set(LibSampleApp\_SOURCES)

list(APPEND LibSampleApp\_SOURCES

CaptionPresenter.cpp

ConnectionObserver.cpp

ConsolePrinter.cpp

ConsoleReader.cpp

GuiRenderer.cpp

InteractionManager.cpp

KeywordObserver.cpp

LocaleAssetsManager.cpp

UIManager.cpp

UserInputManager.cpp

SampleApplication.cpp

SampleEqualizerModeController.cpp)

if (PORTAUDIO)

list(APPEND LibSampleApp\_SOURCES PortAudioMicrophoneWrapper.cpp)

endif()

IF (A4B)

list(APPEND LibSampleApp\_SOURCES RevokeAuthorizationObserver.cpp)

ENDIF()

if (PCC)

list(APPEND LibSampleApp\_SOURCES PhoneCaller.cpp)

endif()

if (MCC)

list(APPEND LibSampleApp\_SOURCES MeetingClient.cpp)

list(APPEND LibSampleApp\_SOURCES CalendarClient.cpp)

endif()

if (ENDPOINT\_CONTROLLERS\_POWER\_CONTROLLER OR ENDPOINT\_CONTROLLERS\_TOGGLE\_CONTROLLER OR

ENDPOINT\_CONTROLLERS\_RANGE\_CONTROLLER OR ENDPOINT\_CONTROLLERS\_MODE\_CONTROLLER)

add\_definitions(-DENABLE\_ENDPOINT\_CONTROLLERS\_MENU)

endif()

if (ENDPOINT\_CONTROLLERS\_POWER\_CONTROLLER)

list(APPEND LibSampleApp\_SOURCES PowerControllerHandler.cpp)

endif()

if (ENDPOINT\_CONTROLLERS\_TOGGLE\_CONTROLLER)

list(APPEND LibSampleApp\_SOURCES ToggleControllerHandler.cpp)

endif()

if (ENDPOINT\_CONTROLLERS\_RANGE\_CONTROLLER)

list(APPEND LibSampleApp\_SOURCES RangeControllerHandler.cpp)

endif()

if (ENDPOINT\_CONTROLLERS\_MODE\_CONTROLLER)

list(APPEND LibSampleApp\_SOURCES ModeControllerHandler.cpp)

endif()

IF (HAS\_EXTERNAL\_MEDIA\_PLAYER\_ADAPTERS)

file(GLOB\_RECURSE SRC\_FILE ${CMAKE\_CURRENT\_SOURCE\_DIR}/ExternalMediaAdapterRegistration/\*.cpp)

foreach(myfile ${SRC\_FILE})

list(APPEND LibSampleApp\_SOURCES ${myfile})

endforeach(myfile)

ENDIF()

add\_library(LibSampleApp SHARED ${LibSampleApp\_SOURCES})

target\_include\_directories(LibSampleApp PUBLIC

# This is relative to project(SampleApp).

"${SampleApp\_SOURCE\_DIR}/include"

"${AudioResources\_SOURCE\_DIR}/include"

"${RegistrationManager\_SOURCE\_DIR}/include"

"${ESP\_SOURCE\_DIR}/include"

"${InterruptModel\_SOURCE\_DIR}/config"

"${PORTAUDIO\_INCLUDE\_DIR}")

target\_link\_libraries(LibSampleApp

DefaultClient

AVSGatewayManager

CapabilitiesDelegate

CBLAuthDelegate

SQLiteStorage

SynchronizeStateSender

EqualizerImplementations

InterruptModel

"${PORTAUDIO\_LIB\_PATH}")

if (PCC)

target\_link\_libraries(LibSampleApp AVSCommon)

endif()

if (MCC)

target\_link\_libraries(LibSampleApp AVSCommon)

endif()

if (ANDROID)

target\_include\_directories(LibSampleApp PUBLIC "${AndroidUtilities}/include")

target\_link\_libraries(LibSampleApp AndroidUtilities)

if (ANDROID\_MEDIA\_PLAYER)

target\_include\_directories(LibSampleApp PUBLIC "${AndroidSLESMediaPlayer}/include")

target\_link\_libraries(LibSampleApp AndroidSLESMediaPlayer)

endif()

endif()

if (GSTREAMER\_MEDIA\_PLAYER)

target\_include\_directories(LibSampleApp PUBLIC "${MediaPlayer\_SOURCE\_DIR}/include")

target\_link\_libraries(LibSampleApp MediaPlayer)

endif()

if(KWD)

target\_link\_libraries(LibSampleApp KeywordDetectorProvider)

endif()

if (${CMAKE\_SYSTEM\_NAME} MATCHES "Darwin")

target\_link\_libraries(LibSampleApp

"-framework CoreAudio"

"-framework AudioToolbox"

"-framework AudioUnit"

"-framework CoreServices"

"-framework Carbon")

elseif(${CMAKE\_SYSTEM\_NAME} MATCHES "Linux")

target\_link\_libraries(LibSampleApp

rt m pthread asound atomic)

endif()

add\_executable(SampleApp

main.cpp)

target\_link\_libraries(SampleApp LibSampleApp)

# install target

asdk\_install\_targets(LibSampleApp TRUE)

target\_link\_libraries(SampleApp "-lwiringPi")

target\_link\_libraries(SampleApp "-ljsoncpp")