# This file controls Flutter-level build steps. It should not be edited.

cmake\_minimum\_required(VERSION 3.10)

set(EPHEMERAL\_DIR "${CMAKE\_CURRENT\_SOURCE\_DIR}/ephemeral")

# Configuration provided via flutter tool.

include(${EPHEMERAL\_DIR}/generated\_config.cmake)

# TODO: Move the rest of this into files in ephemeral. See

# https://github.com/flutter/flutter/issues/57146.

# Serves the same purpose as list(TRANSFORM ... PREPEND ...),

# which isn't available in 3.10.

function(list\_prepend LIST\_NAME PREFIX)

set(NEW\_LIST "")

foreach(element ${${LIST\_NAME}})

list(APPEND NEW\_LIST "${PREFIX}${element}")

endforeach(element)

set(${LIST\_NAME} "${NEW\_LIST}" PARENT\_SCOPE)

endfunction()

# === Flutter Library ===

# System-level dependencies.

find\_package(PkgConfig REQUIRED)

pkg\_check\_modules(GTK REQUIRED IMPORTED\_TARGET gtk+-3.0)

pkg\_check\_modules(GLIB REQUIRED IMPORTED\_TARGET glib-2.0)

pkg\_check\_modules(GIO REQUIRED IMPORTED\_TARGET gio-2.0)

set(FLUTTER\_LIBRARY "${EPHEMERAL\_DIR}/libflutter\_linux\_gtk.so")

# Published to parent scope for install step.

set(FLUTTER\_LIBRARY ${FLUTTER\_LIBRARY} PARENT\_SCOPE)

set(FLUTTER\_ICU\_DATA\_FILE "${EPHEMERAL\_DIR}/icudtl.dat" PARENT\_SCOPE)

set(PROJECT\_BUILD\_DIR "${PROJECT\_DIR}/build/" PARENT\_SCOPE)

set(AOT\_LIBRARY "${PROJECT\_DIR}/build/lib/libapp.so" PARENT\_SCOPE)

list(APPEND FLUTTER\_LIBRARY\_HEADERS

"fl\_basic\_message\_channel.h"

"fl\_binary\_codec.h"

"fl\_binary\_messenger.h"

"fl\_dart\_project.h"

"fl\_engine.h"

"fl\_json\_message\_codec.h"

"fl\_json\_method\_codec.h"

"fl\_message\_codec.h"

"fl\_method\_call.h"

"fl\_method\_channel.h"

"fl\_method\_codec.h"

"fl\_method\_response.h"

"fl\_plugin\_registrar.h"

"fl\_plugin\_registry.h"

"fl\_standard\_message\_codec.h"

"fl\_standard\_method\_codec.h"

"fl\_string\_codec.h"

"fl\_value.h"

"fl\_view.h"

"flutter\_linux.h"

)

list\_prepend(FLUTTER\_LIBRARY\_HEADERS "${EPHEMERAL\_DIR}/flutter\_linux/")

add\_library(flutter INTERFACE)

target\_include\_directories(flutter INTERFACE

"${EPHEMERAL\_DIR}"

)

target\_link\_libraries(flutter INTERFACE "${FLUTTER\_LIBRARY}")

target\_link\_libraries(flutter INTERFACE

PkgConfig::GTK

PkgConfig::GLIB

PkgConfig::GIO

)

add\_dependencies(flutter flutter\_assemble)

# === Flutter tool backend ===

# \_phony\_ is a non-existent file to force this command to run every time,

# since currently there's no way to get a full input/output list from the

# flutter tool.

add\_custom\_command(

OUTPUT ${FLUTTER\_LIBRARY} ${FLUTTER\_LIBRARY\_HEADERS}

${CMAKE\_CURRENT\_BINARY\_DIR}/\_phony\_

COMMAND ${CMAKE\_COMMAND} -E env

${FLUTTER\_TOOL\_ENVIRONMENT}

"${FLUTTER\_ROOT}/packages/flutter\_tools/bin/tool\_backend.sh"

${FLUTTER\_TARGET\_PLATFORM} ${CMAKE\_BUILD\_TYPE}

VERBATIM

)

add\_custom\_target(flutter\_assemble DEPENDS

"${FLUTTER\_LIBRARY}"

${FLUTTER\_LIBRARY\_HEADERS}

)