# This file is copyrighted under the BSD-license for buildsystem files of KDE

# copyright 2010, Patrick Spendrin <ps\_ml@gmx.de>

project(expat)

cmake\_minimum\_required(VERSION 2.6)

set(PACKAGE\_BUGREPORT "expat-bugs@libexpat.org")

set(PACKAGE\_NAME "expat")

set(PACKAGE\_VERSION "2.2.5")

set(PACKAGE\_STRING "${PACKAGE\_NAME} ${PACKAGE\_VERSION}")

set(PACKAGE\_TARNAME "${PACKAGE\_NAME}")

if(WINCE)

set(BUILD\_tools\_default OFF)

else()

set(BUILD\_tools\_default ON)

endif()

if(MSVC OR NOT BUILD\_tools\_default)

set(BUILD\_doc\_default OFF)

else()

find\_program(DOCBOOK\_TO\_MAN NAMES docbook2x-man docbook2man docbook-to-man)

if(DOCBOOK\_TO\_MAN)

set(BUILD\_doc\_default ON)

else()

set(BUILD\_doc\_default OFF)

endif()

endif()

option(BUILD\_tools "build the xmlwf tool for expat library" ${BUILD\_tools\_default})

option(BUILD\_examples "build the examples for expat library" ON)

option(BUILD\_tests "build the tests for expat library" ON)

option(BUILD\_shared "build a shared expat library" ON)

option(BUILD\_doc "build man page for xmlwf" ${BUILD\_doc\_default})

option(USE\_libbsd "utilize libbsd (for arc4random\_buf)" OFF)

option(INSTALL "install expat files in cmake install target" ON)

if(USE\_libbsd)

find\_library(LIB\_BSD NAMES bsd)

if(NOT LIB\_BSD)

message(FATAL\_ERROR "USE\_libbsd option is enabled, but libbsd was not found")

else()

set(HAVE\_LIBBSD TRUE)

endif()

endif()

# configuration options

set(XML\_CONTEXT\_BYTES 1024 CACHE STRING "Define to specify how much context to retain around the current parse point")

option(XML\_DTD "Define to make parameter entity parsing functionality available" ON)

option(XML\_NS "Define to make XML Namespaces functionality available" ON)

if(NOT WIN32)

option(XML\_DEV\_URANDOM "Define to include code reading entropy from `/dev/urandom'." ON)

if(XML\_DEV\_URANDOM)

add\_definitions(-DXML\_DEV\_URANDOM)

endif()

endif()

if(XML\_DTD)

set(XML\_DTD 1)

else(XML\_DTD)

set(XML\_DTD 0)

endif(XML\_DTD)

if(XML\_NS)

set(XML\_NS 1)

else(XML\_NS)

set(XML\_NS 0)

endif(XML\_NS)

if(BUILD\_tests)

enable\_testing()

endif(BUILD\_tests)

include(${CMAKE\_CURRENT\_LIST\_DIR}/ConfigureChecks.cmake)

set(EXTRA\_COMPILE\_FLAGS)

if(FLAG\_NO\_STRICT\_ALIASING)

set(EXTRA\_COMPILE\_FLAGS "${EXTRA\_COMPILE\_FLAGS} -fno-strict-aliasing")

endif()

set(CMAKE\_C\_FLAGS "${CMAKE\_C\_FLAGS} ${EXTRA\_COMPILE\_FLAGS}")

set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} ${EXTRA\_COMPILE\_FLAGS}")

include\_directories(${CMAKE\_CURRENT\_BINARY\_DIR} ${CMAKE\_CURRENT\_SOURCE\_DIR}/lib)

if(MSVC)

add\_definitions(-D\_CRT\_SECURE\_NO\_WARNINGS -wd4996)

endif(MSVC)

if(WIN32)

set(CMAKE\_DEBUG\_POSTFIX "d" CACHE STRING "Add a suffix, usually d on Windows")

endif(WIN32)

set(expat\_SRCS

lib/loadlibrary.c

lib/xmlparse.c

lib/xmlrole.c

lib/xmltok.c

lib/xmltok\_impl.c

lib/xmltok\_ns.c

)

if(BUILD\_shared)

set(\_SHARED SHARED)

if(WIN32)

set(expat\_SRCS ${expat\_SRCS} lib/libexpat.def)

endif(WIN32)

else(BUILD\_shared)

set(\_SHARED STATIC)

if(WIN32)

add\_definitions(-DXML\_STATIC)

endif(WIN32)

endif(BUILD\_shared)

add\_library(expat ${\_SHARED} ${expat\_SRCS})

if(USE\_libbsd)

target\_link\_libraries(expat ${LIB\_BSD})

endif()

set(LIBCURRENT 7) # sync

set(LIBREVISION 7) # with

set(LIBAGE 6) # configure.ac!

math(EXPR LIBCURRENT\_MINUS\_AGE "${LIBCURRENT} - ${LIBAGE}")

if(NOT WIN32)

set\_property(TARGET expat PROPERTY VERSION ${LIBCURRENT\_MINUS\_AGE}.${LIBAGE}.${LIBREVISION})

set\_property(TARGET expat PROPERTY SOVERSION ${LIBCURRENT\_MINUS\_AGE})

set\_property(TARGET expat PROPERTY NO\_SONAME ${NO\_SONAME})

endif(NOT WIN32)

macro(expat\_install)

if(INSTALL)

install(${ARGN})

endif()

endmacro()

expat\_install(TARGETS expat RUNTIME DESTINATION bin

LIBRARY DESTINATION lib

ARCHIVE DESTINATION lib)

set(prefix ${CMAKE\_INSTALL\_PREFIX})

set(exec\_prefix "\${prefix}/bin")

set(libdir "\${prefix}/lib")

set(includedir "\${prefix}/include")

configure\_file(expat.pc.in ${CMAKE\_CURRENT\_BINARY\_DIR}/expat.pc)

expat\_install(FILES lib/expat.h lib/expat\_external.h DESTINATION include)

expat\_install(FILES ${CMAKE\_CURRENT\_BINARY\_DIR}/expat.pc DESTINATION lib/pkgconfig)

if(BUILD\_tools)

set(xmlwf\_SRCS

xmlwf/xmlwf.c

xmlwf/xmlfile.c

xmlwf/codepage.c

xmlwf/readfilemap.c

)

add\_executable(xmlwf ${xmlwf\_SRCS})

set\_property(TARGET xmlwf PROPERTY RUNTIME\_OUTPUT\_DIRECTORY xmlwf)

target\_link\_libraries(xmlwf expat)

expat\_install(TARGETS xmlwf DESTINATION bin)

if(BUILD\_doc)

add\_custom\_command(TARGET expat PRE\_BUILD COMMAND "${DOCBOOK\_TO\_MAN}" "${PROJECT\_SOURCE\_DIR}/doc/xmlwf.xml" && mv "XMLWF.1" "${PROJECT\_SOURCE\_DIR}/doc/xmlwf.1")

expat\_install(FILES "${PROJECT\_SOURCE\_DIR}/doc/xmlwf.1" DESTINATION share/man/man1)

endif()

endif()

if(BUILD\_examples)

add\_executable(elements examples/elements.c)

set\_property(TARGET elements PROPERTY RUNTIME\_OUTPUT\_DIRECTORY examples)

target\_link\_libraries(elements expat)

add\_executable(outline examples/outline.c)

set\_property(TARGET outline PROPERTY RUNTIME\_OUTPUT\_DIRECTORY examples)

target\_link\_libraries(outline expat)

endif(BUILD\_examples)

if(BUILD\_tests)

## these are unittests that can be run on any platform

add\_executable(runtests tests/runtests.c tests/chardata.c tests/structdata.c tests/minicheck.c tests/memcheck.c)

set\_property(TARGET runtests PROPERTY RUNTIME\_OUTPUT\_DIRECTORY tests)

target\_link\_libraries(runtests expat)

add\_test(runtests tests/runtests)

add\_executable(runtestspp tests/runtestspp.cpp tests/chardata.c tests/structdata.c tests/minicheck.c tests/memcheck.c)

set\_property(TARGET runtestspp PROPERTY RUNTIME\_OUTPUT\_DIRECTORY tests)

target\_link\_libraries(runtestspp expat)

add\_test(runtestspp tests/runtestspp)

endif(BUILD\_tests)