cmake\_minimum\_required (VERSION 2.8)

project (opendnp3)

set(OPENDNP3\_MAJOR\_VERSION 2)

set(OPENDNP3\_MINOR\_VERSION 2)

set(OPENDNP3\_MICRO\_VERSION 1)

set(OPENDNP3\_VERSION ${OPENDNP3\_MAJOR\_VERSION}.${OPENDNP3\_MINOR\_VERSION}.${OPENDNP3\_MICRO\_VERSION})

if(NOT CMAKE\_BUILD\_TYPE)

set(CMAKE\_BUILD\_TYPE "Release" CACHE STRING

"Choose the type of build, options are: Debug Release

RelWithDebInfo MinSizeRel."

FORCE)

endif()

message("CMake build is: ${CMAKE\_BUILD\_TYPE}")

include(${PROJECT\_SOURCE\_DIR}/cmake/settings.cmake)

# various optional libraries and projects

option(DNP3\_ALL "Build all optional projects (secauth, demos, tests)" OFF)

option(DNP3\_TEST "Build tests" OFF)

option(DNP3\_DEMO "Build demo applications" OFF)

option(DNP3\_DECODER "Build the decoder library" OFF)

option(DNP3\_TLS "Build TLS client/server support" OFF)

option(DNP3\_JAVA "Build the Java bindings" OFF)

# other options off-by-default that you can enable

option(WERROR "Set all warnings to errors" OFF)

option(STATICLIBS "Builds static versions of all installed libraries" OFF)

option(COVERAGE "Builds the libraries with coverage info for gcov (gcc only)" OFF)

option(PROFILE "Builds the libraries with profiling support (gcc only)" OFF)

if(DNP3\_ALL)

message("enabling all optional components")

set(DNP3\_DEMO ON)

set(DNP3\_TEST ON)

set(DNP3\_TLS ON)

set(DNP3\_DECODER ON)

set(DNP3\_JAVA ON)

endif()

if(DNP3\_JAVA)

find\_package(Java COMPONENTS Development)

find\_package(JNI REQUIRED)

include\_directories(${JNI\_INCLUDE\_DIRS})

endif()

SET(ASIO\_SUBMODULE\_DIR "${PROJECT\_SOURCE\_DIR}/deps/asio/asio/include")

# detection stuff for ASIO

if (EXISTS "${ASIO\_SUBMODULE\_DIR}/asio.hpp")

message("ASIO has been checked out as a git submodule: ${ASIO\_SUBMODULE\_DIR}")

include\_directories(${ASIO\_SUBMODULE\_DIR})

else()

message("ASIO has NOT been checked out as a git submodule...")

if (ASIO\_HOME)

message("ASIO\_HOME defined in cache: ${ASIO\_HOME}")

include\_directories(${ASIO\_HOME})

else()

if(DEFINED ENV{ASIO\_HOME})

message("ASIO\_HOME defined in environment: $ENV{ASIO\_HOME}")

include\_directories($ENV{ASIO\_HOME})

else()

message("ASIO\_HOME was not defined. ASIO expected to be on include path")

endif()

endif()

endif()

# required for ASIO in C++11 only mode

add\_definitions(-DASIO\_STANDALONE)

if(DNP3\_TLS)

add\_definitions(-DOPENDNP3\_USE\_TLS)

find\_package(OpenSSL REQUIRED)

message("OpenSSL libraries: ${OPENSSL\_LIBRARIES}")

if(WIN32)

include\_directories(${OPENSSL\_INCLUDE\_DIR})

endif()

endif()

set(CMAKE\_REQUIRED\_FLAGS ${CMAKE\_CXX\_FLAGS})

# include paths for all the local libraries

include\_directories(./cpp/libs/src)

include\_directories(./cpp/libs/include)

include\_directories(./cpp/tests/libs/src)

# ---- openpal library ----

file(GLOB\_RECURSE openpal\_SRC ./cpp/libs/src/openpal/\*.cpp ./cpp/libs/src/openpal/\*.h ./cpp/libs/include/openpal/\*.h)

add\_library(openpal ${LIB\_TYPE} ${openpal\_SRC})

install(TARGETS openpal DESTINATION lib)

set\_target\_properties(openpal PROPERTIES FOLDER libs)

# ---- opendnp3 library ----

file(GLOB\_RECURSE opendnp3\_SRC ./cpp/libs/src/opendnp3/\*.cpp ./cpp/libs/src/opendnp3/\*.h ./cpp/libs/include/opendnp3/\*.h)

add\_library(opendnp3 ${LIB\_TYPE} ${opendnp3\_SRC})

target\_link\_libraries(opendnp3 openpal)

install(TARGETS opendnp3 DESTINATION lib)

set\_target\_properties(opendnp3 PROPERTIES FOLDER libs)

if(DNP3\_DECODER)

file(GLOB\_RECURSE dnp3decode\_SRC ./cpp/libs/src/dnp3decode/\*.cpp ./cpp/libs/src/dnp3decode/\*.h ./cpp/libs/include/dnp3decode/\*.h)

add\_library(dnp3decode ${LIB\_TYPE} ${dnp3decode\_SRC})

target\_link\_libraries(dnp3decode opendnp3)

install(TARGETS dnp3decode DESTINATION lib)

set\_target\_properties(dnp3decode PROPERTIES FOLDER libs)

endif()

# ---- asiopal library ----

file(GLOB\_RECURSE asiopal\_HPP ./cpp/libs/src/asiopal/\*.h ./cpp/libs/include/asiopal/\*.h)

if(DNP3\_TLS)

file(GLOB\_RECURSE asiopal\_CPP ./cpp/libs/src/asiopal/\*.cpp)

else()

file(GLOB asiopal\_CPP ./cpp/libs/src/asiopal/\*.cpp)

endif()

add\_library(asiopal ${LIB\_TYPE} ${asiopal\_HPP} ${asiopal\_CPP})

set(asiopal\_link\_libraries "openpal")

if(DNP3\_TLS)

set(asiopal\_link\_libraries "${asiopal\_link\_libraries};${OPENSSL\_LIBRARIES}")

endif()

target\_link\_libraries(asiopal ${asiopal\_link\_libraries})

install(TARGETS asiopal DESTINATION lib)

set\_target\_properties(asiopal PROPERTIES FOLDER libs)

# ---- asiodnp3 library ----

file(GLOB\_RECURSE asiodnp3\_HPP ./cpp/libs/src/asiodnp3/\*.h ./cpp/libs/include/asiodnp3/\*.h)

if(DNP3\_TLS)

file(GLOB\_RECURSE asiodnp3\_CPP ./cpp/libs/src/asiodnp3/\*.cpp)

else()

file(GLOB asiodnp3\_CPP ./cpp/libs/src/asiodnp3/\*.cpp)

endif()

add\_library(asiodnp3 ${LIB\_TYPE} ${asiodnp3\_HPP} ${asiodnp3\_CPP})

target\_link\_libraries(asiodnp3 asiopal opendnp3)

install(TARGETS asiodnp3 DESTINATION lib)

set\_target\_properties(asiodnp3 PROPERTIES FOLDER libs)

if(DNP3\_JAVA)

file(GLOB\_RECURSE opendnp3java\_SRC ./java/cpp/\*.h ./java/cpp/\*.cpp)

add\_library(opendnp3java SHARED ${opendnp3java\_SRC})

target\_link\_libraries(opendnp3java asiodnp3)

install(TARGETS opendnp3java DESTINATION lib)

set\_target\_properties(opendnp3java PROPERTIES FOLDER libs)

endif()

# ----- install -----

# common pattern and exludes for all installed headers

set(INSTALL\_ARGS FILES\_MATCHING PATTERN "\*.h" PATTERN ".deps" EXCLUDE PATTERN ".libs" EXCLUDE)

install(DIRECTORY ./cpp/libs/include/ DESTINATION include ${INSTALL\_ARGS})

if(DNP3\_DEMO)

# ----- master demo executable -----

add\_executable(master-demo ./cpp/examples/master/main.cpp)

target\_link\_libraries (master-demo LINK\_PUBLIC asiodnp3 ${PTHREAD})

set\_target\_properties(master-demo PROPERTIES FOLDER demos)

# ----- master demo executable -----

add\_executable(master-gprs-demo ./cpp/examples/master-gprs/main.cpp)

target\_link\_libraries (master-gprs-demo LINK\_PUBLIC asiodnp3 ${PTHREAD})

set\_target\_properties(master-gprs-demo PROPERTIES FOLDER demos)

# ----- outstation demo executable -----

add\_executable(outstation-demo ./cpp/examples/outstation/main.cpp)

target\_link\_libraries (outstation-demo LINK\_PUBLIC asiodnp3 ${PTHREAD})

set\_target\_properties(outstation-demo PROPERTIES FOLDER demos)

if(DNP3\_DECODER)

# ----- decoder executable -----

add\_executable(decoder ./cpp/examples/decoder/main.cpp)

target\_link\_libraries (decoder asiodnp3 dnp3decode ${PTHREAD})

set\_target\_properties(decoder PROPERTIES FOLDER demos)

endif()

if(DNP3\_TLS)

# ----- master tls executable -----

add\_executable(master-tls-demo ./cpp/examples/tls/master/main.cpp)

target\_link\_libraries (master-tls-demo LINK\_PUBLIC asiodnp3 ${PTHREAD})

set\_target\_properties(master-tls-demo PROPERTIES FOLDER demos/tls)

# ----- outstation tls executable -----

add\_executable(outstation-tls-demo ./cpp/examples/tls/outstation/main.cpp)

target\_link\_libraries (outstation-tls-demo LINK\_PUBLIC asiodnp3 ${PTHREAD})

set\_target\_properties(outstation-tls-demo PROPERTIES FOLDER demos/tls)

# ----- master-gprs tls executable -----

add\_executable(master-gprs-tls-demo ./cpp/examples/tls/master-gprs/main.cpp)

target\_link\_libraries (master-gprs-tls-demo LINK\_PUBLIC asiodnp3 ${PTHREAD})

set\_target\_properties(master-gprs-tls-demo PROPERTIES FOLDER demos/tls)

endif()

endif()

if(DNP3\_TEST)

enable\_testing()

# ----- testlib library ------

file(GLOB\_RECURSE testlib\_SRC ./cpp/tests/libs/src/testlib/\*.cpp ./cpp/tests/libs/src/testlib/\*.h)

add\_library(testlib ${testlib\_SRC})

target\_link\_libraries(testlib openpal)

set\_target\_properties(testlib PROPERTIES FOLDER tests/mocks)

# ----- dnp3mocks library ------

file(GLOB\_RECURSE dnp3mocks\_SRC ./cpp/tests/libs/src/dnp3mocks/\*.cpp ./cpp/tests/libs/src/dnp3mocks/\*.h)

add\_library(dnp3mocks ${dnp3mocks\_SRC})

target\_link\_libraries(dnp3mocks opendnp3 testlib)

set\_target\_properties(dnp3mocks PROPERTIES FOLDER tests/mocks)

# ----- openpal tests -----

file(GLOB\_RECURSE openpal\_TESTSRC ./cpp/tests/openpal/src/\*.cpp ./cpp/tests/openpal/src/\*.h)

add\_executable (testopenpal ${openpal\_TESTSRC})

target\_link\_libraries (testopenpal LINK\_PUBLIC testlib ${PTHREAD})

set\_target\_properties(testopenpal PROPERTIES FOLDER tests)

add\_test(testopenpal testopenpal)

# ----- opendnp3 tests -----

file(GLOB\_RECURSE opendnp3\_TESTSRC ./cpp/tests/opendnp3/src/\*.cpp ./cpp/tests/opendnp3/src/\*.h)

add\_executable (testopendnp3 ${opendnp3\_TESTSRC})

target\_link\_libraries (testopendnp3 LINK\_PUBLIC dnp3mocks ${PTHREAD})

set\_target\_properties(testopendnp3 PROPERTIES FOLDER tests)

add\_test(testopendnp3 testopendnp3)

# ----- asiopal tests -----

if(DNP3\_TLS)

file(GLOB\_RECURSE asiopal\_TESTSRC ./cpp/tests/asiopal/src/\*.cpp ./cpp/tests/asiopal/src/\*.h)

else()

file(GLOB asiopal\_TESTSRC ./cpp/tests/asiopal/src/\*.cpp ./cpp/tests/asiopal/src/\*.h ./cpp/tests/asiopal/src/mocks/\*.cpp ./cpp/tests/asiopal/src/mocks/\*.h)

endif()

add\_executable (testasiopal ${asiopal\_TESTSRC})

target\_link\_libraries (testasiopal LINK\_PUBLIC asiopal testlib ${PTHREAD})

set\_target\_properties(testasiopal PROPERTIES FOLDER tests)

add\_test(testasiopal testasiopal)

# ----- asiodnp3 tests -----

file(GLOB\_RECURSE asiodnp3\_TESTSRC ./cpp/tests/asiodnp3/src/\*.cpp ./cpp/tests/asiodnp3/src/\*.h)

add\_executable (testasiodnp3 ${asiodnp3\_TESTSRC})

target\_link\_libraries (testasiodnp3 LINK\_PUBLIC asiodnp3 dnp3mocks ${PTHREAD})

set\_target\_properties(testasiodnp3 PROPERTIES FOLDER tests)

add\_test(testasiodnp3 testasiodnp3)

endif()

add\_custom\_target(

format

WORKING\_DIRECTORY ${CMAKE\_CURRENT\_SOURCE\_DIR}

COMMAND astyle -R ./cpp/\*.h ./cpp/\*.cpp --options=./config/astyle.cfg --exclude=./cpp/libs/include/opendnp3/gen --exclude=./cpp/libs/src/opendnp3/gen --exclude=./cpp/libs/src/opendnp3/objects

COMMAND astyle -R ./java/cpp/adapters/\*.h ./java/cpp/adapters/\*.cpp --options=./config/astyle.cfg

)