# CMake build for libtiff

#

# Copyright © 2015 Open Microscopy Environment / University of Dundee

# Written by Roger Leigh <rleigh@codelibre.net>

#

# Permission to use, copy, modify, distribute, and sell this software and

# its documentation for any purpose is hereby granted without fee, provided

# that (i) the above copyright notices and this permission notice appear in

# all copies of the software and related documentation, and (ii) the names of

# Sam Leffler and Silicon Graphics may not be used in any advertising or

# publicity relating to the software without the specific, prior written

# permission of Sam Leffler and Silicon Graphics.

#

# THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND,

# EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY

# WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#

# IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR

# ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND,

# OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,

# WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF

# LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE

# OF THIS SOFTWARE.

include\_directories(${PROJECT\_SOURCE\_DIR}/libtiff

${PROJECT\_BINARY\_DIR}/libtiff

${CMAKE\_CURRENT\_BINARY\_DIR}

${TIFF\_INCLUDES})

# Test scripts (not used by CMake)

set(TESTSCRIPTS

bmp2tiff\_palette.sh

bmp2tiff\_rgb.sh

gif2tiff.sh

ppm2tiff\_pbm.sh

ppm2tiff\_pgm.sh

ppm2tiff\_ppm.sh

tiffcp-g3.sh

tiffcp-g3-1d.sh

tiffcp-g3-1d-fill.sh

tiffcp-g3-2d.sh

tiffcp-g3-2d-fill.sh

tiffcp-g4.sh

tiffcp-logluv.sh

tiffcp-thumbnail.sh

tiffcp-lzw-compat.sh

tiffdump.sh

tiffinfo.sh

tiffcp-split.sh

tiffcp-split-join.sh

tiff2ps-PS1.sh

tiff2ps-PS2.sh

tiff2ps-PS3.sh

tiff2ps-EPS1.sh

tiff2pdf.sh

tiffcrop-doubleflip-logluv-3c-16b.sh

tiffcrop-doubleflip-minisblack-1c-16b.sh

tiffcrop-doubleflip-minisblack-1c-8b.sh

tiffcrop-doubleflip-minisblack-2c-8b-alpha.sh

tiffcrop-doubleflip-miniswhite-1c-1b.sh

tiffcrop-doubleflip-palette-1c-1b.sh

tiffcrop-doubleflip-palette-1c-4b.sh

tiffcrop-doubleflip-palette-1c-8b.sh

tiffcrop-doubleflip-rgb-3c-16b.sh

tiffcrop-doubleflip-rgb-3c-8b.sh

tiffcrop-extract-logluv-3c-16b.sh

tiffcrop-extract-minisblack-1c-16b.sh

tiffcrop-extract-minisblack-1c-8b.sh

tiffcrop-extract-minisblack-2c-8b-alpha.sh

tiffcrop-extract-miniswhite-1c-1b.sh

tiffcrop-extract-palette-1c-1b.sh

tiffcrop-extract-palette-1c-4b.sh

tiffcrop-extract-palette-1c-8b.sh

tiffcrop-extract-rgb-3c-16b.sh

tiffcrop-extract-rgb-3c-8b.sh

tiffcrop-extractz14-logluv-3c-16b.sh

tiffcrop-extractz14-minisblack-1c-16b.sh

tiffcrop-extractz14-minisblack-1c-8b.sh

tiffcrop-extractz14-minisblack-2c-8b-alpha.sh

tiffcrop-extractz14-miniswhite-1c-1b.sh

tiffcrop-extractz14-palette-1c-1b.sh

tiffcrop-extractz14-palette-1c-4b.sh

tiffcrop-extractz14-palette-1c-8b.sh

tiffcrop-extractz14-rgb-3c-16b.sh

tiffcrop-extractz14-rgb-3c-8b.sh

tiffcrop-R90-logluv-3c-16b.sh

tiffcrop-R90-minisblack-1c-16b.sh

tiffcrop-R90-minisblack-1c-8b.sh

tiffcrop-R90-minisblack-2c-8b-alpha.sh

tiffcrop-R90-miniswhite-1c-1b.sh

tiffcrop-R90-palette-1c-1b.sh

tiffcrop-R90-palette-1c-4b.sh

tiffcrop-R90-palette-1c-8b.sh

tiffcrop-R90-rgb-3c-16b.sh

tiffcrop-R90-rgb-3c-8b.sh

tiff2rgba-logluv-3c-16b.sh

tiff2rgba-minisblack-1c-16b.sh

tiff2rgba-minisblack-1c-8b.sh

tiff2rgba-minisblack-2c-8b-alpha.sh

tiff2rgba-miniswhite-1c-1b.sh

tiff2rgba-palette-1c-1b.sh

tiff2rgba-palette-1c-4b.sh

tiff2rgba-palette-1c-8b.sh

tiff2rgba-rgb-3c-16b.sh

tiff2rgba-rgb-3c-8b.sh

tiff2rgba-quad-tile.jpg.sh)

# This list should contain all of the TIFF files in the 'images'

# subdirectory which are intended to be used as input images for

# tests. All of these files should use the extension ".tiff".

set(TIFFIMAGES

images/logluv-3c-16b.tiff

images/minisblack-1c-16b.tiff

images/minisblack-1c-8b.tiff

images/minisblack-2c-8b-alpha.tiff

images/miniswhite-1c-1b.tiff

images/palette-1c-1b.tiff

images/palette-1c-4b.tiff

images/palette-1c-8b.tiff

images/rgb-3c-16b.tiff

images/rgb-3c-8b.tiff

images/quad-tile.jpg.tiff

images/quad-lzw-compat.tiff)

set(BMPIMAGES

images/palette-1c-8b.bmp

images/rgb-3c-8b.bmp)

set(GIFIMAGES

images/palette-1c-8b.gif)

set(PNMIMAGES

images/minisblack-1c-8b.pgm

images/miniswhite-1c-1b.pbm

images/rgb-3c-8b.ppm)

# All uncompressed image files

set(UNCOMPRESSEDIMAGES

images/minisblack-1c-16b.tiff

images/minisblack-1c-8b.tiff

images/miniswhite-1c-1b.tiff

images/palette-1c-1b.tiff

images/palette-1c-4b.tiff

images/palette-1c-8b.tiff

images/rgb-3c-8b.tiff)

# This list should include all of the files in the 'images'

# subdirectory which are intended to be distributed. This may include

# files which are not currently used by the tests.

set(IMAGES\_EXTRA\_DIST

images/README.txt

${BMPIMAGES}

${GIFIMAGES}

${PNMIMAGES}

${TIFFIMAGES})

extra\_dist(

${TESTSCRIPTS}

${IMAGES\_EXTRA\_DIST}

common.sh)

set(noinst\_HEADERS tifftest.h)

add\_executable(ascii\_tag ascii\_tag.c)

target\_link\_libraries(ascii\_tag tiff port)

add\_executable(long\_tag long\_tag.c check\_tag.c)

target\_link\_libraries(long\_tag tiff port)

add\_executable(short\_tag short\_tag.c check\_tag.c)

target\_link\_libraries(short\_tag tiff port)

add\_executable(strip\_rw strip\_rw.c strip.c test\_arrays.c test\_arrays.h)

target\_link\_libraries(strip\_rw tiff port)

add\_executable(rewrite rewrite\_tag.c)

target\_link\_libraries(rewrite tiff port)

if(JPEG\_SUPPORT)

add\_executable(raw\_decode raw\_decode.c)

target\_link\_libraries(raw\_decode tiff port)

endif()

add\_executable(custom\_dir custom\_dir.c)

target\_link\_libraries(custom\_dir tiff port)

set(TEST\_OUTPUT "${CMAKE\_CURRENT\_BINARY\_DIR}/output")

file(MAKE\_DIRECTORY "${TEST\_OUTPUT}")

set(tiff\_test\_extra\_args

"-DTIFFCP=$<TARGET\_FILE:tiffcp>"

"-DTIFFINFO=$<TARGET\_FILE:tiffinfo>"

"-DTIFFSPLIT=$<TARGET\_FILE:tiffsplit>"

"-DLIBTIFF=$<TARGET\_FILE:tiff>")

if(WIN32)

list(APPEND tiff\_test\_extra\_args "-DWIN32=${WIN32}")

endif()

if(CYGWIN)

list(APPEND tiff\_test\_extra\_args "-DCYGWIN=${CYGWIN}")

endif()

macro(tiff\_test\_convert name command1 command2 command3 infile outfile validate)

add\_test(NAME "${name}"

COMMAND "${CMAKE\_COMMAND}"

"-DCONVERT\_COMMAND1=${command1}"

"-DCONVERT\_COMMAND2=${command2}"

"-DCONVERT\_COMMAND3=${command3}"

"-DINFILE=${infile}"

"-DOUTFILE=${outfile}"

"-DVALIDATE=${validate}"

${tiff\_test\_extra\_args}

-P "${CMAKE\_CURRENT\_SOURCE\_DIR}/TiffTest.cmake")

endmacro()

macro(tiff\_test\_stdout name command infile outfile)

add\_test(NAME "${name}"

COMMAND "${CMAKE\_COMMAND}"

"-DSTDOUT\_COMMAND=${command}"

"-DINFILE=${infile}"

"-DOUTFILE=${outfile}"

${tiff\_test\_extra\_args}

-P "${CMAKE\_CURRENT\_SOURCE\_DIR}/TiffTest.cmake")

endmacro()

macro(tiff\_test\_reader name command infile)

add\_test(NAME "${name}"

COMMAND "${CMAKE\_COMMAND}"

"-DREADER\_COMMAND=${command}"

"-DINFILE=${infile}"

${tiff\_test\_extra\_args}

-P "${CMAKE\_CURRENT\_SOURCE\_DIR}/TiffTest.cmake")

endmacro()

macro(add\_convert\_test\_multi commandname1 commandname2 commandname3

categoryname commandargs1 commandargs2 commandargs3

image validate)

string(REPLACE " " "^" escaped\_commandargs1 "${commandargs1}")

string(REPLACE " " "^" escaped\_commandargs2 "${commandargs2}")

string(REPLACE " " "^" escaped\_commandargs3 "${commandargs3}")

get\_filename\_component(name "${image}" NAME)

get\_filename\_component(base "${image}" NAME\_WE)

set(testname "${commandname1}-${categoryname}-${base}")

if(commandname1)

set(command1

"$<TARGET\_FILE:${commandname1}>^${escaped\_commandargs1}")

else()

set(command1)

endif()

if(commandname2)

set(command2

"$<TARGET\_FILE:${commandname2}>^${escaped\_commandargs2}")

else()

set(command2)

endif()

if(commandname3)

set(command3

"$<TARGET\_FILE:${commandname3}>^${escaped\_commandargs3}")

else()

set(command3)

endif()

set(infile "${CMAKE\_CURRENT\_SOURCE\_DIR}/${image}")

set(outfile "${TEST\_OUTPUT}/${commandname1}-${categoryname}-${name}")

string(REGEX REPLACE "\\.tiff\$" "" name "${name}")

tiff\_test\_convert("${testname}" "${command1}" "${command2}" "${command3}"

"${infile}" "${outfile}" "${validate}")

endmacro()

macro(add\_convert\_test commandname

categoryname commandargs

image validate)

add\_convert\_test\_multi("${commandname}" "" ""

"${categoryname}" "${commandargs}" "" ""

"${image}" "${validate}")

endmacro()

macro(add\_convert\_tests\_multi commandname commandname2 commandname3

categoryname

commandargs1 commandargs2 commandargs3

images validate)

foreach(file ${${images}})

add\_convert\_test\_multi("${commandname1}" "${commandname2}"

"${commandname3}" "${categoryname}"

"${commandargs1}" "${commandargs2}"

"${commandargs3}" "${file}" "${validate}")

endforeach()

endmacro()

macro(add\_convert\_tests commandname categoryname commandargs images validate)

foreach(file ${${images}})

add\_convert\_test("${commandname}" "${categoryname}"

"${commandargs}" "${file}" "${validate}")

endforeach()

endmacro()

macro(add\_stdout\_test commandname commandargs image)

string(REPLACE " " "^" escaped\_commandargs "${commandargs}")

get\_filename\_component(name "${image}" NAME)

get\_filename\_component(base "${image}" NAME\_WE)

set(testname "${commandname}-${base}")

set(command "$<TARGET\_FILE:${commandname}>^${escaped\_commandargs}")

set(infile "${CMAKE\_CURRENT\_SOURCE\_DIR}/${image}")

set(outfile "${TEST\_OUTPUT}/${commandname}-${name}")

string(REGEX REPLACE "\\.tiff\$" "" name "${name}")

tiff\_test\_stdout("${testname}" "${command}" "${infile}" "${outfile}")

endmacro()

macro(add\_reader\_test commandname commandargs image)

string(REPLACE " " "^" escaped\_commandargs "${commandargs}")

get\_filename\_component(name "${image}" NAME)

get\_filename\_component(base "${image}" NAME\_WE)

set(testname "${commandname}-${base}")

set(command "$<TARGET\_FILE:${commandname}>^${escaped\_commandargs}")

set(infile "${CMAKE\_CURRENT\_SOURCE\_DIR}/${image}")

string(REGEX REPLACE "\\.tiff\$" "" name "${name}")

tiff\_test\_reader("${testname}" "${command}" "${infile}")

endmacro()

# BMP

add\_convert\_test(bmp2tiff palette "" "images/palette-1c-8b.bmp" TRUE)

add\_convert\_test(bmp2tiff rgb "" "images/rgb-3c-8b.bmp" TRUE)

# GIF

add\_convert\_test(gif2tiff palette "" "images/palette-1c-8b.gif" TRUE)

# PPM

add\_convert\_test(ppm2tiff miniswhite "" "images/miniswhite-1c-1b.pbm" TRUE)

add\_convert\_test(ppm2tiff minisblack "" "images/minisblack-1c-8b.pgm" TRUE)

add\_convert\_test(ppm2tiff rgb "" "images/rgb-3c-8b.ppm" TRUE)

# tiffcp

add\_convert\_test(tiffcp g3 "-c g3" "images/miniswhite-1c-1b.tiff" FALSE)

add\_convert\_test(tiffcp g31d "-c g3:1d" "images/miniswhite-1c-1b.tiff" FALSE)

add\_convert\_test(tiffcp g31dfill "-c g3:1d:fill" "images/miniswhite-1c-1b.tiff" FALSE)

add\_convert\_test(tiffcp g32d "-c g3:2d" "images/miniswhite-1c-1b.tiff" FALSE)

add\_convert\_test(tiffcp g32dfill "-c g3:2d:fill" "images/miniswhite-1c-1b.tiff" FALSE)

add\_convert\_test(tiffcp g4 "-c g4" "images/miniswhite-1c-1b.tiff" FALSE)

add\_convert\_test(tiffcp none "-c none" "images/quad-lzw-compat.tiff" FALSE)

add\_convert\_test\_multi(tiffcp tiffcp "" logluv "-c none" "-c sgilog" ""

"images/logluv-3c-16b.tiff" FALSE)

add\_convert\_test\_multi(tiffcp thumbnail "" thumbnail "g3:1d" "" ""

"images/miniswhite-1c-1b.tiff" FALSE)

# tiffdump

add\_reader\_test(tiffdump "" "images/miniswhite-1c-1b.tiff")

# tiffinfo

add\_reader\_test(tiffinfo "-c -D -d -j -s" "images/minisblack-1c-16b.tiff")

# tiffcp split/join

foreach(image ${UNCOMPRESSEDIMAGES})

list(APPEND ESCAPED\_UNCOMPRESSED "${CMAKE\_CURRENT\_SOURCE\_DIR}/${image}")

endforeach()

string(REPLACE ";" "^" ESCAPED\_UNCOMPRESSED "${ESCAPED\_UNCOMPRESSED}")

add\_test(NAME "tiffcp-split"

COMMAND "${CMAKE\_COMMAND}"

"-DTESTFILES=${ESCAPED\_UNCOMPRESSED}"

"-DCONJOINED=${TEST\_OUTPUT}/tiffcp-split-conjoined.tif"

"-DSPLITFILE=${TEST\_OUTPUT}/tiffcp-split-split-"

${tiff\_test\_extra\_args}

-P "${CMAKE\_CURRENT\_SOURCE\_DIR}/TiffSplitTest.cmake")

add\_test(NAME "tiffcp-split-join"

COMMAND "${CMAKE\_COMMAND}"

"-DTESTFILES=${ESCAPED\_UNCOMPRESSED}"

"-DCONJOINED=${TEST\_OUTPUT}/tiffcp-split-join-conjoined.tif"

"-DSPLITFILE=${TEST\_OUTPUT}/tiffcp-split-join-split-"

"-DRECONJOINED=${TEST\_OUTPUT}/tiffcp-split-join-reconjoined.tif"

${tiff\_test\_extra\_args}

-P "${CMAKE\_CURRENT\_SOURCE\_DIR}/TiffSplitTest.cmake")

# PDF

add\_stdout\_test(tiff2pdf "" "images/miniswhite-1c-1b.tiff" TRUE)

# RGBA

add\_convert\_tests(tiff2rgba default "" TIFFIMAGES TRUE)

# Test rotations

add\_convert\_tests(tiffcrop R90 "-R90" TIFFIMAGES TRUE)

# Test flip (mirror)

add\_convert\_tests(tiffcrop doubleflip "-F both" TIFFIMAGES TRUE)

# Test extracting a section 60 pixels wide and 60 pixels high

add\_convert\_tests(tiffcrop extract "-U px -E top -X 60 -Y 60" TIFFIMAGES TRUE)

# Test extracting the first and fourth quarters from the left side.

add\_convert\_tests(tiffcrop extractz14 "-E left -Z1:4,2:4" TIFFIMAGES TRUE)