
Subject: Re: Building TheIDE with using CMake

Posted by [Sender Ghost](#) on Thu, 26 May 2016 13:03:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Radek.

coolman wrote on Thu, 26 May 2016 06:16

Based on this discussion I created bash script to generate CMakeList.txt from the Ultimate++ project - see https://github.com/CoolmanCZ/upp_cmake

Looks like, you did a lot of work to automate CMakeLists.txt files generation. Very nice.

I managed to build TheIDE on FreeBSD 10.2 with using "Ninja" CMake generator (just for example), with some fixes. I added "link_directories(/usr/local/lib)" (because this is where local libraries reside on FreeBSD by default) and some checks for Clang compiler:

Toggle patch

```
diff -ruN upp_cmake.orig/GenerateCMakeFiles-lib.sh upp_cmake/GenerateCMakeFiles-lib.sh
```

```
--- upp_cmake.orig/GenerateCMakeFiles-lib.sh 2016-05-26 13:09:40
```

```
+++ upp_cmake/GenerateCMakeFiles-lib.sh
```

```
@@ -983,11 +983,12 @@
```

```
    echo >> ${OFN}
    echo '# Set compiler flags' >> ${OFN}
    echo 'if ( "${CMAKE_CXX_COMPILER_ID}" STREQUAL "Clang" )' >> ${OFN}
+   echo ' set ( CMAKE_COMPILER_IS_CLANG TRUE )' >> ${OFN}
    echo ' set ( CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Wno-logical-op-parentheses" )'
>> ${OFN}
    echo 'endif()' >> ${OFN}

    echo >> ${OFN}
-   echo 'if ( CMAKE_COMPILER_IS_GNUCC )' >> ${OFN}
+   echo 'if ( CMAKE_COMPILER_IS_GNUCC OR CMAKE_COMPILER_IS_CLANG )' >> ${OFN}
    echo ' set ( CMAKE_CXX_FLAGS_RELEASE "${CMAKE_CXX_FLAGS_RELEASE}
-std=c++11 -O3 -ffunction-sections -fdata-sections" )' >> ${OFN}
    echo ' set ( CMAKE_CXX_FLAGS_DEBUG "${CMAKE_CXX_FLAGS_DEBUG} -std=c++11
-O0" )' >> ${OFN}
    echo 'elseif ( MSVC )' >> ${OFN}
@@ -1045,6 +1046,10 @@
    echo '    include_directories( ${PNG_INCLUDE_DIR} )' >> ${OFN}
    echo '    list ( APPEND main_${LINK_LIST} ${PNG_LIBRARIES} )' >> ${OFN}
    echo ' endif()' >> ${OFN}
+   echo >> ${OFN}
+   echo ' if ( ${CMAKE_SYSTEM_NAME} MATCHES BSD )' >> ${OFN}
+   echo '    link_directories( /usr/local/lib )' >> ${OFN}
+   echo ' endif()' >> ${OFN}
    echo 'endif()' >> ${OFN}

    echo >> ${OFN}
```

Then I configured upp_cmake/GenerateCMakeFiles.sh file for FreeBSD, used it, generated CMake building files and make a build. Just example for current U++ nightly source files:

```
% cd upp_cmake
% sed -i '.bak' -e '/^UPP_SRC_DIR/s/=/?&/' -e 's/LINUX/BSD -DflagFREEBSD/'
GenerateCMakeFiles.sh
% wget http://www.ultimatepp.org/downloads/upp-x11-src-9886.tar.gz
% tar -xf upp-x11-src-9886.tar.gz
% env UPP_SRC_DIR=upp-x11-src-9886/uppsrc bash GenerateCMakeFiles.sh
% mkdir build && cd build && cmake -G Ninja ../
% ninja
```

Edit:

Submitted pull request with proposed patch in this message. And it was committed later.
