Subject: Re: Building TheIDE with using CMake
Posted by cyrion on Sat, 14 Feb 2015 00:05:10 GMT
View Forum Message <> Reply to Message

Quote:
What I said might look like as rudeness, but it is not.

No problem, I was not there for a few hours, I just saw your answer now.
Thank you for the updated links!

Quote:
There were many CMake releases (and possible compatibility issues for previous versions) from 2011 year.
Constantly updating the attachments for new versions of U++ and CMake is not productive, in my opinion, especially when done manually.

This topic is open for questions, related to the topic messages. But they might be quite outdated, when used with modern U++ and CMake.

If you have a (new) method or a way how to use CMake and U++, then better to create a new topic about this, in my opinion.

From my side, I'm just happy I found a solution to my problem with your help!
In fact I started creating my CMakeFiles without knowing you already explored that problem, then I found this forum entry when I looked for a solution to my colors bug.
Thought it was the good place to ask - and in fact it was :) Thanks again!

I really love the UPP way to create nice and small GUI apps without external dependencies, and I just wanted to make this working with QtCreator.
TheIDE is great, fast & smooth ; I like the nested code blocks with the different colors, the << completion and the indispensable statistics and elapsed time :) but IMHO it lacks some recent functionalities I'm really used to: easy global/local search & replace, simultaneous multiple lines editing (aka "column mode" or "block selection mode"), refactoring, ctrk/k+key stuff, VIM integration, etc. Also JOM for parallel builds is probably as fast as BLITZ at first sight, maybe faster. I'll check that but I first have to add the PCH to be fair :)

So my goal here was not to automatically create CMakeFiles from upp projects, I'm good with the idea of maintaining my CMakeFiles manually, but of course that's only a workaround.

About my final solution - no competition here ;) - I updated your function add_init_file for this one:

```
function(create_cpps_from_icpps )
   file( GLOB icpp_files RELATIVE "${CMAKE_CURRENT_SOURCE_DIR}"
"${CMAKE_CURRENT_SOURCE_DIR}/*.icpp"   )
   foreach( icppFile ${icpp_files} )
      set(output_file "${CMAKE_CURRENT_BINARY_DIR}/${icppFile}.cpp")
      file(WRITE "${output_file}" "#include
\"${CMAKE_CURRENT_SOURCE_DIR}/${icppFile}\"\n")
```

```
    endforeach()
endfunction()
```

It creates the icpp.cpp files in the binary dir. Each one of them having the name of the original icpp it makes easier to catch compile / link errors compared to the multiple init.cpp that had the same name.

So as you may have noticed, I do not rely on Mirek ini files. That's probably not a good idea, but when I tried using them it added a lot of unnecessary (in my case) dependencies, missing symbols and also multiple symbols definitions that broke everything.
I didn't really digged very far to check why, but I think ini files can potentially add the same cpp multiple times.
The way I do makes possible to only link with the required modules, and not the others.

To finish, my main app looks like that :

```
set( sources
 SQLApp.h
 SQLApp.sch
 query.cpp
 book.cpp
 borrow.cpp
 main.cpp
)

file( GLOB_RECURSE ini_files "${CMAKE_CURRENT_BINARY_DIR}/../*.icpp.cpp"   )

add_executable( SQLApp WIN32 ${sources} ${ini_files} )
```

The GLOB_RECURSE stuff is not very pretty, but it does the job: it links the icpp to the app the exact same way that fixed my problem at first.

For now its working, but I only use a subset of UPP, namely :

```
add_subdirectory( "plugin/z"  )
add_subdirectory( Core )
add_subdirectory( Draw )
add_subdirectory( "plugin/bmp"  )
add_subdirectory( "plugin/png"  )
add_subdirectory( RichText )
add_subdirectory( CtrlCore )
add_subdirectory( CtrlLib  )
add_subdirectory( "plugin/sqlite3"  )
add_subdirectory( Sql  )
add_subdirectory( SqlCtrl  )
add_subdirectory( Report  )
add_subdirectory( CodeEditor  )
add_subdirectory( "plugin/pcre" )
```

add_subdirectory("plugin/zip")

So maybe it will break in the future, I don't know!