set(CMAKE\_MODULE\_PATH ${PROJECT\_SOURCE\_DIR})

cmake\_minimum\_required(VERSION 2.8 FATAL\_ERROR)

cmake\_policy(VERSION 2.8)

set(CMAKE\_MACOSX\_RPATH 1)

file(GLOB\_RECURSE sources \*.cpp)

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

execute\_process(

COMMAND

$ENV{LIBTWML\_HOME}/src/ops/scripts/get\_inc.sh

RESULT\_VARIABLE

TF\_RES

OUTPUT\_VARIABLE

TF\_INC)

if (NOT (${TF\_RES} EQUAL "0"))

message(${TF\_RES})

message(FATAL\_ERROR "Failed to get include path for tensorflow")

endif()

execute\_process(

COMMAND

$ENV{LIBTWML\_HOME}/src/ops/scripts/get\_lib.sh

RESULT\_VARIABLE

TF\_RES

OUTPUT\_VARIABLE

TF\_LIB)

if (NOT (${TF\_RES} EQUAL "0"))

message(${TF\_RES})

message(FATAL\_ERROR "Failed to get lib path for tensorflow")

endif()

find\_path(

TWML\_INC

NAMES "twml.h"

PATHS $ENV{LIBTWML\_HOME}/include)

add\_library(twml\_tf MODULE ${sources})

set(CMAKE\_MODULE\_PATH ${CMAKE\_MODULE\_PATH} "$ENV{LIBTWML\_HOME}/cmake")

if (UNIX)

if (APPLE)

set (CMAKE\_CXX\_FLAGS "-undefined dynamic\_lookup -stdlib=libc++ ${CMAKE\_CXX\_FLAGS}")

# -Wl,-all\_load ensures symbols not used by twml\_tf are also included.

# -Wl,-noall\_load limits the scope of the previous flag.

set (LINK\_ALL\_OPTION "-Wl,-all\_load")

set (NO\_LINK\_ALL\_OPTION "-Wl,-noall\_load")

set(TF\_FRAMEWORK\_LIB ${TF\_LIB}/libtensorflow\_framework.1.dylib)

else()

# -Wl,--whole-archive ensures symbols not used by twml\_tf are also included.

# -Wl,--no-whole-archive limits the scope of the previous flag.

set (LINK\_ALL\_OPTION "-Wl,--whole-archive")

set (NO\_LINK\_ALL\_OPTION "-Wl,--no-whole-archive")

set(TF\_FRAMEWORK\_LIB ${TF\_LIB}/libtensorflow\_framework.so.1)

endif()

endif()

target\_include\_directories(

twml\_tf

PRIVATE

${CMAKE\_CURRENT\_SOURCE\_DIR}

${TWML\_INC}

# TF\_INC needs to be the last to avoid some weird white-spacing issues with generated Makefile.

${TF\_INC} # Needed because of some header files auto-generated during build time.

${TF\_INC}/external/nsync/public/

)

target\_link\_libraries(twml\_tf

PUBLIC

# Since we are using twml\_tf as the "one" dynamic library,

# we want it to have the C function symbols needed for other functions as well.

${LINK\_ALL\_OPTION} twml ${NO\_LINK\_ALL\_OPTION}

${TF\_FRAMEWORK\_LIB}

)