits.hpp

${HEADER\_DIR}/internal/catch\_version.h

${HEADER\_DIR}/internal/catch\_version.hpp

${HEADER\_DIR}/internal/catch\_wildcard\_pattern.hpp

${HEADER\_DIR}/internal/catch\_windows\_h\_proxy.h

${HEADER\_DIR}/internal/catch\_xmlwriter.hpp

)

CheckFileList(INTERNAL\_HEADERS ${HEADER\_DIR}/internal)

# Please keep these ordered alphabetically

set(REPORTER\_HEADERS

${HEADER\_DIR}/reporters/catch\_reporter\_automake.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_bases.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_compact.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_console.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_junit.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_multi.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_tap.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_teamcity.hpp

${HEADER\_DIR}/reporters/catch\_reporter\_xml.hpp

)

CheckFileList(REPORTER\_HEADERS ${HEADER\_DIR}/reporters)

# Specify the headers, too, so CLion recognises them as project files

set(HEADERS

${TOP\_LEVEL\_HEADERS}

${EXTERNAL\_HEADERS}

${INTERNAL\_HEADERS}

${REPORTER\_HEADERS}

)

set(BENCH\_SOURCES

${BENCHMARK\_DIR}/BenchMain.cpp

${BENCHMARK\_DIR}/StringificationBench.cpp

)

CheckFileList(BENCH\_SOURCES ${BENCHMARK\_DIR})

# Provide some groupings for IDEs

SOURCE\_GROUP("Tests" FILES ${TEST\_SOURCES})

SOURCE\_GROUP("Surrogates" FILES ${IMPL\_SOURCES})

SOURCE\_GROUP("Benchmarks" FILES ${BENCH\_SOURCES})

# configure the executable

include\_directories(${HEADER\_DIR})

# Projects consuming Catch via ExternalProject\_Add might want to use install step

# without building all of our selftests.

if (NOT NO\_SELFTEST)

add\_executable(SelfTest ${TEST\_SOURCES} ${IMPL\_SOURCES} ${HEADERS})

add\_executable(Benchmark ${BENCH\_SOURCES} ${HEADERS})

# Add desired warnings

if ( CMAKE\_CXX\_COMPILER\_ID MATCHES "Clang|AppleClang|GNU" )

target\_compile\_options( SelfTest PRIVATE -Wall -Wextra )

target\_compile\_options( Benchmark PRIVATE -Wall -Wextra )

endif()

if ( CMAKE\_CXX\_COMPILER\_ID MATCHES "MSVC" )

target\_compile\_options( SelfTest PRIVATE /W4 /w44265 /WX )

target\_compile\_options( Benchmark PRIVATE /W4 )

endif()

# configure unit tests via CTest

enable\_testing()

add\_test(NAME RunTests COMMAND SelfTest)

add\_test(NAME ListTests COMMAND SelfTest --list-tests)

set\_tests\_properties(ListTests PROPERTIES PASS\_REGULAR\_EXPRESSION "[0-9]+ test cases")

add\_test(NAME ListTags COMMAND SelfTest --list-tags)

set\_tests\_properties(ListTags PROPERTIES PASS\_REGULAR\_EXPRESSION "[0-9]+ tags")

endif() # !NO\_SELFTEST

install(DIRECTORY "single\_include/" DESTINATION "include/catch")