

Troubleshooting Guide for CPP project

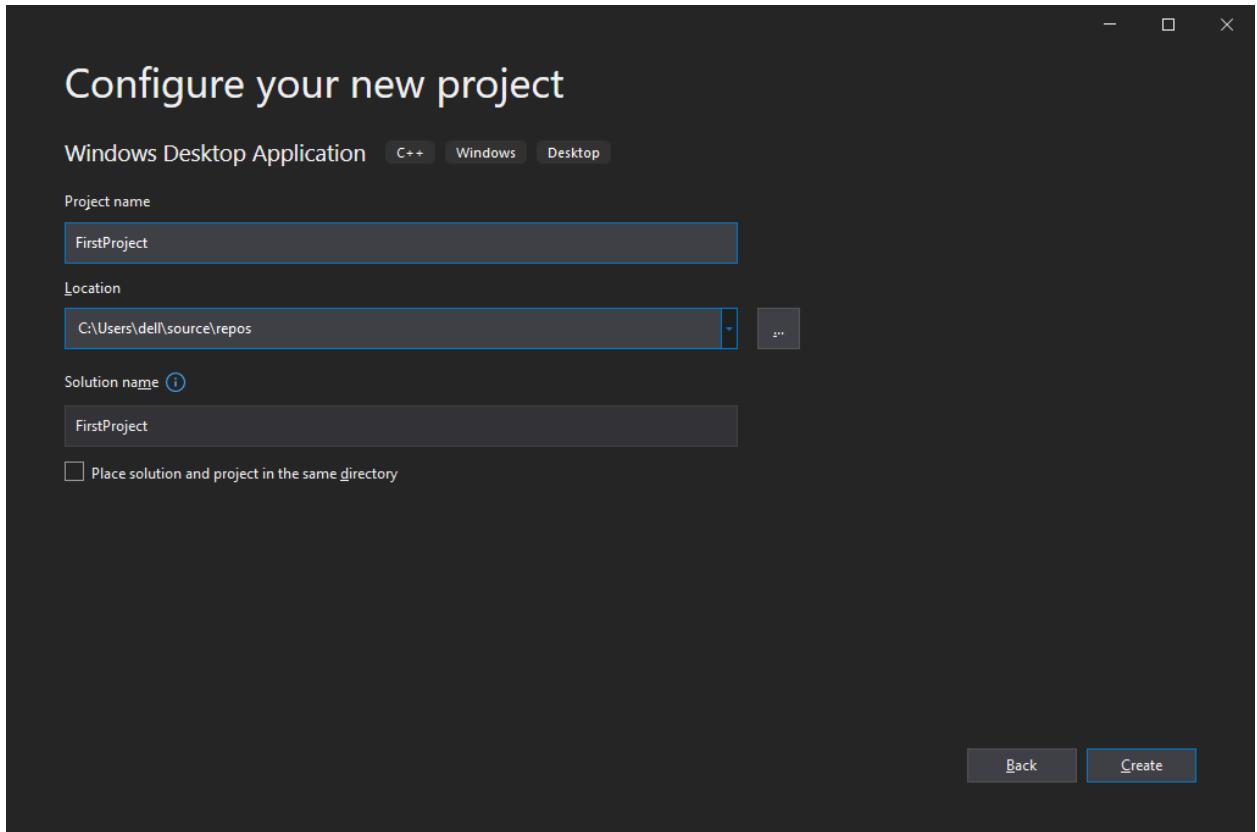
Eric.S

eUgeigne

1. Building dependencies.

- 1) Building multiple projects in a solution
- *Creating multiple projects in a solution*

You can create a project in a solution like this.

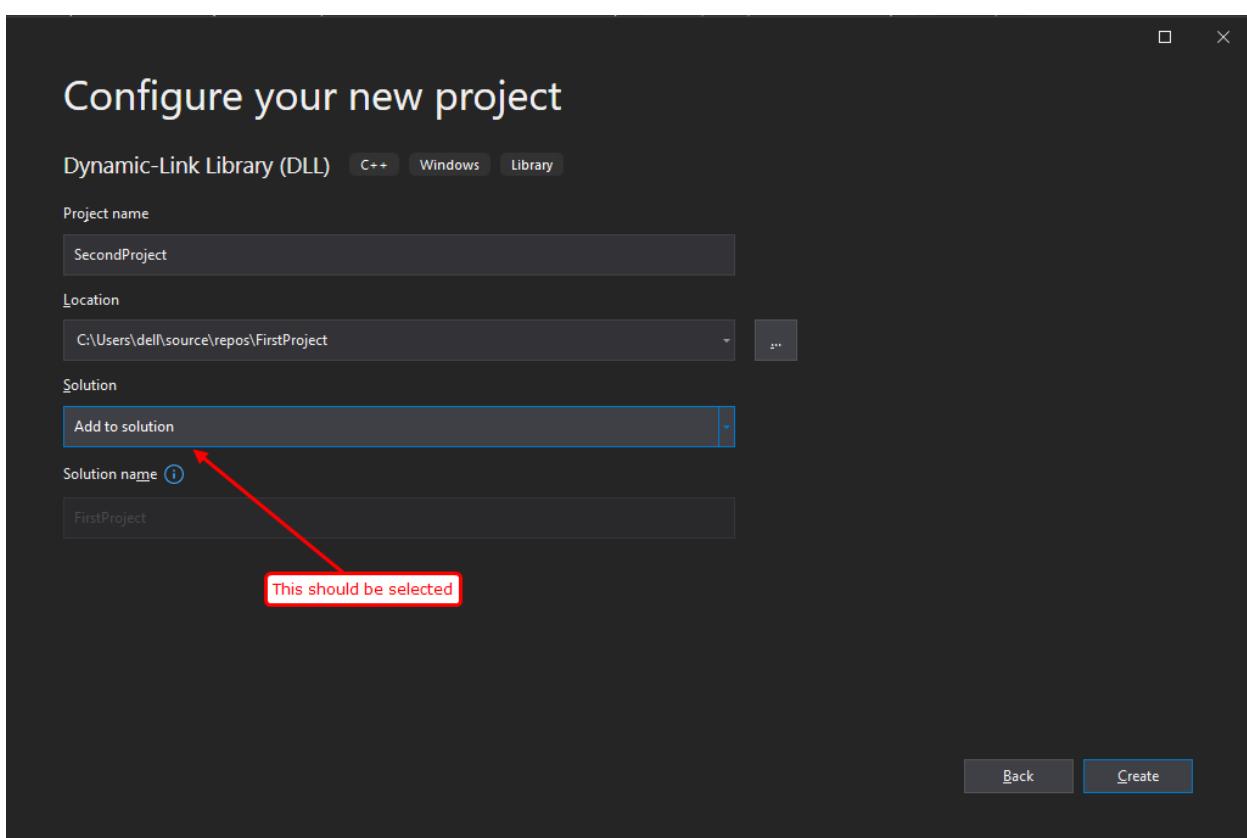
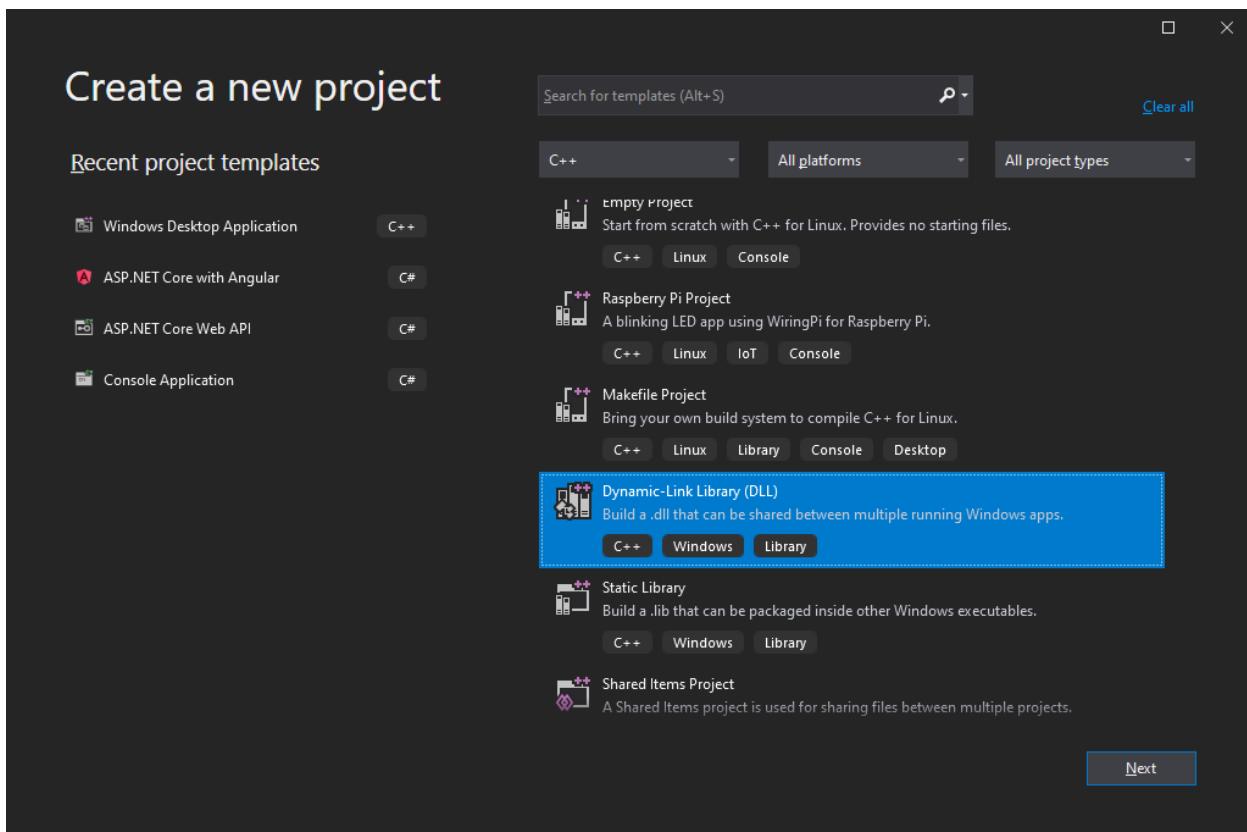


Then this solution has one project in it.

A screenshot of the Visual Studio IDE interface. The title bar says "FirstProject". The left sidebar shows "Server Explorer" and "Toolbox". The main area is a code editor with "FirstProject.cpp" open, containing C++ code for a Windows application. The code includes declarations for global variables like hInst, szTitle, and szWindowClass, and function prototypes for MyRegisterClass, InitInstance, WndProc, and About. It also defines the wWinMain entry point. The status bar at the bottom indicates "Ready". On the right side, the "Solution Explorer" window is open, showing a single project node named "FirstProject" under "Solution 'FirstProject' (1 of 1 project)". A red arrow points from the text "one project" to the "FirstProject" node in the Solution Explorer. The "Properties" and "Git Changes" tabs are also visible in the Solution Explorer.

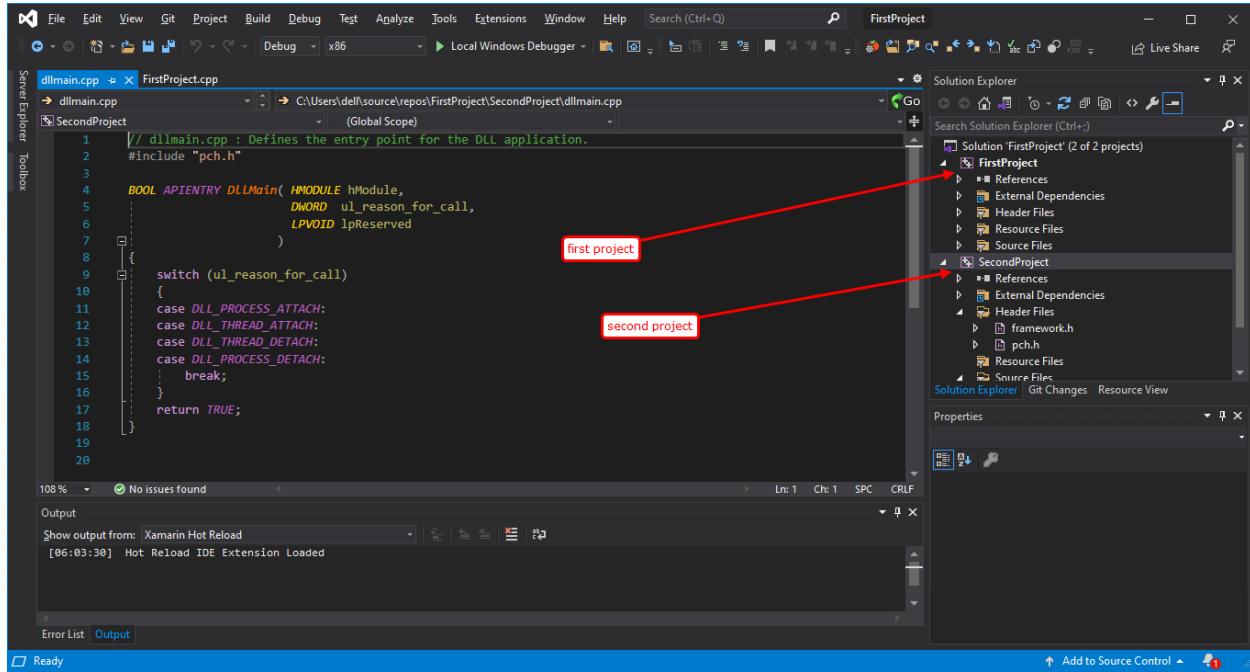
You can create a second project in this solution.

A screenshot of the Visual Studio IDE interface, similar to the previous one but with a different focus. The title bar says "FirstProject". The left sidebar shows "Server Explorer" and "Toolbox". The "File" menu is open, with "New Project..." highlighted. The main area is a code editor with "FirstProject.cpp" open, showing the same C++ code as before. The status bar at the bottom indicates "Ready". On the right side, the "Solution Explorer" window is open, showing the "FirstProject" node under "Solution 'FirstProject' (1 of 1 project)". The "Properties" and "Git Changes" tabs are visible in the Solution Explorer. The "Output" window at the bottom shows "Xamarin Hot Reload" and "[06:03:30] Hot Reload IDE Extension Loaded".

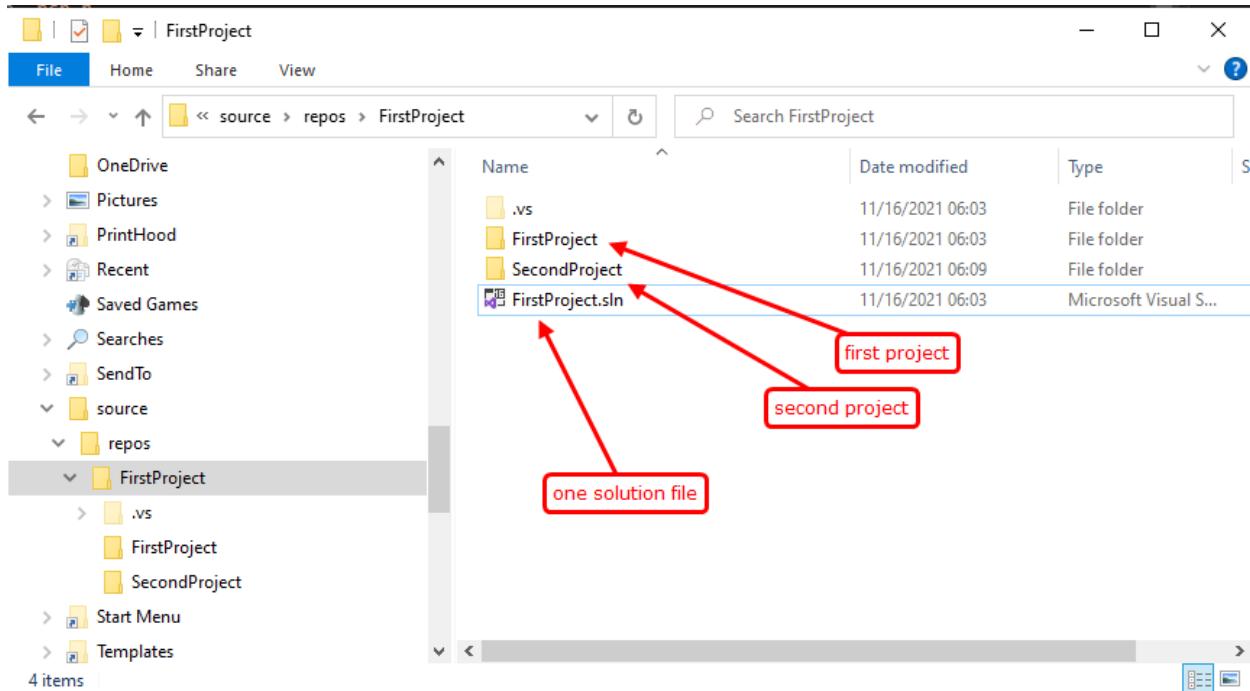


Don't forget to set the selection to "Add to solution".

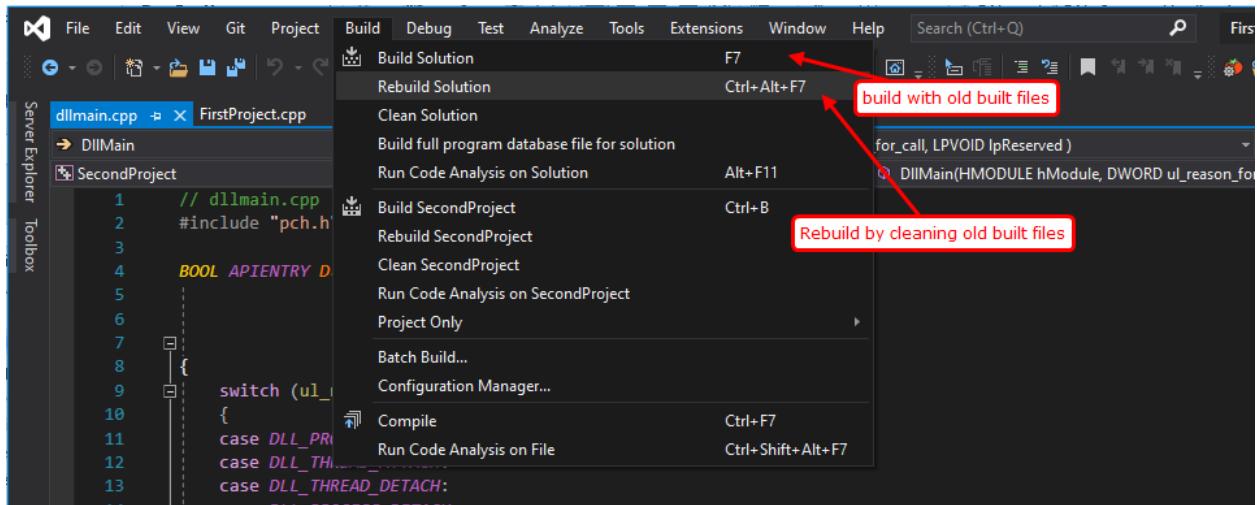
Now you can see two projects in a solution.



Two projects are located in explorer like



- How to build and run an individual project



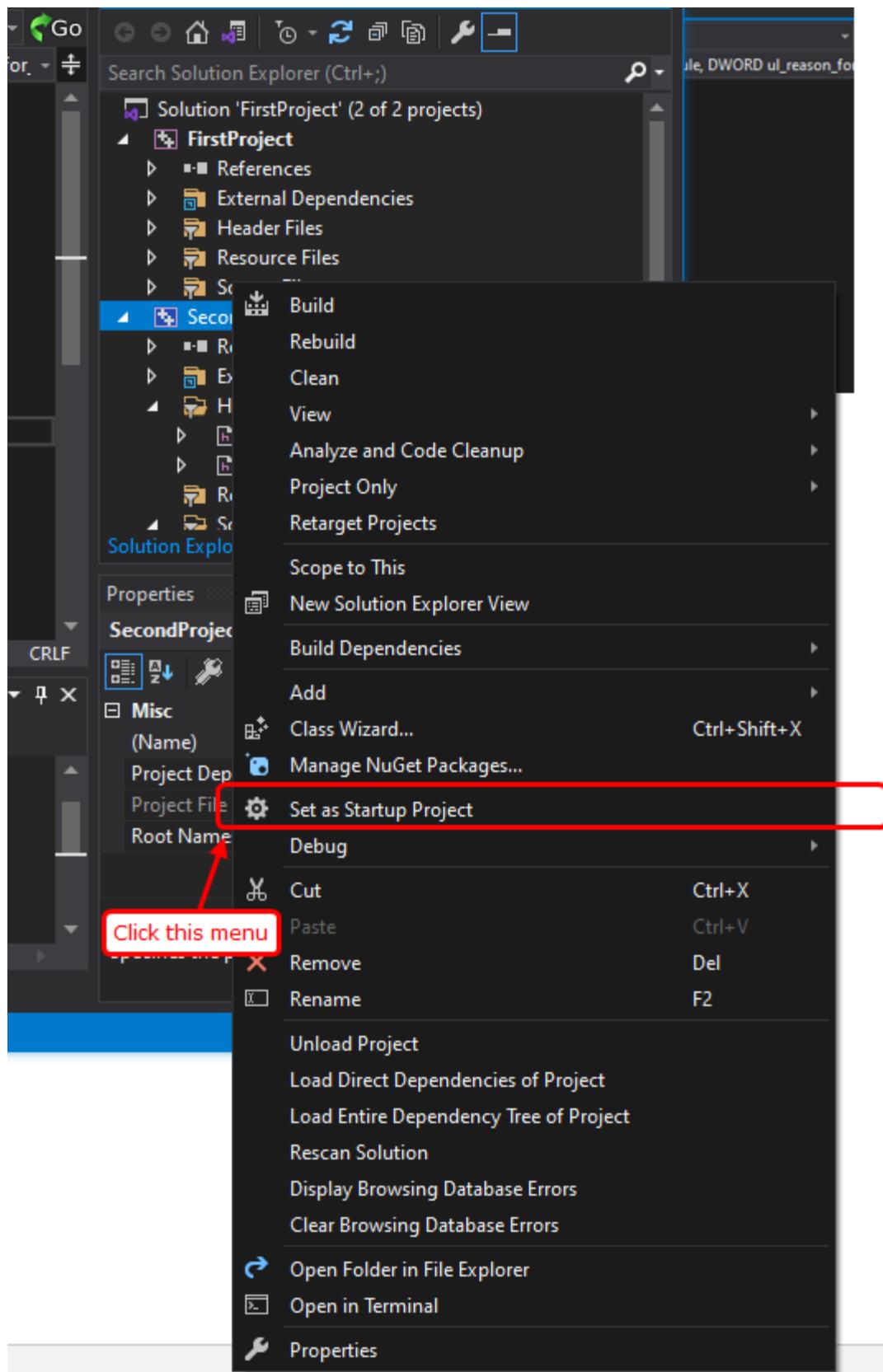
As you can see the picture above, you may press F7 key or “Ctrl + Alt + F7” key to built or rebuild project.

This will build all the projects in a solution.

To run the project, you should select the project to run.

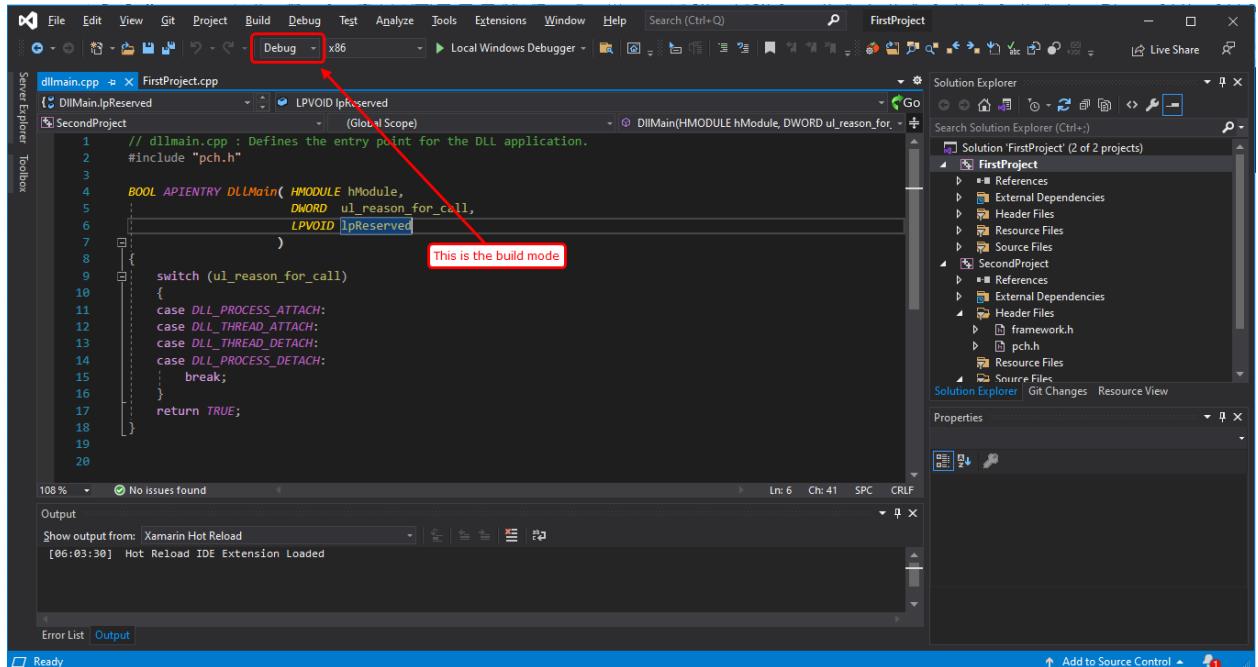
It is called “active” project or “startup” project.

If you want to run second project, please select second project as “active”.

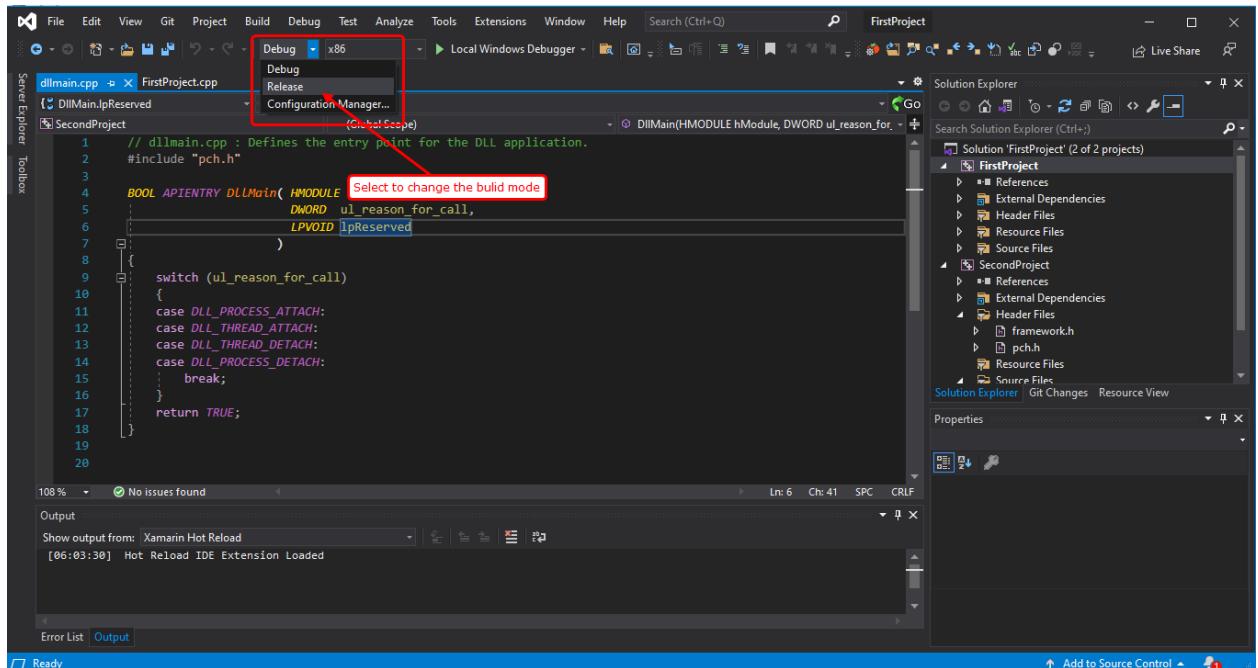


- *How to change the build mode*

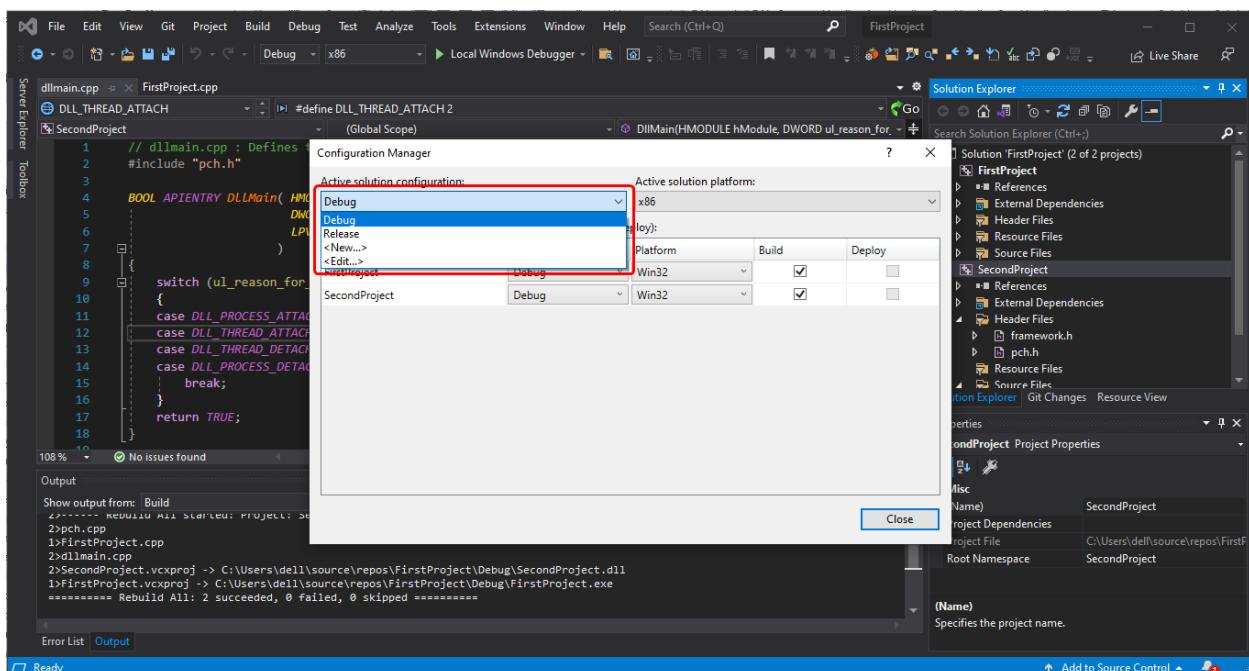
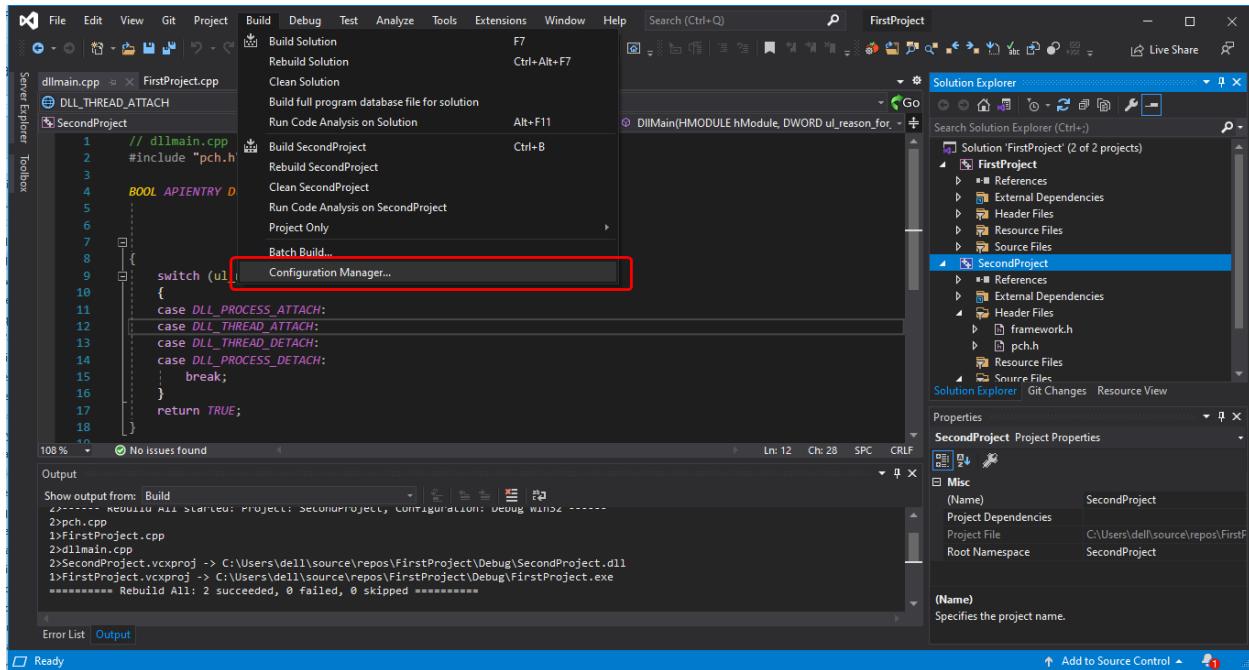
By standards, Visual Studio has 2 modes for windows application. "Debug" and "Release".



You can change the build mode by this dropdown list.



Or you can change build mode by the Build menu.



- *What could change by build modes*

Debug and Release modes build the project and make cpp application in “Debug” or “Release” folder.

But the size of the application varies to the build mode.

In Debug mode, Visual Studio adds some debug information to output application, such as function names, variable names, calling structures, command line numbers..

Furthermore, debug mode will make cpp app more analytic but it will cause its application speed slower.

In Release mode, Visual Studio does not add any debug information to output application.

Release mode makes output application less analytic and more optimized so that Release-built application runs faster than Debug-built application.

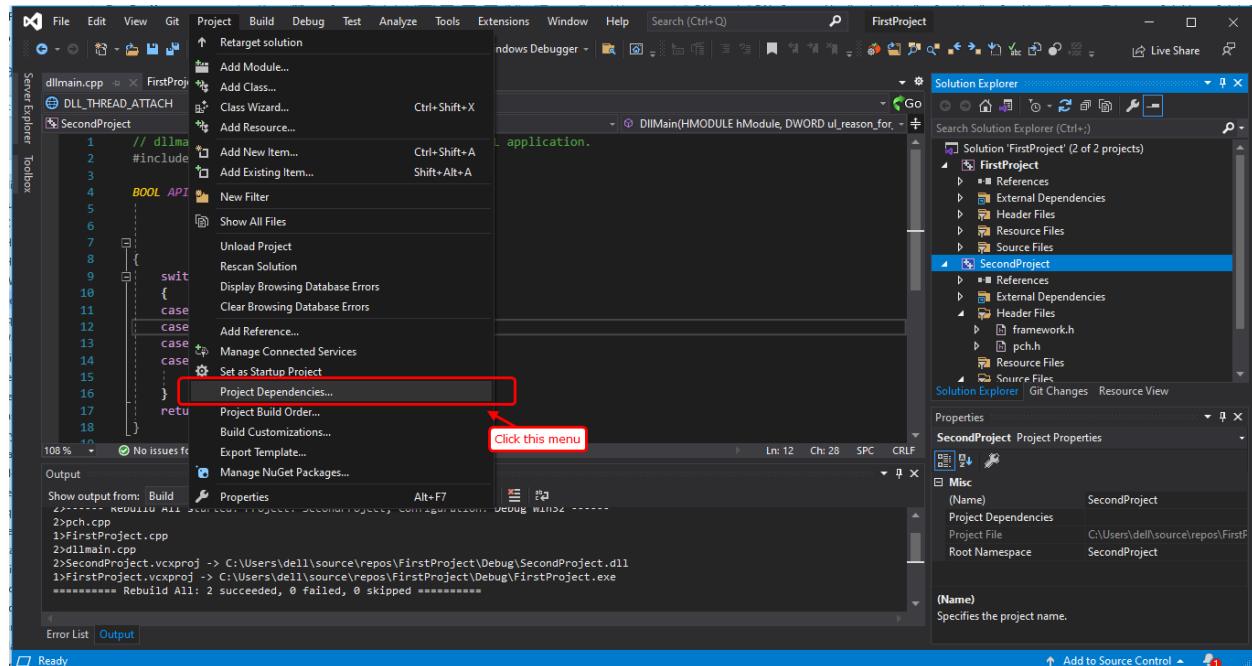
	Debug	Release
Size	Bigger	Smaller
Speed	Slower	Faster
analytic	enough	Hard
Use	Developing	Product

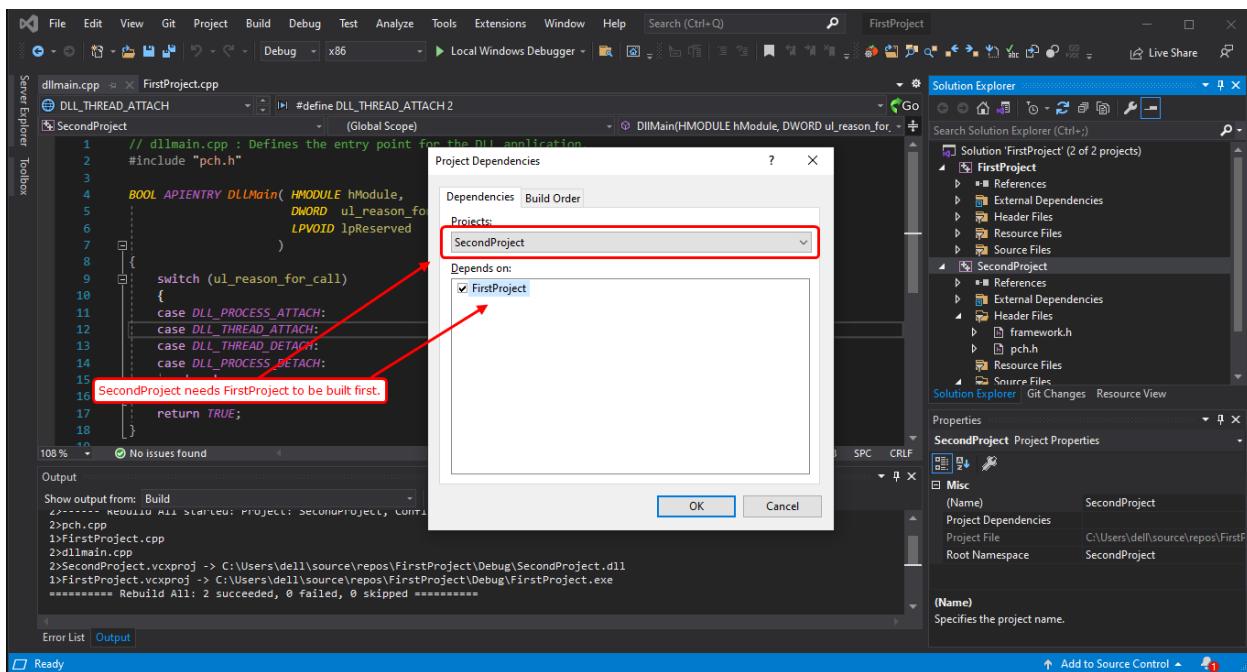
So while developing application, we usually use Debug mode because we need to debug and see the full information about errors.

But to make a product, we build in Release mode because it will run faster and less crackable to hackers because of its less analytic structure and information.

- *Setting dependencies for the project*

When you have multiple projects in a solution, you can build project in a specified order.





FirstProject is the one of the dependencies of SecondProject.

Because you set “SecondProject” as startup project, when you build SecondProject, FirstProject is built first and then SecondProject is built.

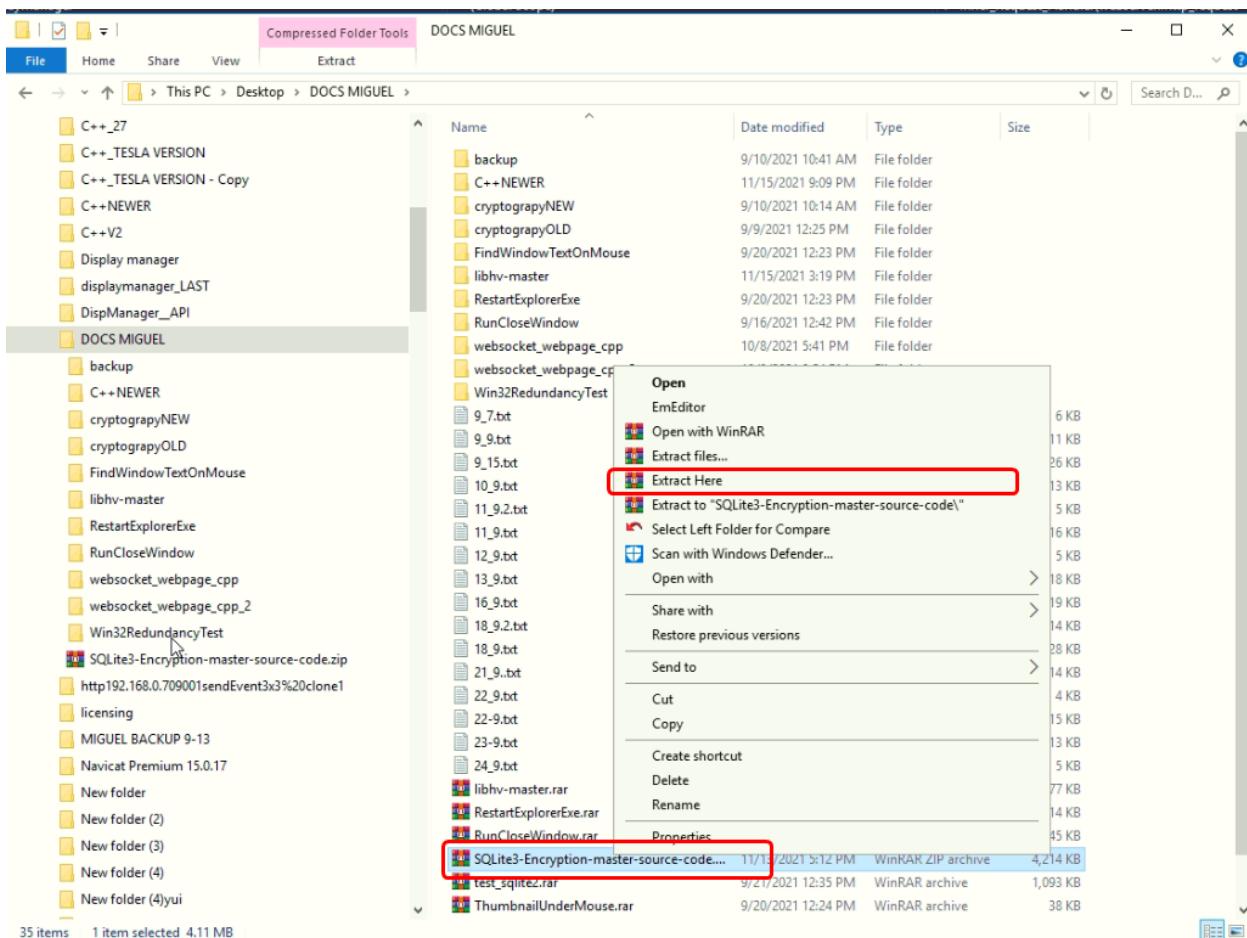
This is used when FirstProject is a library-project and SecondProject uses the library.

If no dependency is set to projects, multiple projects are built by order.

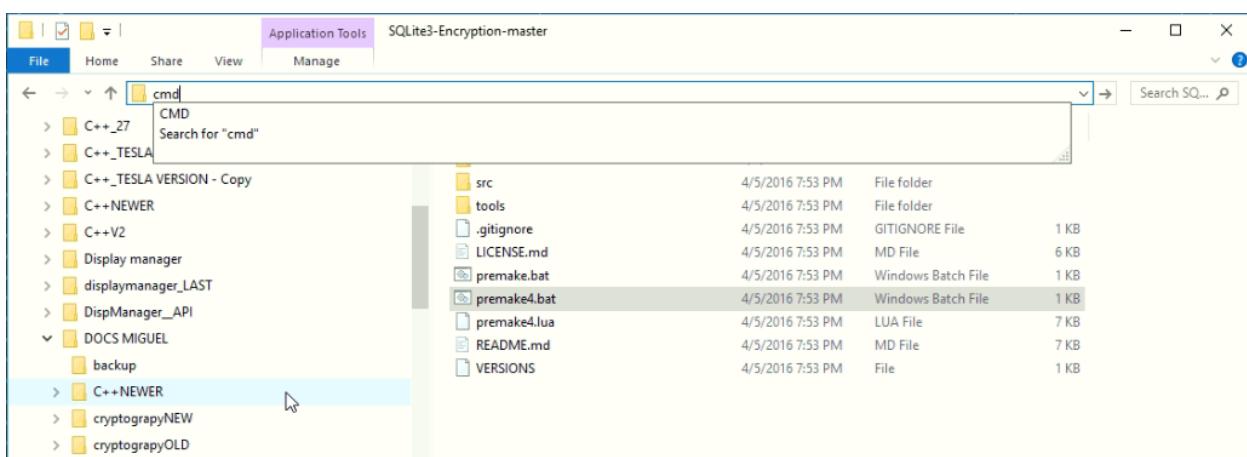
If dependency is set to projects, then Visual Studio reorders to build by the dependency of all the projects.

2) Sqlite library with encryption

First, extract the source project in some directory.



Run cmd console.



And then please run premake4.bat.

```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\DOCS MIGUEL\SQLite3-Encryption-master>premake4.bat
```

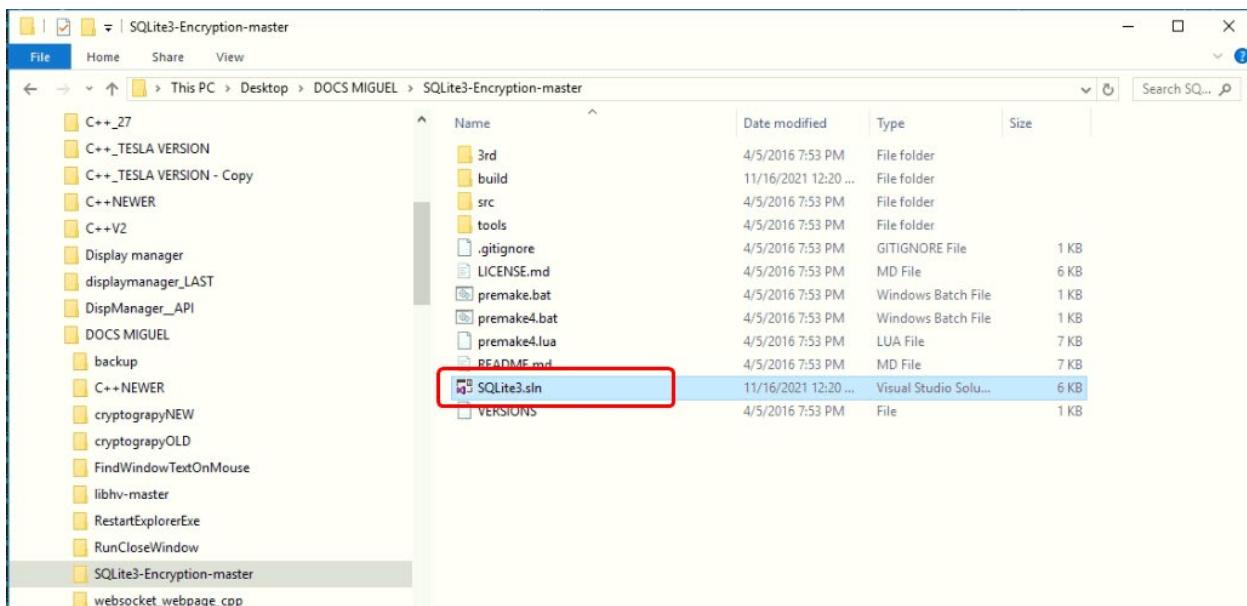
```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\DOCS MIGUEL\SQLite3-Encryption-master>premake4.bat
Building configurations...
Running action 'vs2012'...
Generating SQLite3.sln...
Generating build/sqlite3_lib/sqlite3_lib.vcxproj...
Generating build/sqlite3_lib/sqlite3_lib.vcxproj.user...
Generating build/sqlite3_lib/sqlite3_lib.vcxproj.filters...
Generating build/sqlite3_dll/sqlite3_dll.vcxproj...
Generating build/sqlite3_dll/sqlite3_dll.vcxproj.user...
Generating build/sqlite3_dll/sqlite3_dll.vcxproj.filters...
Generating build/sqlite3_shell/sqlite3_shell.vcxproj...
Generating build/sqlite3_shell/sqlite3_shell.vcxproj.user...
Generating build/sqlite3_shell/sqlite3_shell.vcxproj.filters...
Done.

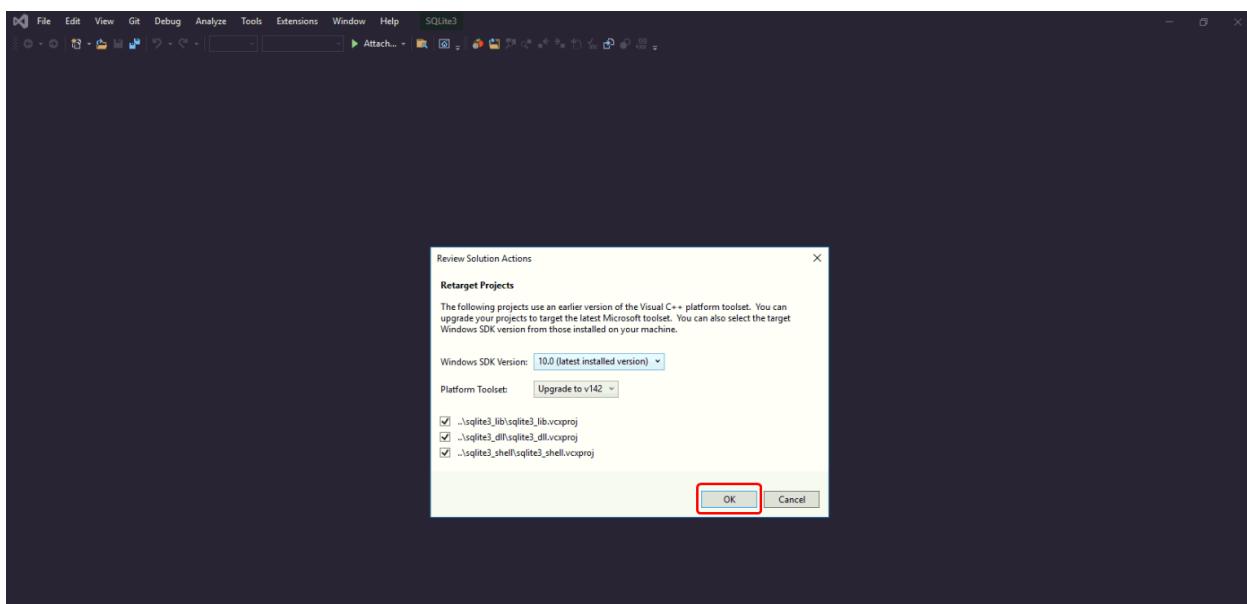
C:\Users\Administrator\Desktop\DOCS MIGUEL\SQLite3-Encryption-master>
```

One solution file and several project files are generated.

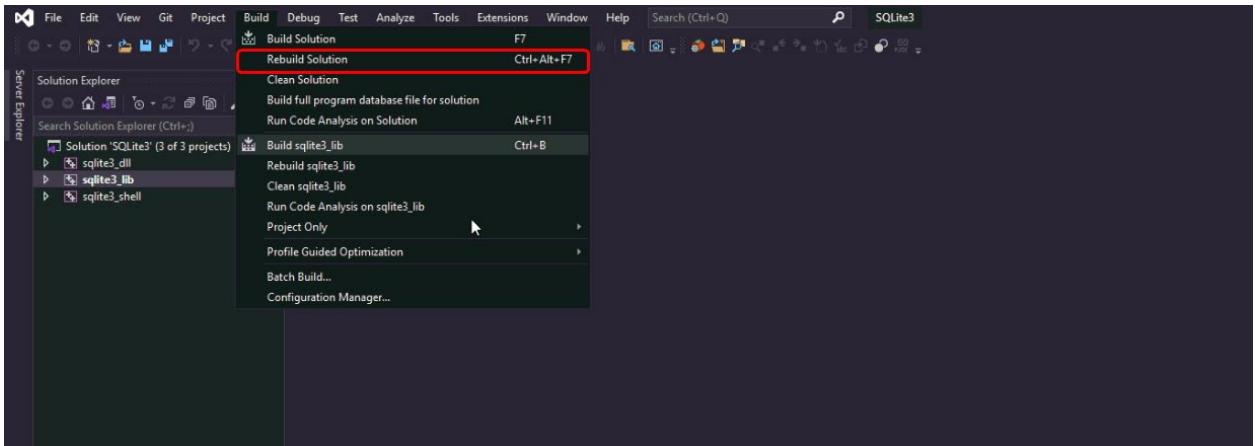
Double-click to open a visual studio solution.



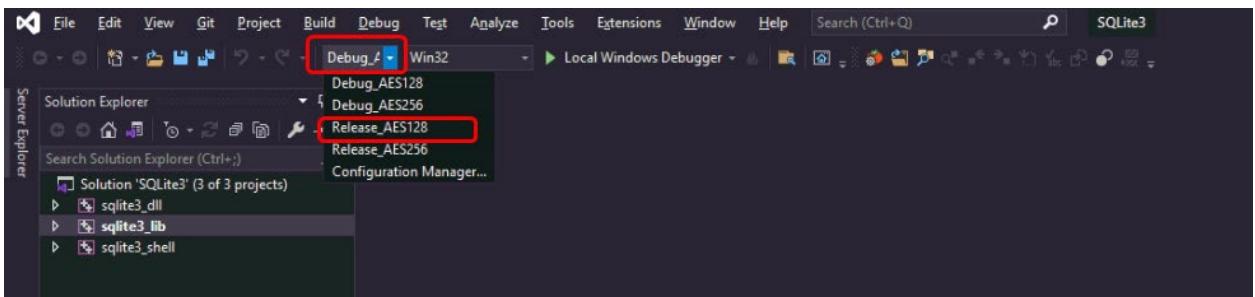
Convert the projects into the current Visual Studio version.



To build all the projects in a solution, Run the menu Build



To get faster running library, please change build mode to “Release”.

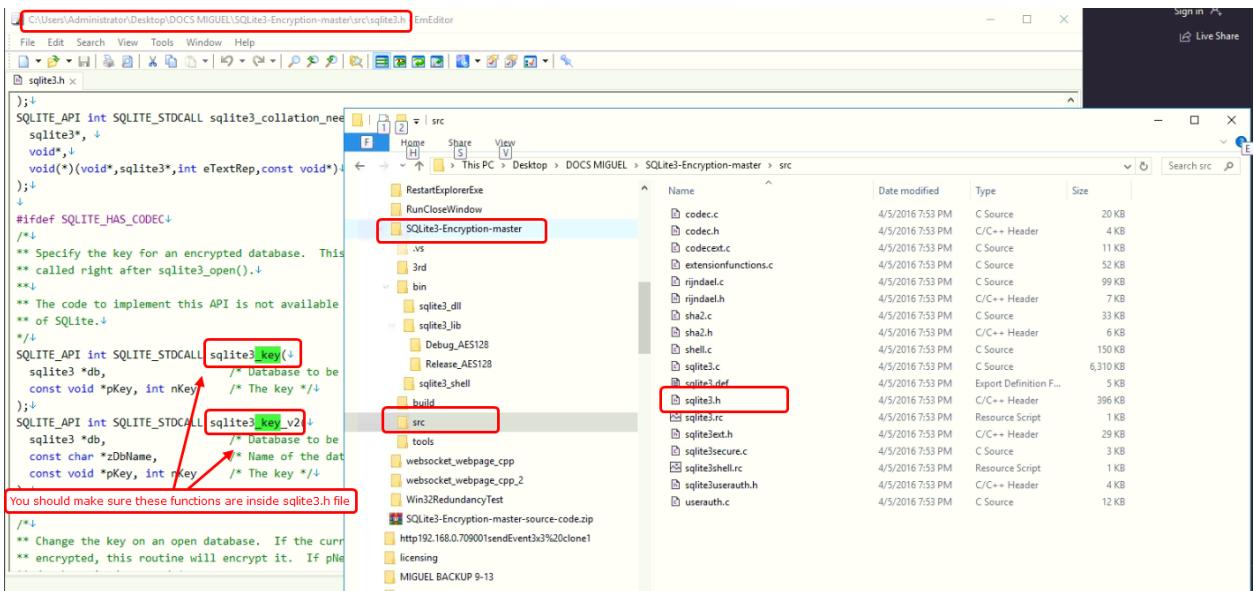


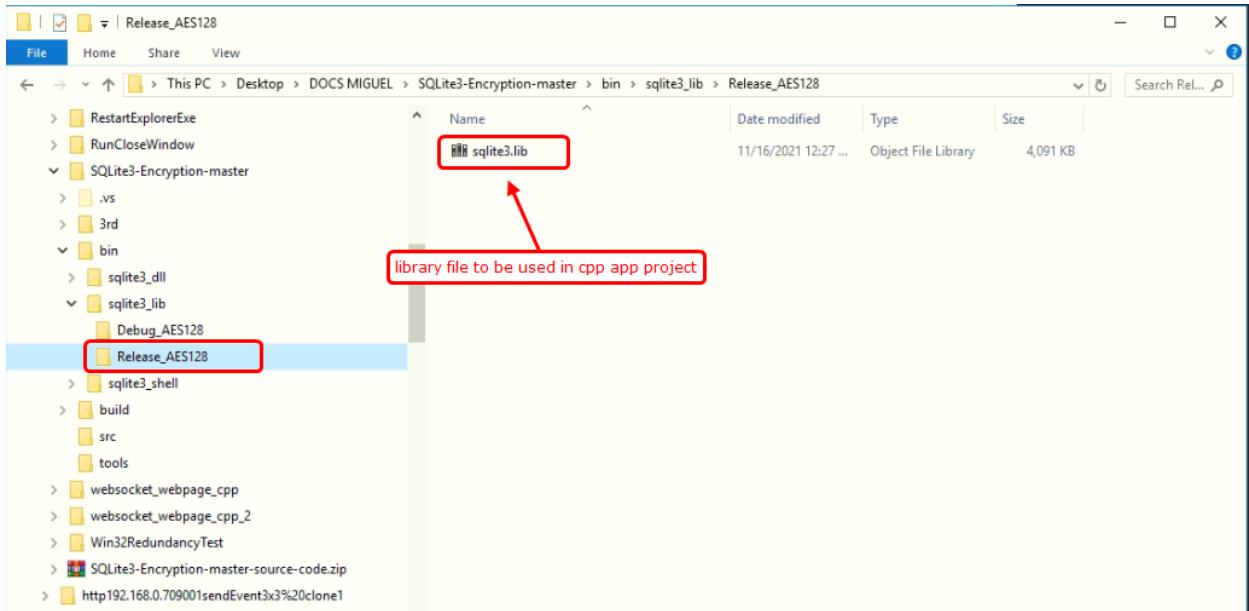
We choose “Release_AES128”. Please remember “Release_AES128”. AES128 is the encryption algorithm for sqlite db files with passwords.

Later when you make an sqlite db file, you should open it by AES128 encryption method.

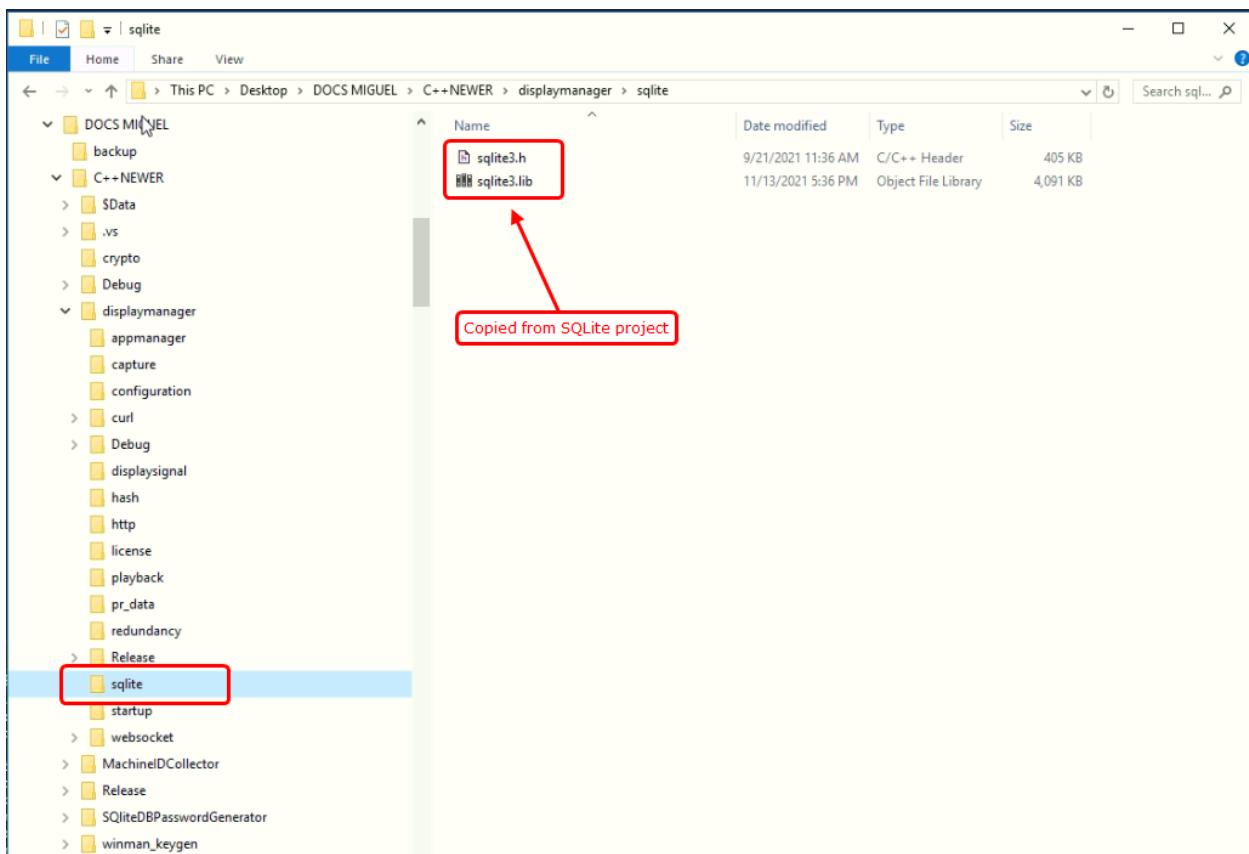
After rebuilding the projects, you can see the library files generated in the project folder.

To use this library in your cpp application, you have to copy header file(sqlite3.h) and library file(sqlite3.lib).





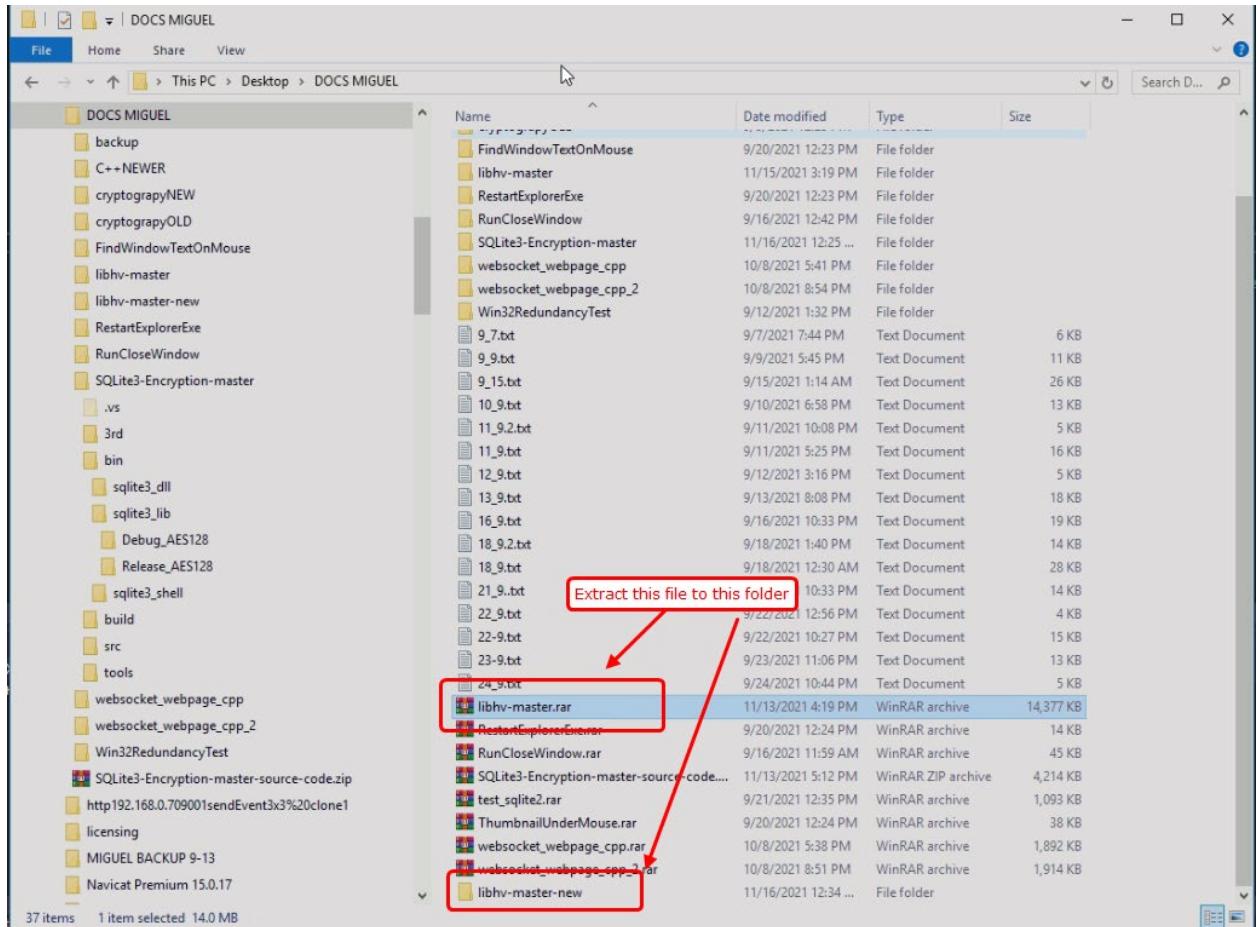
These two files are copied in the cpp project directory folder.



3) Websocket library

First our websocket library is an open-source project(libhv-master.rar).

Extract source code in some directory. Here we named “libhv-master-new”.



This project is not a normal Visual Studio project. It is called CMake project.

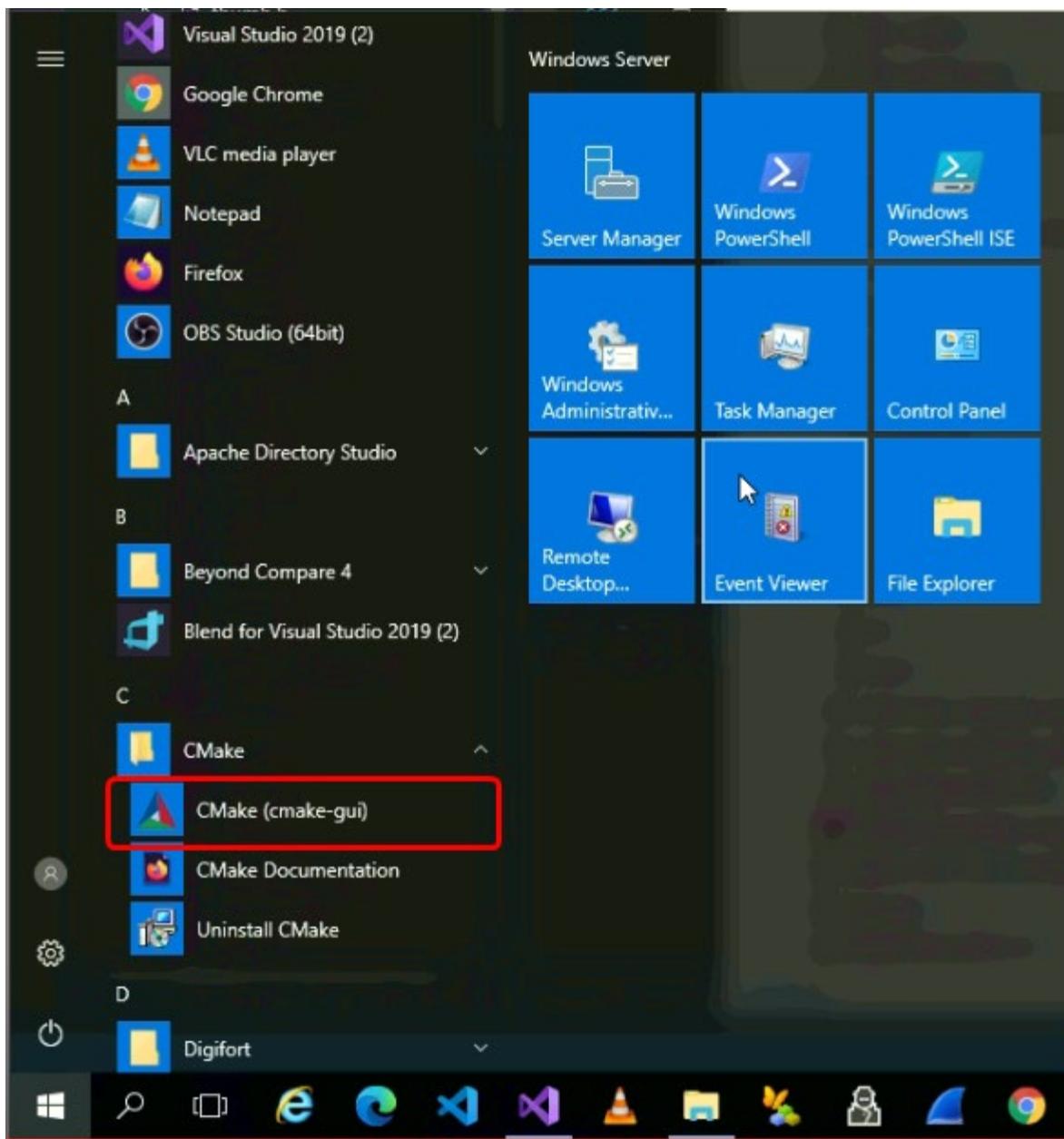
To built this project, you use CMake tool.

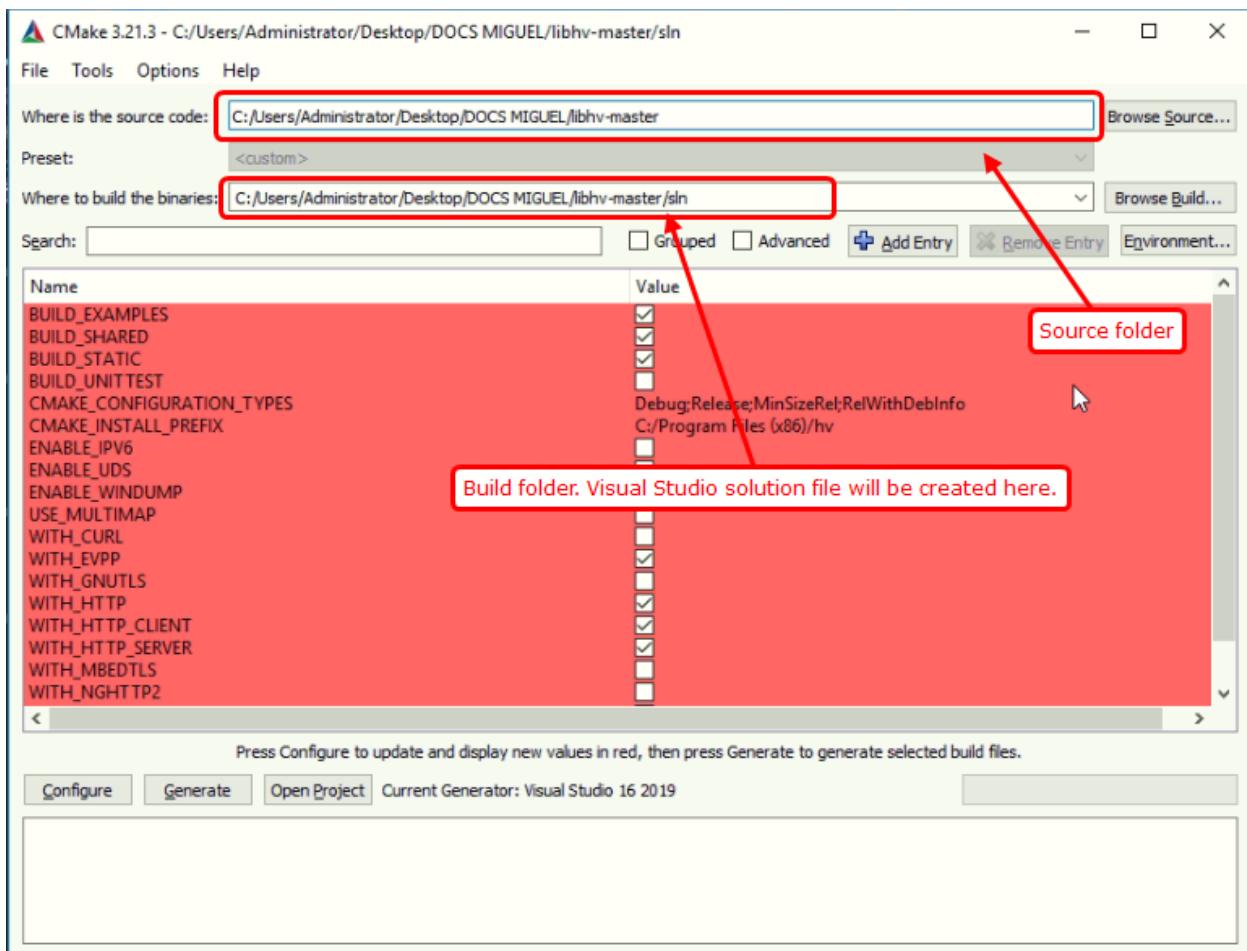
	Name	Date modified	Type	Size
DOCS MIGUEL	.github	9/29/2021 9:51 AM	File folder	
backup	base	9/29/2021 9:51 AM	File folder	
C++ NEWER	cert	9/29/2021 9:51 AM	File folder	
cryptographyNEW	cmake	9/29/2021 9:51 AM	File folder	
cryptographyOLD	cpputil	9/29/2021 9:51 AM	File folder	
FindWindowTextOnMouse	docs	9/29/2021 9:51 AM	File folder	
libhv-master	echo-servers	9/29/2021 9:51 AM	File folder	
libhv-master-new	etc	9/29/2021 9:51 AM	File folder	
libhv-master	event	9/29/2021 9:51 AM	File folder	
RestartExplorerExe	evpp	9/29/2021 9:51 AM	File folder	
RunCloseWindow	examples	9/29/2021 9:51 AM	File folder	
SQLCipher3-Encryption-master	html	9/29/2021 9:51 AM	File folder	
.vs	http	9/29/2021 9:51 AM	File folder	
3rd	misc	9/29/2021 9:51 AM	File folder	
bin	protocol	9/29/2021 9:51 AM	File folder	
sqlite3.dll	scripts	9/29/2021 9:51 AM	File folder	
sqlite3.lib	sln	10/5/2021 3:53 PM	File folder	
Debug_AES128	ssl	9/29/2021 9:51 AM	File folder	
Release_AES128	unittest	9/29/2021 9:51 AM	File folder	
sqlite3_shell	util	9/29/2021 9:51 AM	File folder	
build	.clang-format	9/29/2021 9:51 AM	CLANG-FORMAT ...	3 KB
src	.gitattributes	9/29/2021 9:51 AM	GITATTRIBUTES File	1 KB
tools	.gitignore	9/29/2021 9:51 AM	GITIGNORE File	1 KB
websocket_webpage.cpp	.travis.yml	9/29/2021 9:51 AM	YML File	1 KB
websocket_webpage.cpp_2	BUILD.md	9/29/2021 9:51 AM	MD File	3 KB
Win32RedundancyTest	CMakeLists.txt	9/29/2021 9:51 AM	Text Document	7 KB
SQLite3-Encryption-master-source-code.zip	config.ini	9/29/2021 9:51 AM	Configuration sett...	1 KB
http192.168.0.709001sendEvent3x3%20clone1	config.mk	9/29/2021 9:51 AM	Makefile	1 KB
licensing	configure	9/29/2021 9:51 AM	File	6 KB
MIGUEL BACKUP 9-13	getting_started.sh	9/29/2021 9:51 AM	SH File	3 KB
	hconfig.h	10/4/2021 3:28 PM	C/C++ Header	2 KB
	hconfig.h.in	9/29/2021 9:51 AM	IN File	2 KB
	hexport.h	9/29/2021 9:51 AM	C/C++ Header	3 KB

There is no solution file in the folder.

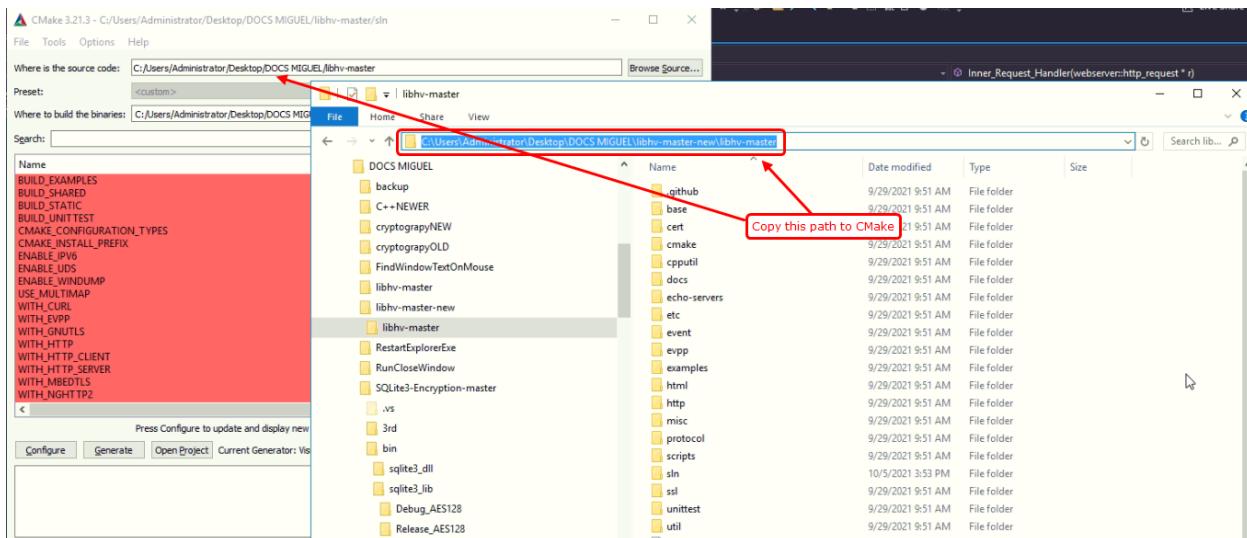
CMake tool will configure all for Visual Studio.

First run CMake tool.



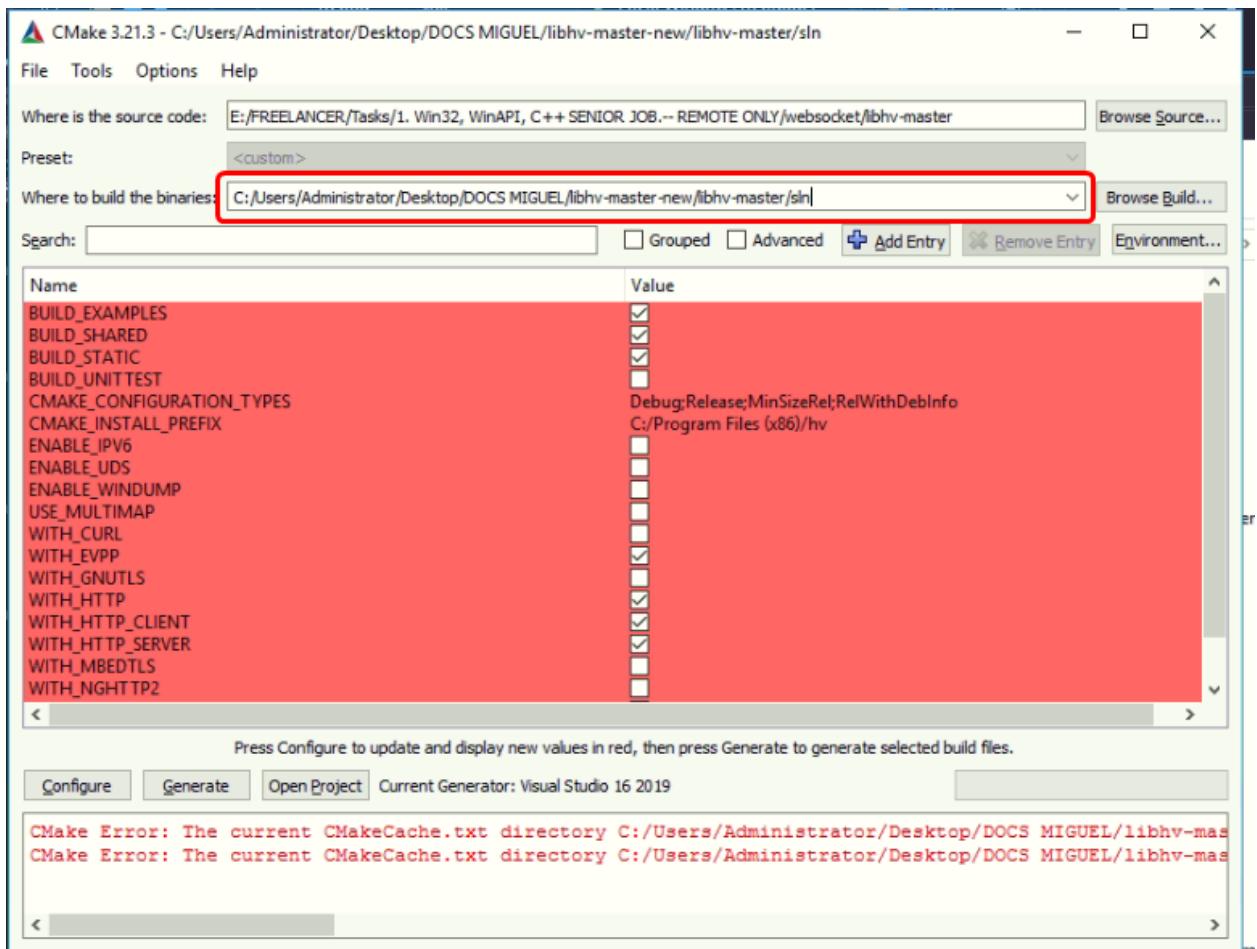


Input the CMake project folder to Souce code text box.

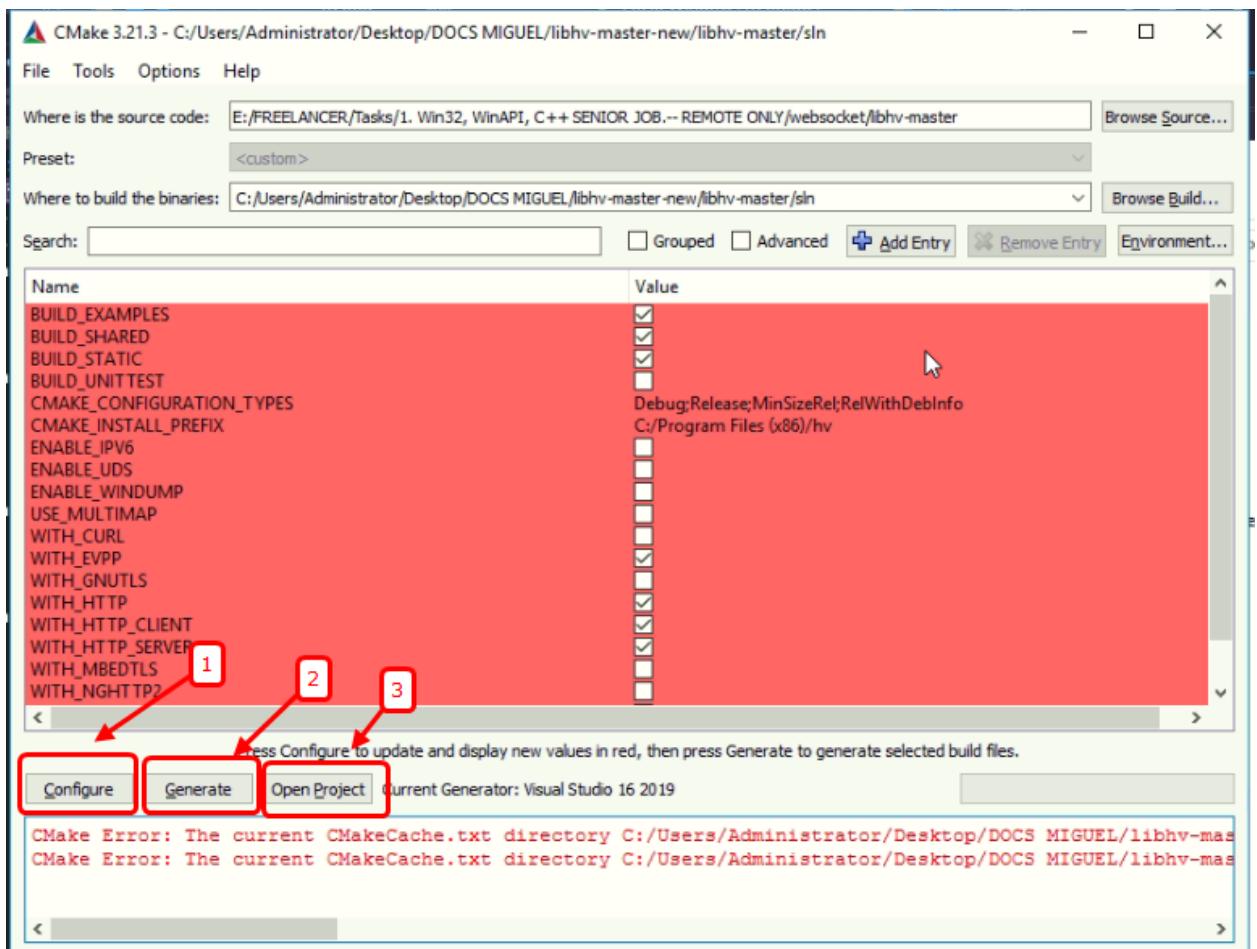


You can make build folder by inputting some directory. But here we use "sln" subdirectory in the project folder.

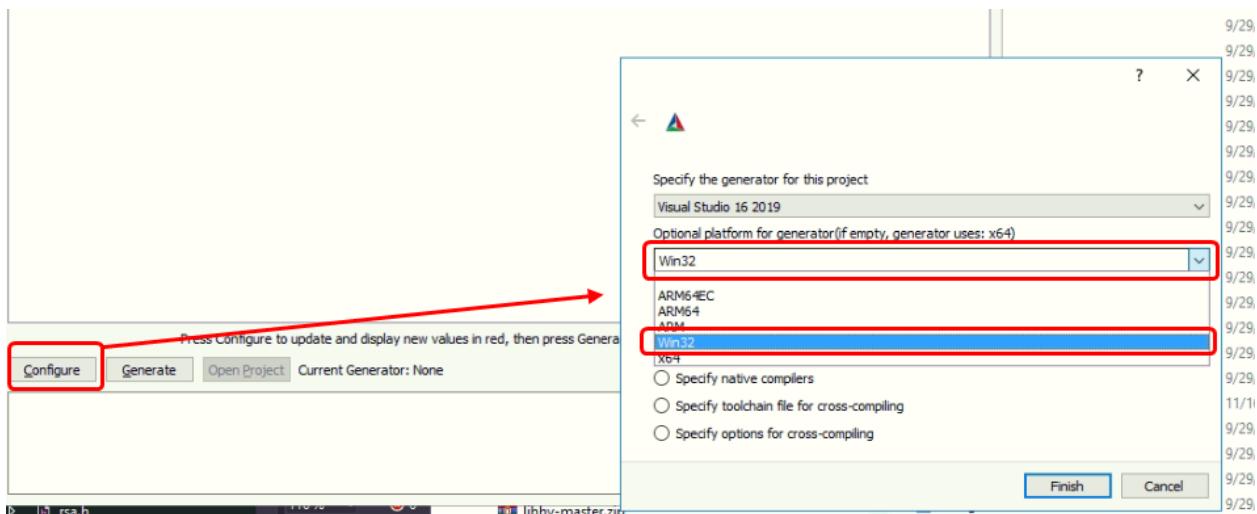
"sln" is not existing but will be created.



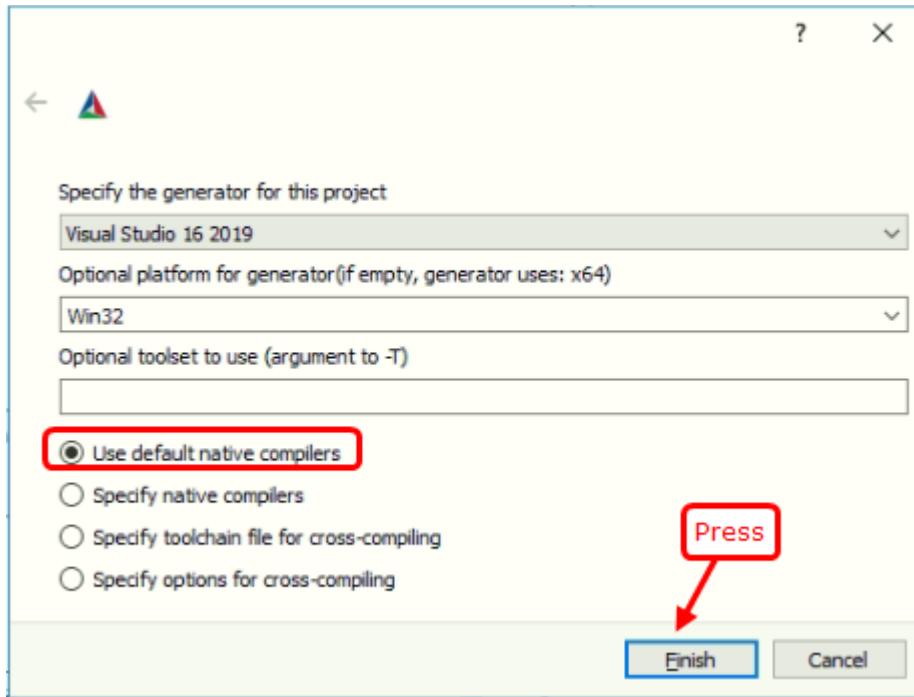
Press the buttons “Configure”, “Generate”, “Open Project” in order.



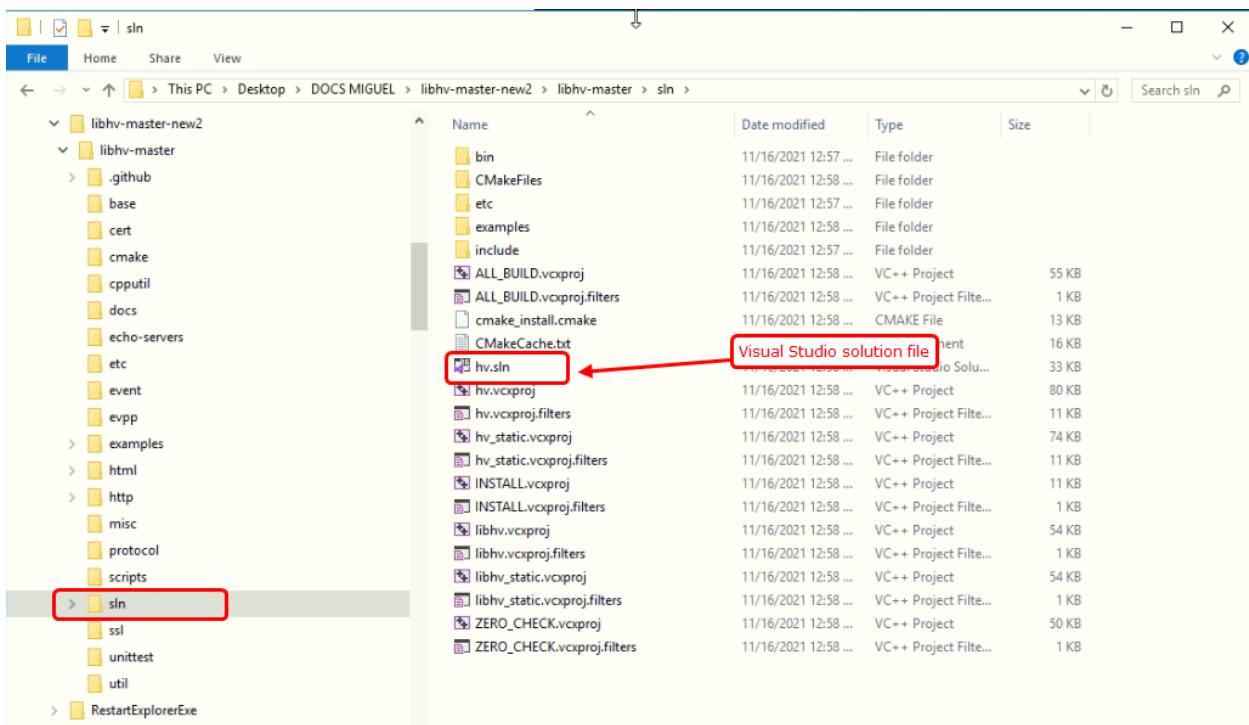
When you press configure button, it will show



And press "Finish".



You should check the “sln” directory is created.



You can open project by double-clicking hv.sln file.

At first, you build the project by setting “ALL_BUILD” project as “startup” project.

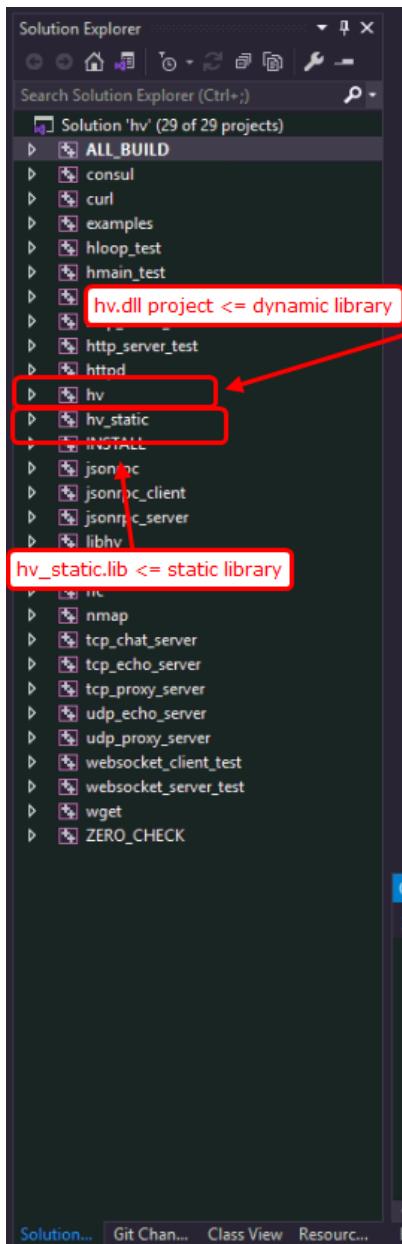
You can build this project in Debug and Release mode.

```

Output
Show output from: Build
26>----- Skipped Rebuild All: Project: libhv_static, Configuration: Debug Win32 -----
26>Project not selected to build for this solution configuration
24>Building Custom Rule C:\Users\Administrator\Desktop\DOCS MIGUEL\libhv-master-new2\libhv-master\examples\CMakeLists.txt
22>websocket_server_test.vcxproj -> C:\Users\Administrator\Desktop\DOCS MIGUEL\libhv-master-new2\libhv-master\sln\bin\Debug\websocket_server_test.exe
23>wget.vcxproj -> C:\Users\Administrator\Desktop\DOCS MIGUEL\libhv-master-new2\libhv-master\sln\bin\Debug\wget.exe
27>----- Rebuild All started: Project: ALL_BUILD, Configuration: Debug Win32 -----
28>----- Skipped Rebuild All: Project: examples, Configuration: Debug Win32 -----
29>Project not selected to build for this solution configuration
27>Building Custom Rule C:\Users\Administrator\Desktop\DOCS MIGUEL\libhv-master-new2\libhv-master\CMakeLists.txt
29>----- Skipped Rebuild All: Project: INSTALL, Configuration: Debug Win32 -----
29>Project not selected to build for this solution configuration
----- Rebuild All: 25 succeeded, 0 failed, 4 skipped -----

```

It is good with no errors while building.



You can see the multiple projects in a solution. But the main projects under our consideration are two projects, hv and hv_static.

“hv” project will make hv.dll and hv.lib files and those will be linked to our cpp app project.

If we use this dynamic library, websocket core features run in hv.dll so cpp app does not have websocket code. Cpp app only links whenever websocket code is need while running.

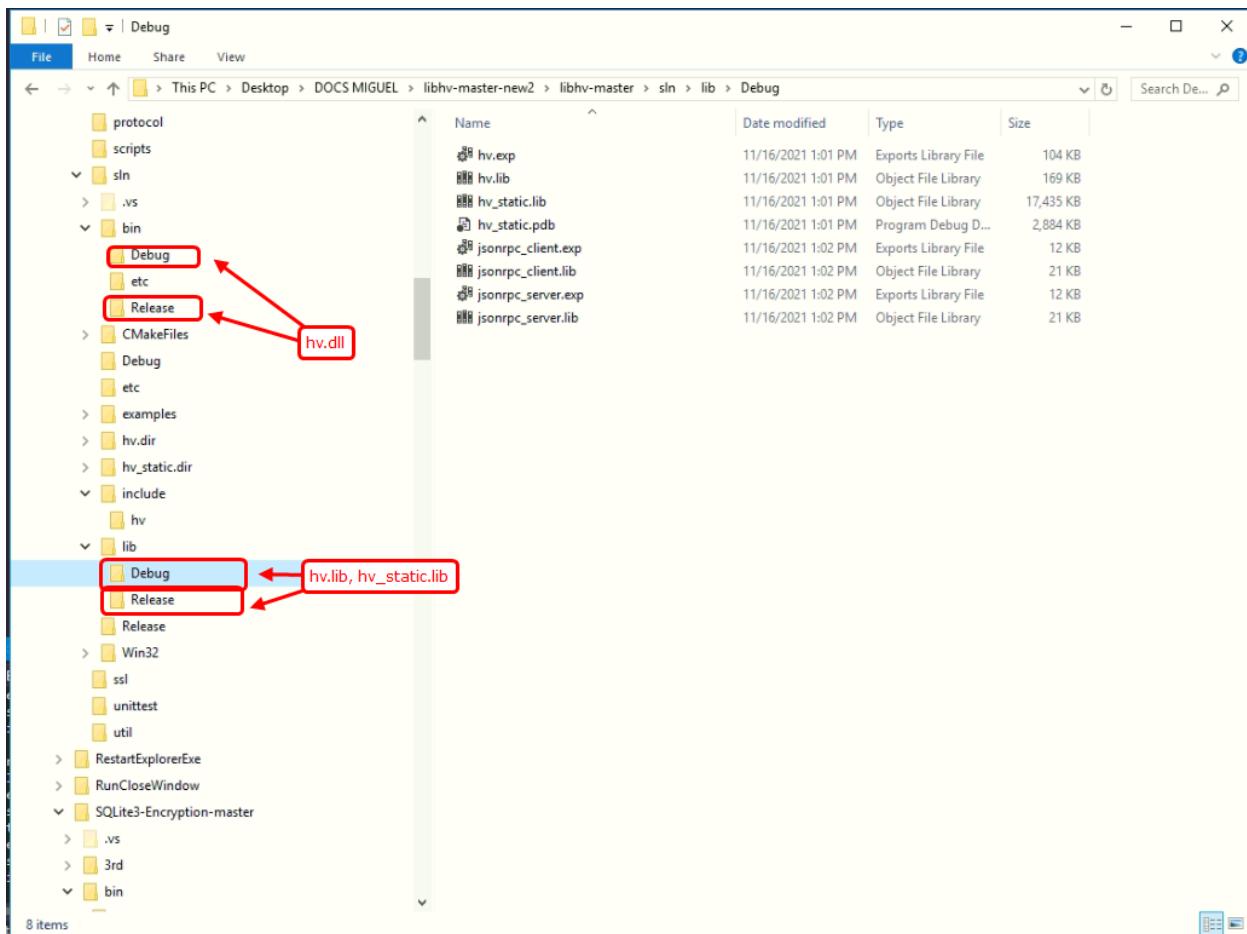
“hv_static” project will make hv_static.lib file and it will be linked to our cpp app project.

If we use this static library, websocket core features will be embedded into cpp app executive so that cpp app will run websocket code in itself.

So If developer directs more security against cracking, static library is highly recommended.

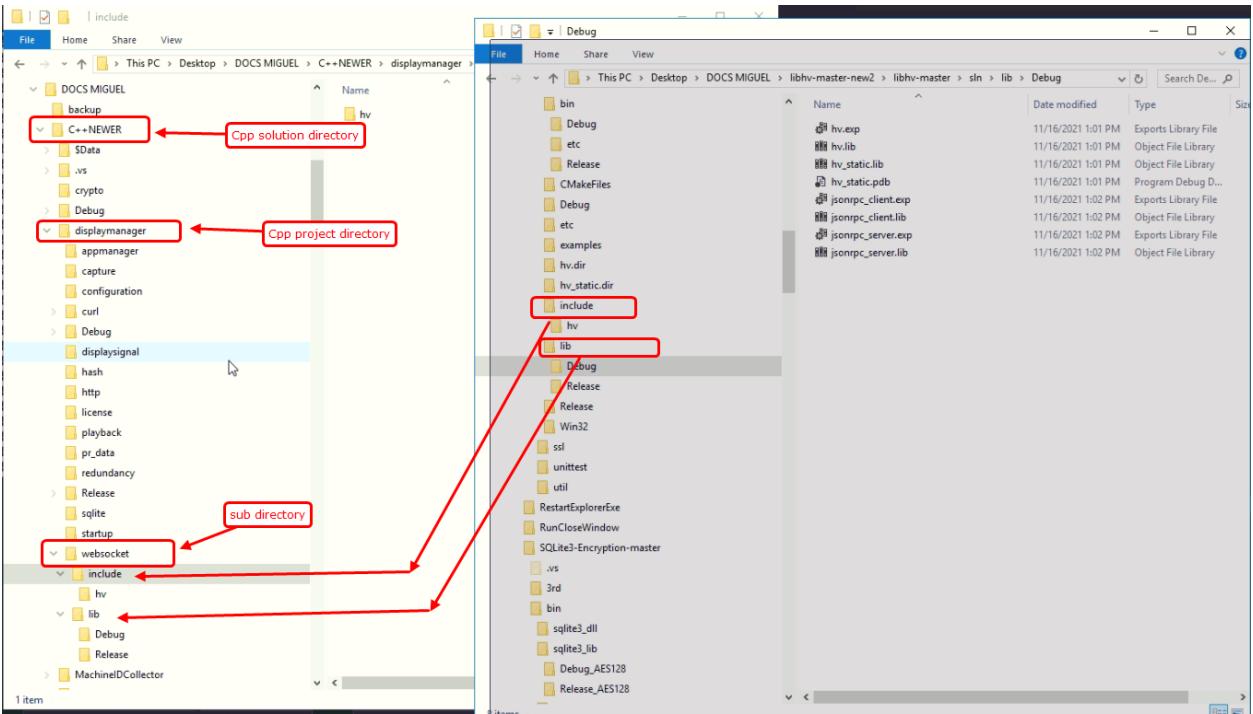
“hv.dll” file is generated in “sln/bin/Debug” or “sln/bin/Release” folder.

“hv.lib” and “hv_static.lib” files are generated in “sln/lib/Debug” or “sln/lib/Release” folder.

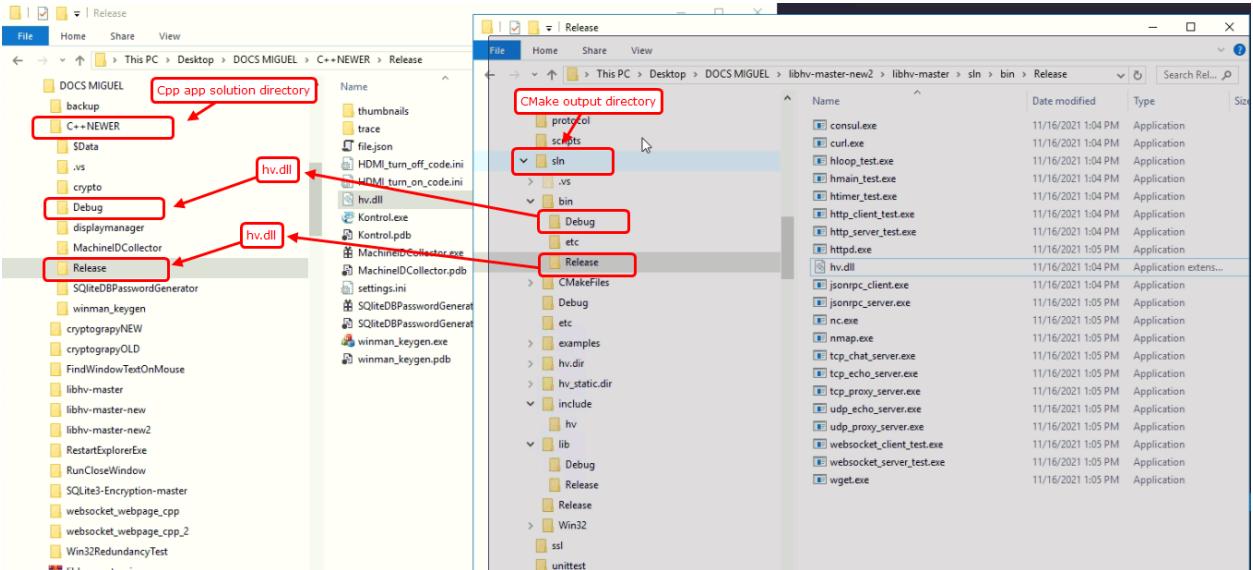


To use this websocket library in our cpp project, we need to copy some files into cpp project directory.

Header files("sln/include") and library files("sln/lib") are copied to the cpp project directory.



"hv.dll" in Debug and Release mode will be copied into the same directory as cpp application.



Header files are used for the prototypes of the functions to be used while compiling.

Library files are used for the linkage to the functions in a dll file while building.

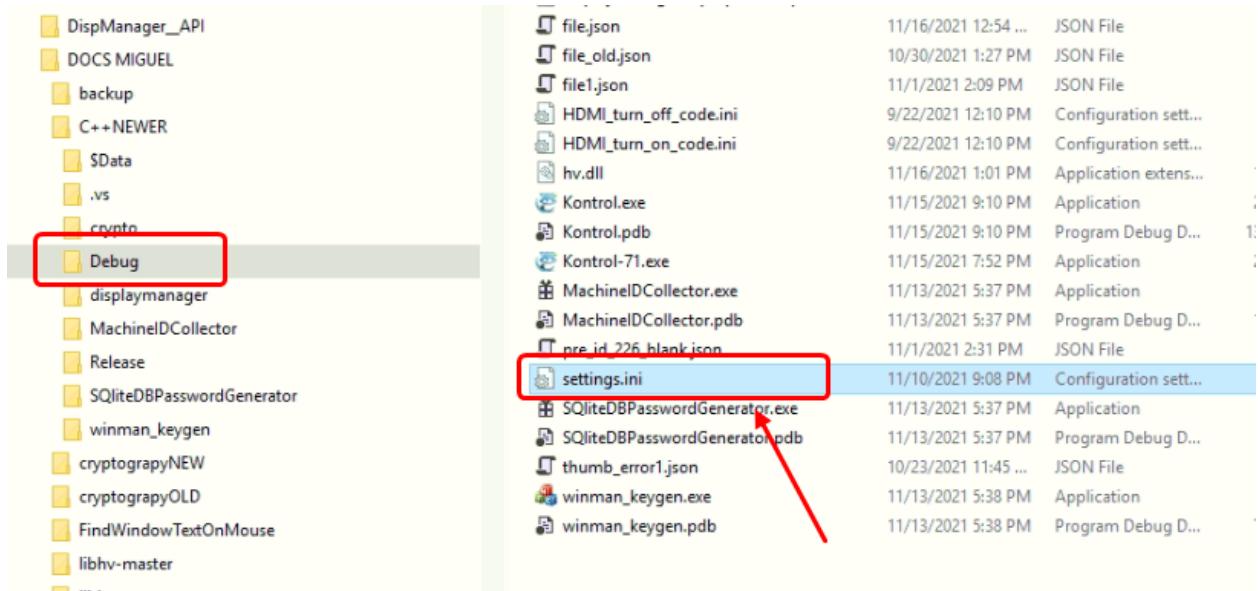
Dll files are used as a websocket running module for cpp app at runtime.

2. Configuration

1) Settings.ini file

In the running directory of cpp application, you can see the file "settings.ini".

This is the configuration file for running cpp app.



The content of this file has the several meanings.

```
settings.ini - Notepad
File Edit Format View Help
[REDACTED]
#websocket port number for move/resize
ws_port_move_resize:6001 [comment mark]

#websocket port number for thumbnail images
ws_port_thumbnail:6002 [blank line]
# period to send thumbnail images to angular webpage in milliseconds
thumbnail_period:70

#mate:192.168.0.71
```

The first line is like

```
webserver:192.168.0.71:5000
```

The line which starts with “webserver” shows IP address for requesting presentation data. In this case, server IP is 192.168.0.71 and its port number is 5000.

The second line starts with a character '#';

The lines which start with this character are considered comments, i.e. it is not used in cpp app.

So a user can add as many comment lines as he wants.

Third line is

```
ws_port_move_resize:6001
```

The line which starts with “ws_port_move_resize:” is considered as websocket server port for move/resize.

In this case, port number is 6001.

Fourth line is a blank line. (no characters, even ‘space’)

This line is not used. So we can use this blank line to well-looking configuration.

Sixth line is

```
ws_port_thumbnail:6002
```

The line which starts with “ws_port_thumbnail:” is considered as websocket server port for thumbnail.

In this case, port number is 6002.

Ninth line is

```
thumbnail_period:70
```

The line which starts with “thumbnail_period:” is considered as period in milliseconds of making thumbnail images.

In this case, thumbnail period is 70 milliseconds.

Cpp app makes thumbnail images every 70 milliseconds. If it takes more than 70 milliseconds to capture thumbnail images at a time, Cpp will make thumbnail images continuously.

The last line is

```
#mate:192.168.0.71
```

This line is commented. If we remove a character '#' from this line, it will work for redundancy.

The line which starts with “mate:” is considered as IP address of mate computer for redundancy.

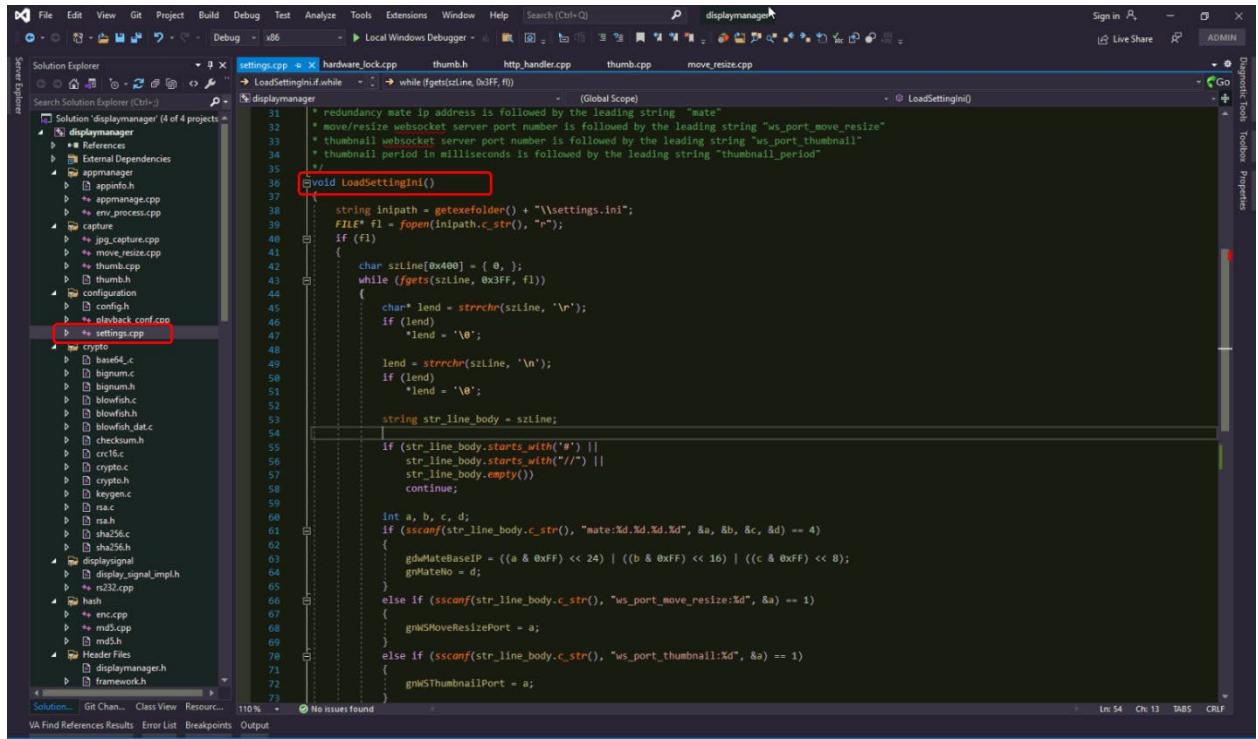
Let's assume that 192.168.0.71 exists and cpp app runs at it.

If we run cpp app at this computer, it will work as a slave, so that it will not play any presentation.

When cpp app at 192.168.0.71 goes faulty, or crashes, or closed, then cpp app at this computer will be a master and play the same presentation as the last one at 192.168.0.71.

If 192.168.0.71 does not exist or no cpp app runs at it, cpp app at this computer runs as a master.

If this line is commented, no redundancy is enabled for cpp app so that it only runs as a master.



```
string inipath = getexefolder() + "\\settings.ini";
FILE* f1 = fopen(inipath.c_str(), "r");
if (f1)
{
    char szline[0x400] = { 0 };
    while (fgets(szLine, 0x3FF, f1))
    {
        char* lend = strchr(szLine, '\n');
        if (lend)
            *lend = '\0';

        lend = strchr(szLine, '\r');
        if (lend)
            *lend = '\0';

        string str_line_body = szLine;

        if (str_line_body.starts_with('#') ||
            str_line_body.starts_with("//") ||
            str_line_body.empty())
            continue;

        int a, b, c, d;
        if (sscanf(str_line_body.c_str(), "mate:%d.%d.%d.%d", &a, &b, &c, &d) == 4)
        {
            gdwMateBaseIP = ((a & 0xFF) << 24) | ((b & 0xFF) << 16) | ((c & 0xFF) << 8);
            gmMateIo = d;
        }
        else if (sscanf(str_line_body.c_str(), "ws_port_move_resize:%d", &a) == 1)
        {
            gmWSMoveResizePort = a;
        }
        else if (sscanf(str_line_body.c_str(), "ws_port_thumbnail:%d", &a) == 1)
        {
            gmWSThumbnailPort = a;
        }
    }
}
```

You will find the function “LoadSettingIni()”

- *How to change the settings.ini file name or path.*

You can find the following line in LoadSettingIni() function.

```
string inipath = getexefolder() + "\\settings.ini";
```

Here “inipath” variable represents the path of configuration file.

getexefolder() returns the folder path of cpp app.

“inipath” will point to the path of the settings.ini file located in the directory of cpp app.

You can change the path by changing “inipath” variable value.

- *How to add comments in settings.ini file*

Settings.ini file can have the comments by line.

As explained above, ‘#’ is the leading character of the comment line.

In LoadSettingIni() function, you can see the code like this.

```
if (str_line_body.starts_with('#') ||
    str_line_body.starts_with("//") ||
    str_line_body.empty())
    continue;
```

As you can see here, comment line starts with a character ‘#’ or a string “//” like C comment lines.

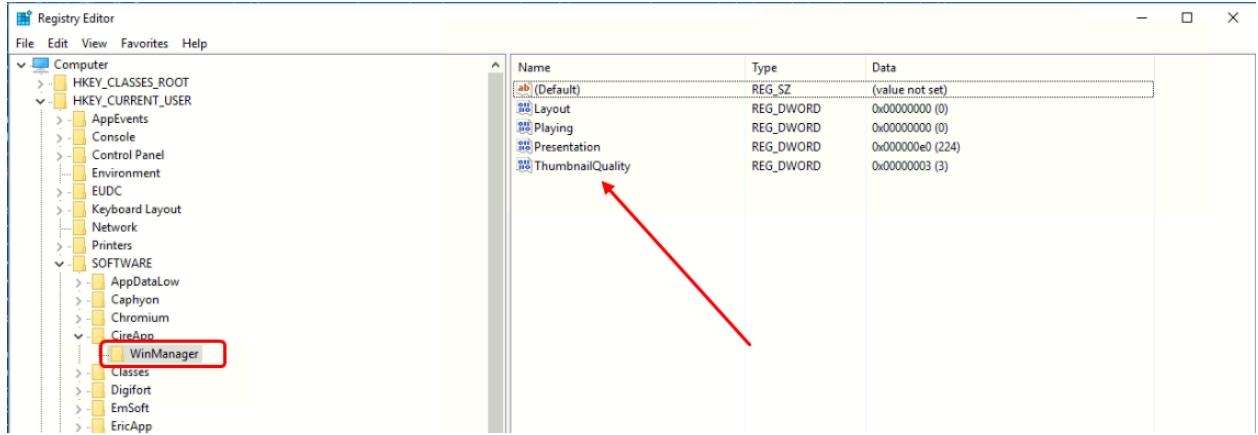
But there is no way to comment sub-string in the line like C commenting `/**/`.

If you would like to add more symbols to represent comment lines, change the code above as you want.

2) Registry

Cpp app uses registry items for saving the last playing context.

HKEY_CURRENT_USER\Software\CireApp\WinManager



This key(it is called “Registry Key”) contains 4 values.

Playing	Flag that represents the presentation was playing before closing app. It is updated whenever you play a presentation, or stop presentation.
Presentation	Presentation id which is played by cpp app. It is updated whenever you play a presentation.
Layout	Layout index which is played by cpp app. It is updated whenever you play a presentation or going to the next layout automatically.
ThumbnailQuality	It is one of the following numbers. 1, 2, 3 It is updated when cpp app receives http command from webpage.

- *When is it loaded from registry?*

Loading registry values are proceeded in `GetLastLayout()` function in “playback_conf.cpp”.

```

1 //include "../framework.h"
2 #include "config.h"
3 #include "../playback/status.h"
4 #include "../capture/thumb.h"
5
6 /*
7 * At first, opens a registry key "HKEY_CURRENT_USER\Software\CireApp\WinManager"
8 * Gets the value "Presentation", "Layout" as an unsigned integer.
9 * If either of them exceeds maximum value, then initialize those values and stops presentation running.
10 * Otherwise, resume presentation with those indices.
11 */
12 bool GetLastLayout()
13 {
14     HKEY hKey = NULL;
15     LONG lResult = 0;
16     BOOL fSuccess = TRUE;
17     DWORD dwSize(sizeof(DWORD));
18
19     int nPresentationId;
20     int nLayout;
21     int nPlaying;
22     int nThumbnailQuality = 3;

```

You select this function name and press “Shift + Alt + F” key.

Then you will see the result of references in the project.

```

10     * Otherwise, resume presentation with those Indices.
11 */
12 bool GetLastLayout()
13 {
14     HKEY hKey = NULL;
15     LONG lResult = 0;
16     BOOL fSuccess = TRUE;
17     DWORD dwSize(sizeof(DWORD));
18
19     int nPresentationId;
20     int nLayout;
21     int nPlaying;
22     int nThumbnailQuality = 3;
23
24     fSuccess = RegCreateKeyExW(HKEY_CURRENT_USER, L"Software\CireApp\WinManager", 0, NULL, 0, (KEY_WRITE | KEY_READ), NULL, &hKey, NULL);
25
26     if (fSuccess)
27     {
28         // HKEY_CURRENT_USER\Software\CireApp\WinManager key contains 4 values.
29         // Presentation = 214 <- this is the last played presentation id configured in webpage.
30         // This presentation should exist in file.json file.
31         fSuccess = RegQueryValueExW(hKey, L"Presentation", 0, NULL, (BYTE*)&nPresentationId, &dwSize);
32         // Layout = 0 <- this is the last played layout index in a presentation
33         fSuccess = RegQueryValueExW(hKey, L"Layout", 0, NULL, (BYTE*)&nLayout, &dwSize);
34         // Playing = 0 (or 1), before closing cpp app, some presentation was playing if this value is 1, not playing otherwise.
35         // If you set this value to be 0 and you run cpp app, then cpp does not play the presentation at startup.
36         // But you see this value to be 1 and you run cpp app, then cpp plays the presentation at startup.
37         fSuccess = RegQueryValueExW(hKey, L"Playing", 0, NULL, (BYTE*)&nPlaying, &dwSize);
38

```

VA Find References Results

- Found 3 (+1 hidden) references in 3 files to GetLastLayout (searched project)
- C:\Users\Administrator\Desktop\DOCS MIGUEL C+++NEWER\displaymanager\displaymanager.vcxproj
 - 12: bool GetLastLayout()
 - 13: h
 - 14: (4): bool GetLastLayout();
 - 15: +4: C:\Users\Administrator\Desktop\DOCS MIGUEL C+++NEWER\displaymanager\displaymanager\configuration\playback.conf.h
 - 16: WinMain (277): GetLastLayout();

As you can see, WinMain() function calls this function.

It is called at startup of cpp application. It is called just once.

So we may change the registry value. If we run cpp app after changing, cpp app will read registry items.

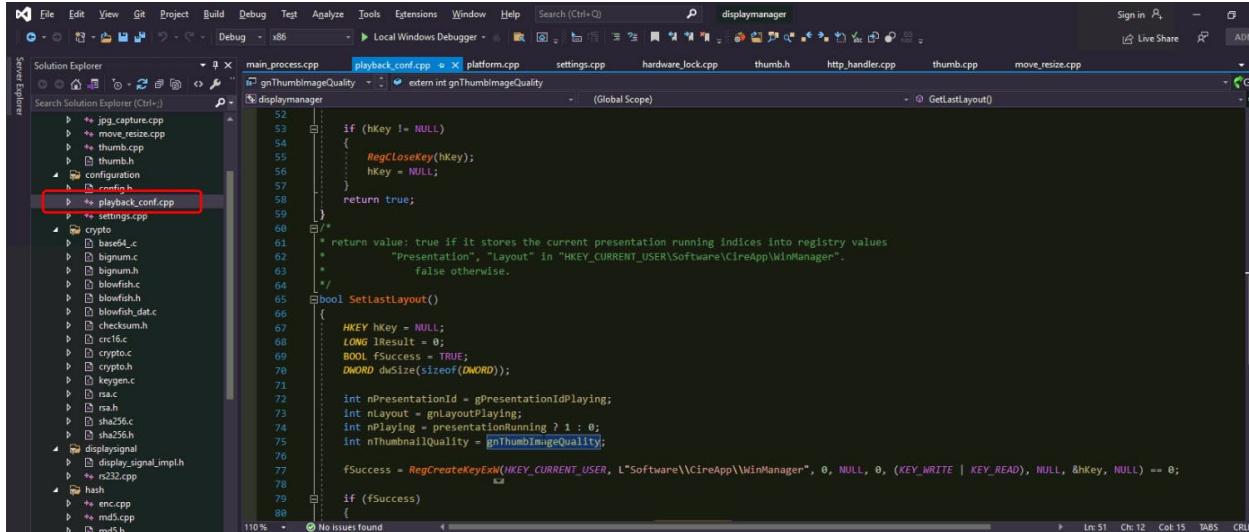
Variables corresponding to registry values are as follows.

Registry value	variable	Initial value	File name
Playing	BOOL presentationRunning	FALSE	main_process.cpp

Layout	int gnLayoutPlaying	0	main_process.cpp
Presentation	int gPresentationPlaying	0	main_process.cpp
ThumbnailQuality	int gnThumbnailQuality	3	thumb.cpp

- When is it saved?

In playback_conf.cpp file, SetLastLayout() function saves the context of playing presentation.



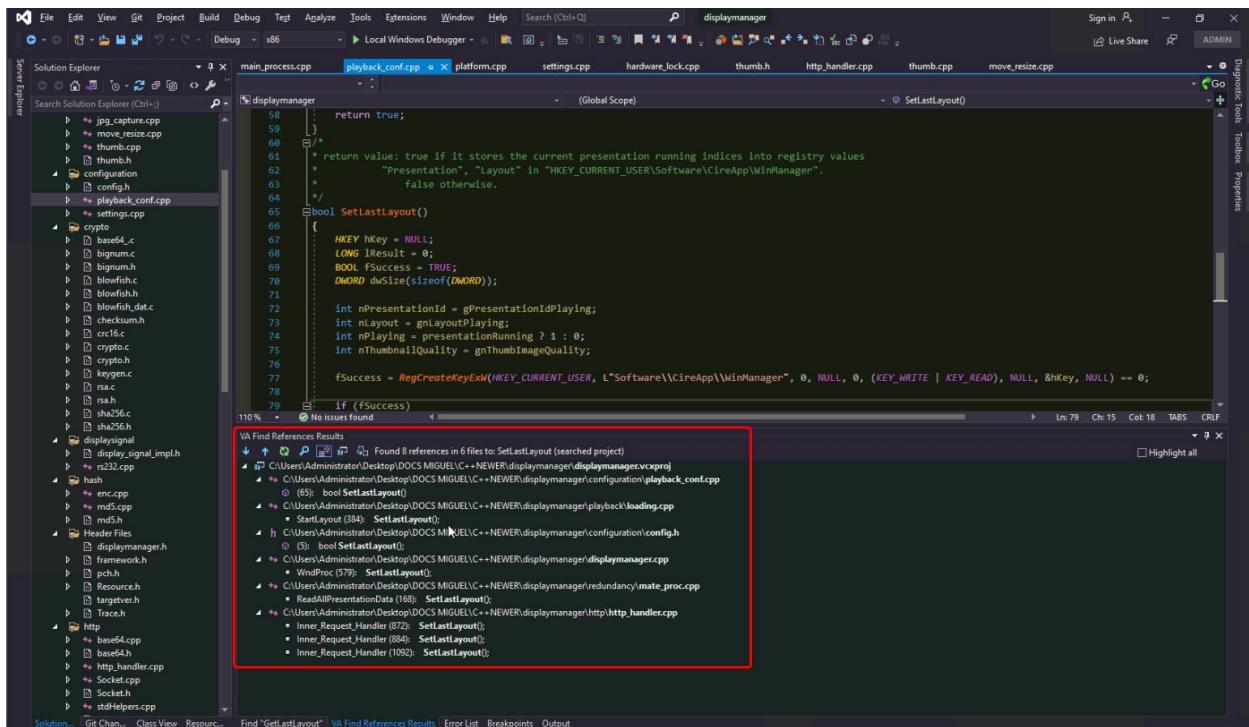
```

52
53     if (hKey != NULL)
54     {
55         RegCloseKey(hKey);
56         hKey = NULL;
57     }
58     return true;
59 }
60 */
61 /* return value: true if it stores the current presentation running indices into registry values
62 * "Presentation", "Layout" in "HKEY_CURRENT_USER\Software\CireApp\WinManager".
63 * false otherwise.
64 */
65 bool SetLastLayout()
66 {
67     HKEY hKey = NULL;
68     LONG lResult = 0;
69     BOOL fSuccess = TRUE;
70     DWORD dwSize(sizeof(DWORD));
71
72     int nPresentationId = gPresentationIdPlaying;
73     int nLayout = gnLayoutPlaying;
74     int nPlaying = presentationRunning ? 1 : 0;
75     int nThumbnailQuality = gnThumbnailQuality;
76
77     fSuccess = RegCreateKeyEx(HKEY_CURRENT_USER, L"Software\CireApp\WinManager", 0, NULL, 0, (KEY_WRITE | KEY_READ), NULL, &hKey, NULL) == 0;
78
79     if (fSuccess)
80     {
81         if (RegSetValueEx(hKey, L"Presentation", 0, REG_EXPAND_SZ, (const BYTE*)&nPresentationId, sizeof(nPresentationId)) != 0)
82             fSuccess = FALSE;
83         if (RegSetValueEx(hKey, L"Layout", 0, REG_EXPAND_SZ, (const BYTE*)&nLayout, sizeof(nLayout)) != 0)
84             fSuccess = FALSE;
85         if (RegSetValueEx(hKey, L"Playing", 0, REG_EXPAND_SZ, (const BYTE*)&nPlaying, sizeof(nPlaying)) != 0)
86             fSuccess = FALSE;
87         if (RegSetValueEx(hKey, L"ThumbnailQuality", 0, REG_DWORD, (const BYTE*)&nThumbnailQuality, sizeof(nThumbnailQuality)) != 0)
88             fSuccess = FALSE;
89     }
90 }

```

You select the function name in editor, and press “Shift + Alt + F” key.

The result is shown at the bottom of the window.



VA Find References Results

- Found 8 references in 6 files to: SetLastLayout (searched project)
- C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\displaymanager.vcxproj
 - ..\C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\configuration\playback_conf.cpp
 - (55): bool SetLastLayout()
 - ..\C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\loading.cpp
 - StartLayout (384): SetLastLayout();
 - ..\C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\configuration\config.h
 - (5): bool SetLastLayout();
 - ..\C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\displaymanager.cpp
 - WindProc (579): SetLastLayout();
 - ..\C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\redundancy\mate_proc.cpp
 - ReadAllPresentationData (18): SetLastLayout();
 - ..\C:\Users\Administrator\Desktop\DOCS MIGUEL C++\NEVER\displaymanager\http\http_handler.cpp
 - Inner_Request_Handler (872): SetLastLayout();
 - Inner_Request_Handler (884): SetLastLayout();
 - Inner_Request_Handler (1050): SetLastLayout();

Calling this function are the following cases.

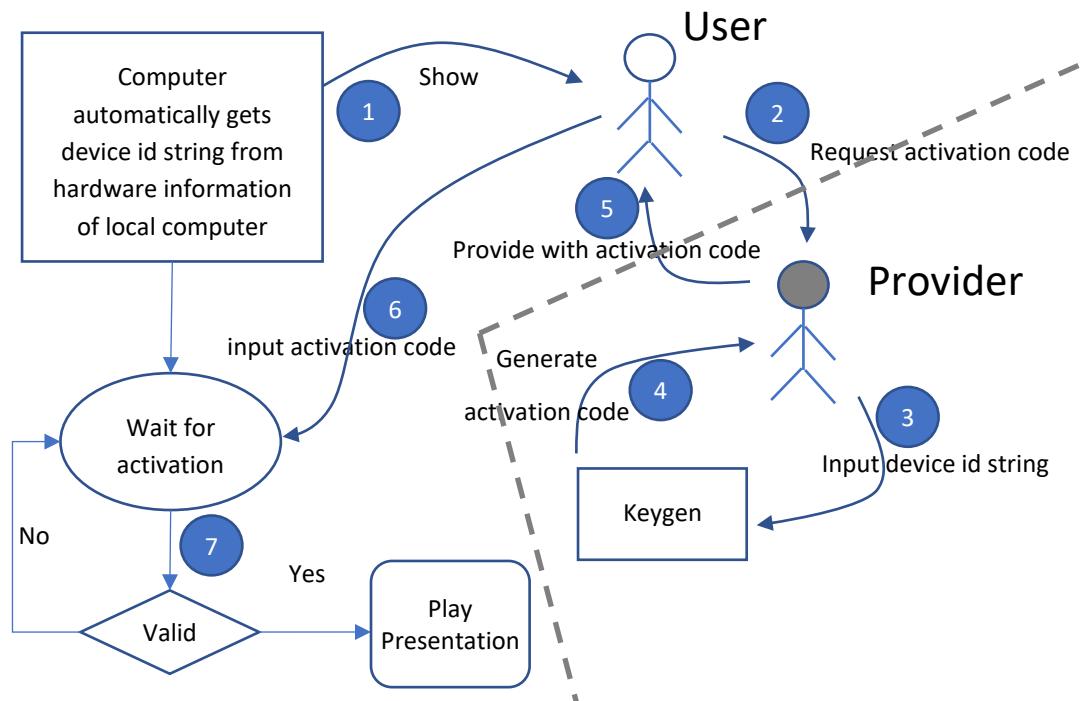
Function	File	Description
StartLayout	loading.cpp	When a user plays a presentation on webpage or it automatically goes to the next layout or rewind.
WndProc	displaymanager.cpp	When the main window is closed.
ReadAllPresentationData	mate_proc.cpp	When cpp app runs as a slave and receives presentation data and context from its master(mate).
Inner_Request_Handler	http_handler.cpp	When a user presses “close-app” button on webpage. When a user presses “shutdown” button on webpage. When a user changes thumbnail quality on webpage.

3. Licensing

1) Cryptography

We use cryptography for licensing.

Licensing will work as the following.



Keygen is private program of provider. It can make activation code for user's device id string.

Keygen application(3 => 4) uses CRC16, SHA256, RSA2048 encryption(private key), blowfish encode, base64 encode.

Cpp application(6 => 7) verifies activation code by base64 decode, blowfish decode, RSA2048 decryption(public key), SHA256, CRC16.

For more information, please see "Kontrol Licensing Manual.pdf".

2) Hash

We use hash algorithms for two features.

Getting device id string(MD5)

Generating and verifying activation code.(SHA256)

MD5 code is an open source.

It is located in "crypto" folder. md5.h, md5.cpp.

SHA256 code is also an open source which can be found in sha256.c and sha256.h in "crypto" folder.

```
hardware_lock.cpp MachineIDCollector.cpp check.cpp move_resize.cpp
displaymanager
├── displaymanager
│   ├── External Dependencies
|   |   └── appmanager
|   |       ├── appinfo.h
|   |       ├── appmanager.cpp
|   |       ├── env_process.cpp
|   |       ├── capture
|   |       |   ├── jpg_capture.cpp
|   |       |   ├── move_resize.cpp
|   |       |   ├── move_resize.h
|   |       |   └── thumb
|   |       ├── configuration
|   |       |   ├── config.h
|   |       |   ├── playback.conf.cpp
|   |       |   ├── settings.cpp
|   |       └── crypto
|   |           ├── base64.c
|   |           ├── bignum.c
|   |           ├── bignum.h
|   |           ├── blowfish.c
|   |           ├── blowfish.h
|   |           ├── blowfish_dat.c
|   |           ├── checksum.h
|   |           ├── crc16.c
|   |           ├── crypto.c
|   |           ├── crypto.h
|   |           └── keygen.c
|   |               ├── md5.cpp
|   |               └── md5.h
|   |               └── sha256.c
|   |                   └── sha256.h
|   └── displaymanager
|       ├── displaymanager.cpp
|       ├── displaymanager.h
|       ├── signal_impl.h
|       └── n232.cpp
|   └── Header Files
|       ├── displaymanager.h
|       ├── framework.h
|       ├── pch.h
|       ├── Resource.h
|       └── targetver.h
└── MachineIDCollector.cpp
    └── check.cpp
    └── move_resize.cpp

displaymanager
└── CheckHardwareLock()
    └── hardware_lock.cpp
```

hardware_lock.cpp

```
1 //include "../../../framework.h"
2 //include "license.h"
3
4 extern char gszDeviceKey[100]; // MD5 hash string to uniquely identify the local machine.
5
6 /*
7 * Check the physical device components
8 * return values:
9 *  0: Information of 5 physical device components is equal to an appropriate one below.
10 *  1: Hard disk serial number is not equal to "WJB08LP2"
11 *  2: MAC address of the network adapter attached to local machine is not equal to "B8:CB:29:94:F3:80".
12 *  3: CPU id of the local machine is not equal to "GenuineIntel"
13 *  4: Windows Product Id is not equal to "00376-30715-28674-AA103".
14 *  5: Computer name of Windows system is not equal to "WIN-P2E307NBH7R".
15 */
16 bool CheckHardwareLock()
17 {
18     string devid;
19     // get the serial number of the hard disk attached
20     // hardware lock
21     if(1)
22         // ... computer
23
24     ShowOutputFrom: Build
1>hidden_file.cpp
1>license_db.cpp
1>main.cpp
1>main_process.cpp
1>md5.h
1>rsa.cpp
1>rsa.h
1>sha256.cpp
1>sha256.h
1>sqlite3.lib(sqlite3secure.obj) : MSIL .netmodule or module compiled with /GL found; restarting link with /LTCG; add /LTCG to the link command line to improve linker performance
1>LINK : warning LNK4098: defaultlib 'MSVCRT' conflicts with use of other libs; use /NODEFAULTLIB:library
1>MSVCRTD.lib(initializers.obj) : warning LNK4098: defaultlib 'libcmt.lib' conflicts with use of other libs; use /NODEFAULTLIB:library
1>MSVCRTD.lib(msvcrt.dll) : warning LNK4098: defaultlib 'msvcrt.dll' conflicts with use of other libs; use /NODEFAULTLIB:library
1>linked executing code
1>displaymanager.vcxproj -> C:\Users\Administrator\Desktop\DOCS MIGUEL\C+++NEWER\Release\Kontrol.exe
1>Done building project "displaymanager.vcxproj".
***** Rebuild All: 4 succeeded, 0 failed, 0 skipped *****
```

- How to change generation of device id string

In license folder, there is a file named “check.cpp”.

In this file, you can find GetDeviceHashString() function.

This function generates device id string by hard disk serial number, windows product id, mac address, Computer name, cpu id using MD5 hash algorithm.

```

19     extern bool GetLicenseFromDb(char* lic_code);
20
21     /*
22      * Calculate device hash string from hardware information
23     */
24
25     string GetDeviceHashString(
26         const char* hard_disk_sn,
27         const char* win_prod_id,
28         const char* mac_address,
29         const char* machine_name,
30         const char* cpu_id
31     )
32     {
33         MD5 md5;
34
35         // add hard disk serial number
36         const char* po = hard_disk_sn;
37         md5.update(po, strlen(po));
38
39         // add windows product id
40         po = win_prod_id;
41         md5.update(po, strlen(po));
42
43         // // add mac address
44         // po = mac_address;
45         // md5.update(po, strlen(po));
46
47         // // add computer name
48         // po = machine_name;
49         // md5.update(po, strlen(po));
50
51         // add cpu id
52         po = cpu_id;
53         md5.update(po, strlen(po));
54
55         md5.finalize();
56
57         return md5.toHexString();
58     }
59
60
61     /*

```

It has 5 parameters but we did not use mac address and computer name.

A user can change computer name on purpose and mac address can be changed if a user change his/her NIC.

So this function has 5 parameters but actually device id string is generated by 3 parameters.

If you want to enable/disable the parameter's role, comment/uncomment the code lines corresponding to the parameter.

3) Algorithm

For more information about licensing algorithm, please see “Kontrol Licensing Manual.pdf”.

4) Activation code generation

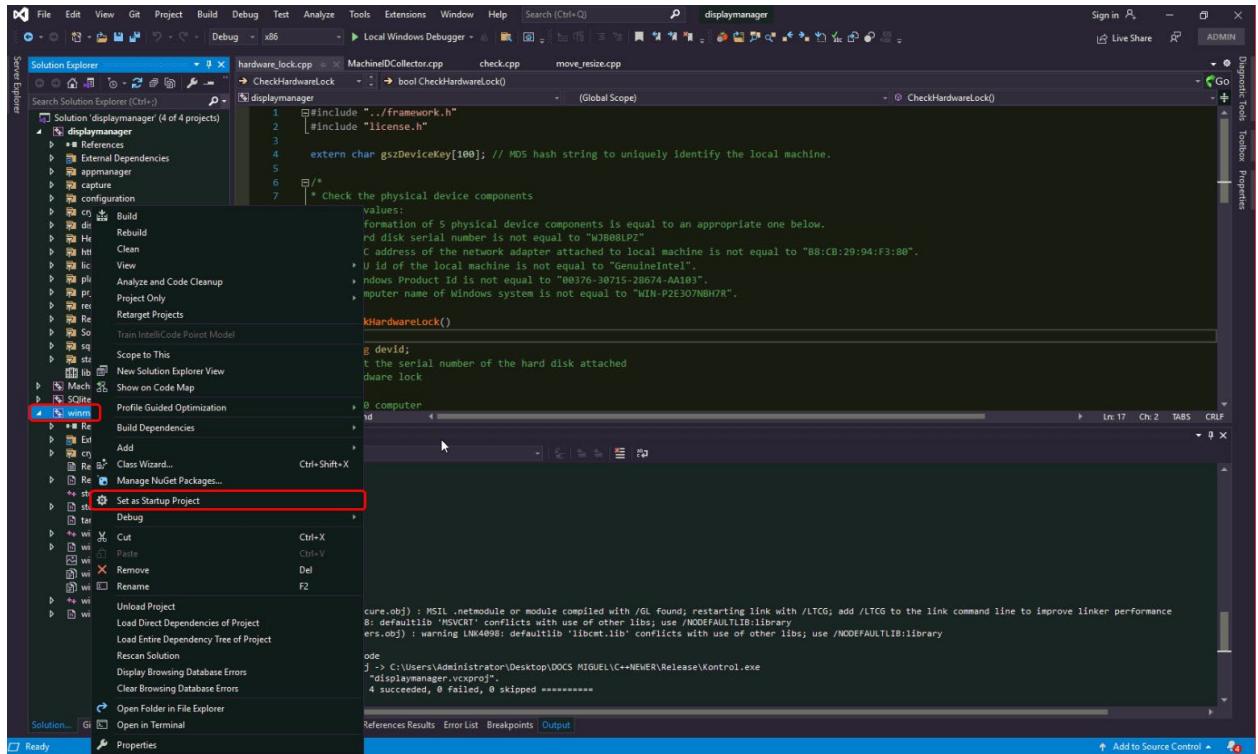
A provider generates activation code for users.

When a user delivers his/her device id string to provider, the provider generates activation code.

In this section, let's assume that you are a provider.

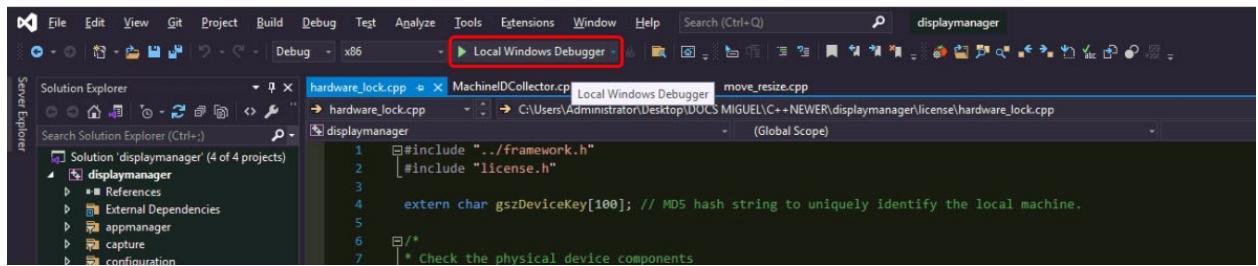
You can build “winman_keygen” project in a cpp app solution.

At first, please make winman_keygen be “startup” project.

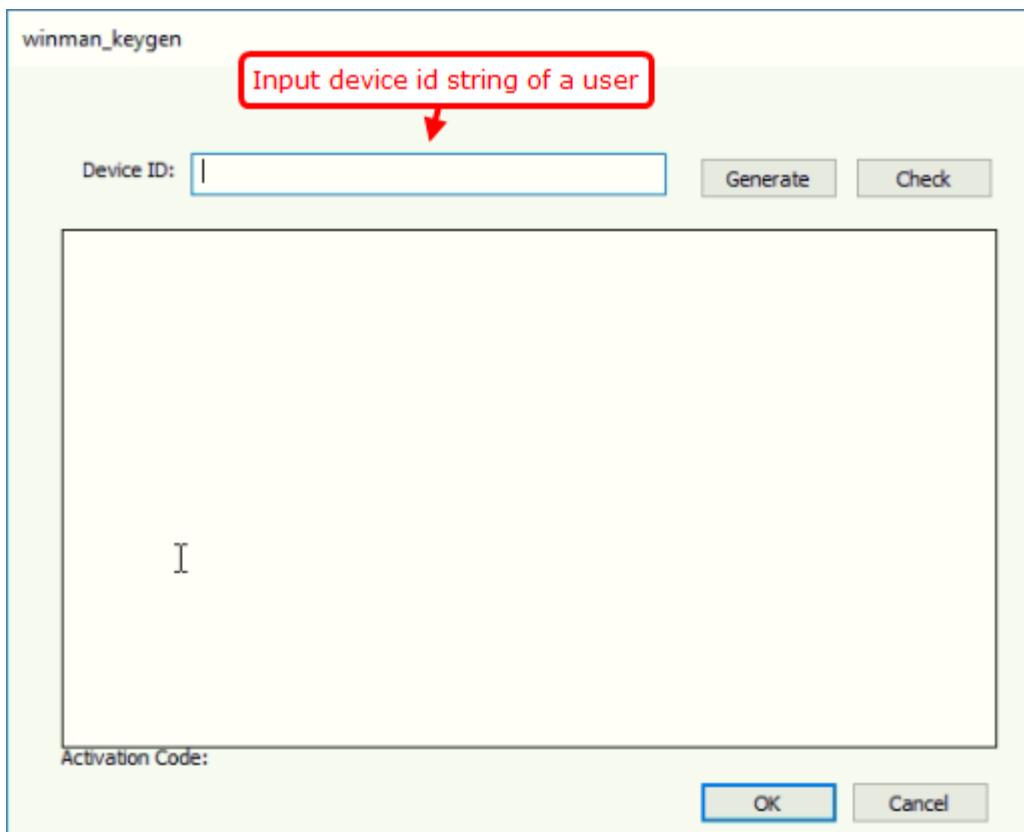


And press “F7” to build.

If you want to run in Visual Studio, press “F5” key or



You will see the keygen app window.



After input user's device id string, when you click generate, activation code is generated.

```

175 };
176
177 void CWinman_keygenDlg::OnBnClickedGenerate()
178 {
179     // TODO: Add your control notification handler code here
180     RSA2048_KEY_BLOB privkey;
181     CString str;
182     char* activate_code = NULL;
183     char dev_info[1024];
184     size_t devinfo_len = 0;
185
186     privkey.blob_len = sizeof(gauicPvKey);
187     privkey.blob = malloc(privkey.blob_len);
188     memcpy(privkey.blob, gauicPvKey, privkey.blob_len);
189
190     GetDlgItemText(EDIT_DEVID, str);
191
192     devinfo_len = wcstombs(dev_info, str.GetBuffer(), str.GetLength());
193
194     uint32_t dwSeq = 0;
195
196     time_t t = time(0);
    
```

Output window:

```

'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\shell32.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\cfgmgr32.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\ole32.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\oleaut32.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\olepro7.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\kernel32.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\user32.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\RPCRT4.dll'.
'winman_keygen.exe' (Win32): Loaded 'C:\Windows\System32\msasn1.dll'.
The thread 0x1e28 has exited with code 0 (0x0).
The thread 0x1554 has exited with code 0 (0x0).
The thread 0x1554 has exited with code 0 (0x0).
The thread 0x1d68 has exited with code 0 (0x0).
The thread 0x1d68 has exited with code 0 (0x0).
The thread 0x1d68 has exited with code 0 (0x0).
The thread 0x1d68 has exited with code 0 (0x0).
The program '[5248] winman_keygen.exe' has exited with code 0 (0x0).
    
```

OnBnClickedGenerate() function will make activation code and shows it on dialog box.

- *What is Keygen.info?*

Activation code contains “sequence” which is the number that represents the unique id of activation number.

Whenever a provider generates an activation code, this sequence number incremented by one.

At initial, it is 0.

So the first activation code has sequence number 0.

And it will be 1.

The second activation code has sequence number 1.

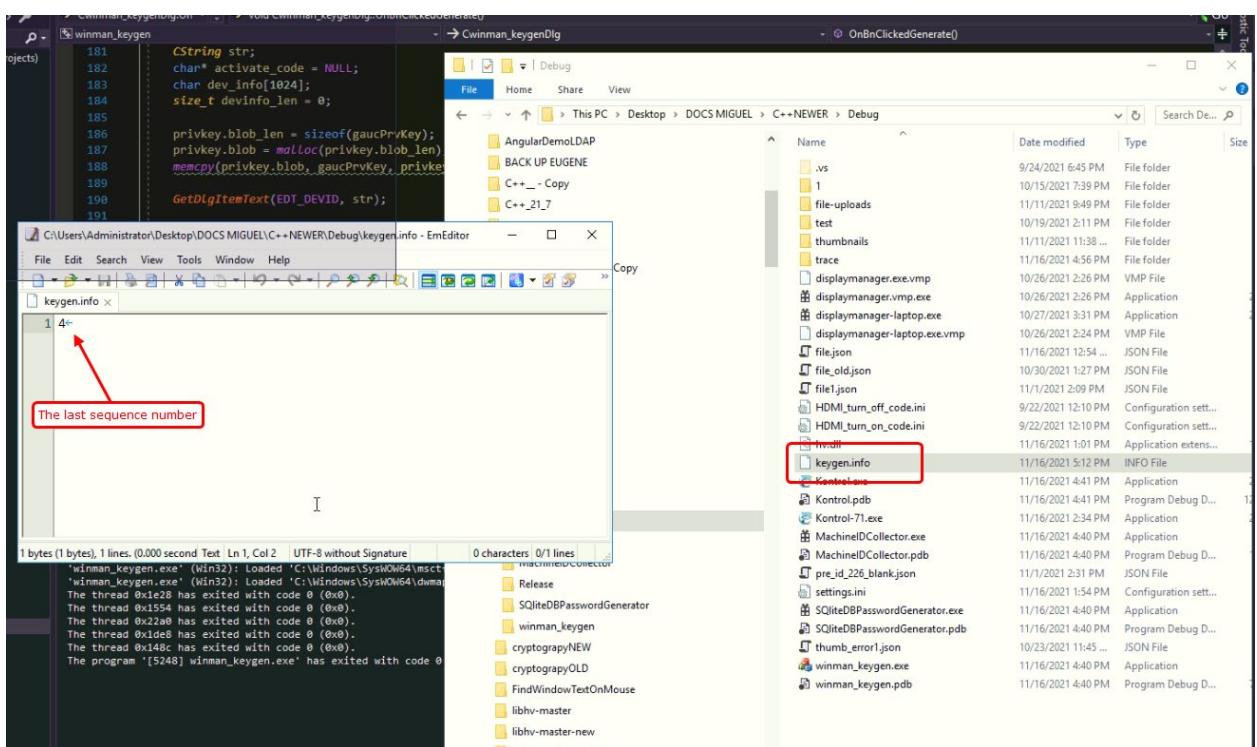
And it will be 2.

...

If a provider closes keygen app, next time this sequence number should be kept and incremented by one whenever a provider generates activation code.

So sequence number should be saved permanently.

We implemented it by “keygen.info” file.



This “keygen.info” file saves the last sequence number.

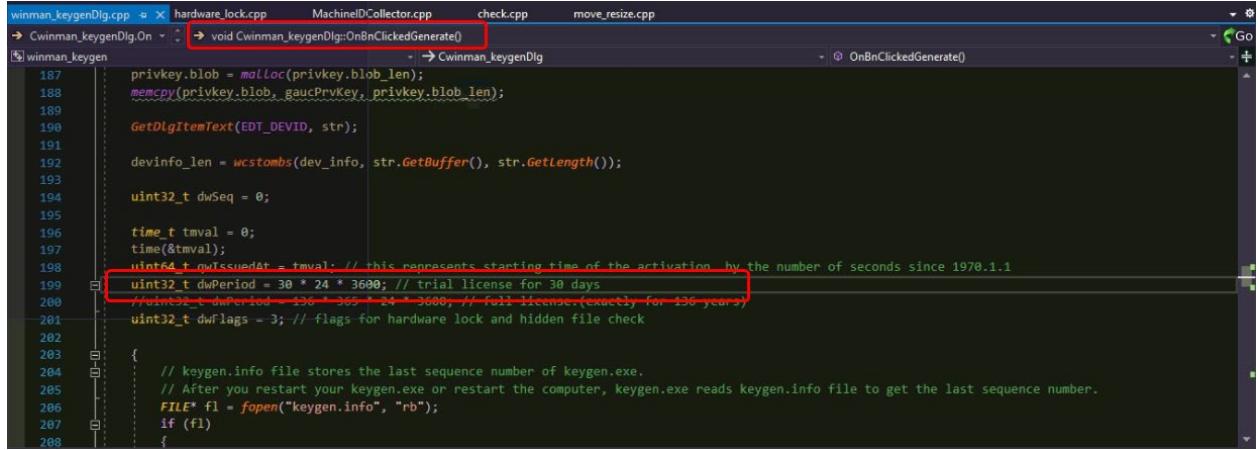
When a provider runs keygen app and generate an activation code, keygen app reads the content of “keygen.info” file and increment it by one, generates activation code by the sequence number and save the new sequence number back to “keygen.info” file.

In a word, “keygen.info” saves the last sequence number of all published activation codes.

If you change the content of “keygen.info” file, for example “101”, then 102 will be the sequence number of next activation code and 102 will be saved in “keygen.info” file.

- *How to change the trial period(expiration)?*

OnBnClickedGenerate() function contains this part.



```

winman_keygenDlg.cpp  hardware_lock.cpp  MachineIDCollector.cpp  check.cpp  move_resize.cpp
→ Cwinman_keygenDlg.On  → void Cwinman_keygenDlg::OnBnClickedGenerate()
`- winman_keygen
187     privkey.blob = malloc(privkey.blob_len);
188     memcpy(privkey.blob, gaucPrvKey, privkey.blob_len);
189
190     GetDlgItemText(EDT_DEVID, str);
191
192     devinfo_len = wcstombs(dev_info, str.GetBuffer(), str.GetLength());
193
194     uint32_t dwSeq = 0;
195
196     time_t tmval = 0;
197     time(&tmval);
198     uint64_t dwIssuedAt = tmval; // this represents starting time of the activation by the number of seconds since 1970.1.1
199     uint32_t dwPeriod = 30 * 24 * 3600; // trial license for 30 days
200     //dwPeriod = 138 * 30 * 24 * 3600; // full license(exactly for 138 years)
201     uint32_t dwFlags = 3; // flags for hardware lock and hidden file check
202
203     {
204         // keygen.info file stores the last sequence number of keygen.exe.
205         // After you restart your keygen.exe or restart the computer, keygen.exe reads keygen.info file to get the last sequence number.
206         FILE* f1 = fopen("keygen.info", "rb");
207         if (f1)
208             {

```

In this picture, dwPeriod is $30 * 24 * 3600 = 2592000$. That is trial period is 2592000 seconds = 30 days.

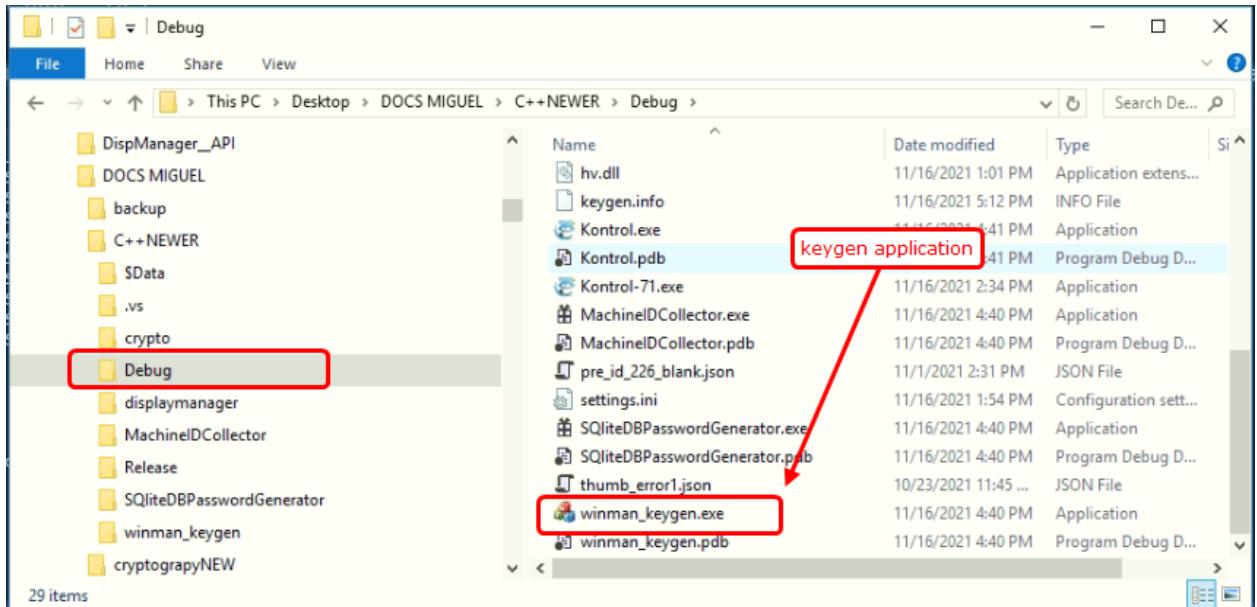
If you want to change the trial period, please change dwPeriod value.

For example, the table below shows the case of “dwPeriod” variable.

C++ code	seconds	Description
dwPeriod = 3600	3600	will expire after 1 hour.
dwPeriod = 24 * 3600	86400	will expire after 1 day
dwPeriod = 30 * 24 * 3600	2592000	will expire after 30 days(1 month)
dwPeriod = 3 * 30 * 24 * 3600	7776000	will expire after 3 months
dwPeriod = 12 * 30 * 24 * 3600	31104000	will expire after 12 months(1 year)
dwPeriod = 5 * 12 * 30 * 24 * 3600	155520000	will expire after 5 years
dwPeriod = 138 * 12 * 30 * 24 * 3600	4292352000	will expire after 138 years (Full license)

dwPeriod is 4-byte integer. Thus the maximum trial period is 138 years. As a matter of fact, it is equal to “full” license.

After changing the value of “dwPeriod”, you have to build keygen app.



- *What is flag?*

When a provider generates an activation code, flag value is used.

```

privkey.blob = malloc(privkey.blob_len);
memcpy(prvkey.blob, gauPrvKey, privkey.blob_len);

GetDlgItemText(EDT_DEVID, str);

devinfo_len = wcstombs(dev_info, str.GetBuffer(), str.GetLength());

uint32_t dwSeq = 0;

time_t tval = 0;
time(&tval);

uint64_t qIssuedAt = tval; // this represents starting time of the activation. by the number of seconds since 1970.1.1
uint32_t dwPeriod = 30 * 24 * 3600; // trial license for 30 days
//uint32_t dwPeriod = 136 * 365 * 24 * 3600; // full license.(exactly for 136 years)

uint32_t dwFlags = 3; // flags for hardware lock and hidden file check

{
    // keygen.info file stores the last sequence number of keygen.exe.
    // After you restart your keygen.exe or restart the computer, keygen.exe reads keygen.info file to get the last sequence number.
    FILE* f1 = fopen("keygen.info", "rb");
    if (f1)
    {
        ...
    }
}

```

This value 3 means that this activation code is for “Kontrol”. In other words, it is the type of software which requires license. 3 means “Kontrol”.

Our cpp app(Kontrol) checks this flag value from activation code and if it is not 3, cpp app fails to run.

In displaymanager project, there is a file named “flags.cpp” in license folder.

```

1 //include "../framework.h"
2 //include "license.h"
3
4 extern uint32_t gdwFlags;
5
6 bool IsValidFlag()
7 {
8     return gdwFlags == 3;
9 }
10

```

IsValidFlag() function checks the flag value of activation code.

"gdwFlags" is a global variable which is gotten from a valid activation code.

If it is not 3, cpp app considers the activation code is not valid.

So if you want to change flag value, for example to 101, then you should change the value in "winman_keygenDlg.cpp" file of "winman_keygen" project like

```

187 privateKey_blob = malloc(privateKey_blob_len);
188 memcpy(privateKey_blob, gauPvKey, privateKey_blob_len);
189
190 GetDlgItemText(EDIT_DEVID, str);
191
192 devInfo_len = wcstombs(dev_info, str.GetBuffer(), str.GetLength());
193
194 uint32_t dwSeq = 0;
195
196 time_t tmval = 0;
197 time(&tmval);
198 uint64_t qwlssedAt = tmval; // this represents starting time of the activation. by the number of seconds since 1970.1.1
199 uint32_t dwPeriod = 30 * 24 * 3600; // trial license for 30 days
200 //uint32_t dwPeriod = 136 * 365 * 24 * 3600; // full license(exactly for 136 years)
201 uint32_t dwFlags = 3; // flags for hardware lock and hidden file check
202
203 {
204     // keygen.info file stores the last sequence number of keygen.exe.
205     // After you restart your keygen.exe or restart the computer, keygen.exe reads keygen.info file to get the last sequence number.
206     FILE* f1 = fopen("keygen.info", "rb");
207     if (f1)
208     {

```

And in "flags.cpp" of "displaymanager" project like

The screenshot shows the Microsoft Visual Studio IDE interface. In the Solution Explorer, the file 'flags.cpp' is selected. In the code editor, there is a function named 'isValidFlag()' with the following code:

```

1  #include "../framework.h"
2  #include "license.h"
3
4  extern uint32_t gdwFlags;
5
6  bool isValidFlag()
7  {
8      return gdwFlags == 3;
9  }
10

```

A red box highlights the line 'return gdwFlags == 3;'. A callout box points to the value '101' in the line above, with the text 'This value to 101'.

And build both projects at a time.

Cpp app (Kontrol) will run with flag value 101 by the activation code generated by newly built keygen app because newly built keygen app will use flag value 101.

5) sqlite db management

In Cpp app, licensing data are stored in sqlite database file with password.

"license_db.cpp" in "license" folder contains several functions to handle data in sqlite db.

The screenshot shows the Microsoft Visual Studio IDE interface. In the Solution Explorer, the file 'license_db.cpp' is selected. In the code editor, there is a function named 'GetSqliteDbInstance()' with the following code:

```

4
5  #pragma comment (lib, "sqlite3.lib")
6
7  #define DB_FILENAME    "sysrq1.bin"
8
9  void GenerateSQLiteDBPassword(const char* devid, char* dbpass);
10
11 extern char gsDeviceKey[100]; // MD5 hash string to uniquely identify the local machine.
12 extern uint64_t gqLicRefTime;
13 extern uint64_t gqLicElapsedTime;
14
15 /**
16  * Get licensing sqlite db instance from local db file.
17  */
18 int GetSqliteDbInstance(sqlite3** ppdb, bool bDeleteDb)
19 {
20     int rc;
21
22     char szSystemPath[MAX_PATH];
23     memset(szSystemPath, 0, sizeof(szSystemPath));
24     UINT ret = GetWindowsDirectoryA(szSystemPath, sizeof(szSystemPath));
25     char szDBPath[MAX_PATH];

```

A red box highlights the line 'return gdwFlags == 3;'. A callout box points to the value '101' in the line above, with the text 'This value to 101'.

Let's consider some issues here.

- How to change the sqlite db file path and name?

In license_db.cpp, there is a function named "GetSqliteDbInstance".

This function handles the sqlite db file.

```
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) displaymanager
Solution Explorer Local Windows Debugger x86 Search Solution Explorer (Ctrl+Shift+F)
References External Dependencies appmanager capture configuration crypto displaysignal Header Files http license basic_op.cpp check.cpp Device.cpp Device.h expiration.cpp flags.cpp hardware_lock.cpp hidden_file.cpp license.h license_db.cpp playback pr_data redundancy Resource Files Source Files sqlite startup libcurl_a.lib MachineIDCollector SQLiteDBPasswordGenerator
4 #pragma comment (lib, "sqlite/sqlite3.lib")
5
6 #define DB_FILENAME "sysrq1.bin"
7
8 void GenerateSQLiteDBPassword(const char* devId, char* dbpass);
9
10 extern char gszDeviceKey[100]; // MD5 hash string to uniquely identify the local machine.
11 extern uint64_t gowlicRefTime;
12 extern uint64_t gowlicElapsedTime;
13
14 */
15
16 /* Get licensing sqlite db instance from local db file.
17 */
18 int GetSqliteDbInstance(sqlite3** ppdb, bool bDeleteDb)
19 {
20     int rc;
21
22     char szSystemPath[MAX_PATH];
23     memset(szSystemPath, 0, sizeof(szSystemPath));
24     UINT ret = GetWindowsDirectoryA(szSystemPath, sizeof(szSystemPath));
25     char szDBPath[MAX_PATH];
26     sprintf(szDBPath, "%s\\%s", szSystemPath, DB_FILENAME);
27
28     if (bDeleteDb)
29         return DeleteFileA(szDBPath);
30
31     rc = sqlite3_open(szDBPath, ppdb);
32     if (rc != SQLITE_OK)
```

Now in this function sqlite db file is created at "C:\Windows\sysrq1.bin".

If you want to change the sqlite db file path,

```
18 int GetSqliteDbInstance(sqlite3** ppdb, bool bDeleteDb)
19 {
20     int rc;
21
22     char szSystemPath[MAX_PATH];
23     memset(szSystemPath, 0, sizeof(szSystemPath));
24     UINT ret = GetWindowsDirectoryA(szSystemPath, sizeof(szSystemPath));
25     char szDBPath[MAX_PATH];
26     sprintf(szDBPath, "%s\\%s", szSystemPath, DB_FILENAME);
27
28     if (bDeleteDb)
29         return DeleteFileA(szDBPath);
30
31     rc = sqlite3_open(szDBPath, ppdb);
32     if (rc != SQLITE_OK)
```

In GetSqliteDbInstance function, change the variable szDBPath.

For example, if you want to create sqlite db file at D:\HiddenPath\r1.bin, then please change the code like

```
sprint(szDBPath, "D:\\HiddenPath\\r1.bin");
```

Then sqlite db file will be created at that location.

- *How to make sqlite db file without password*

In GetSqliteDbInstance function, you will see the code lines like

```

16     * Get licensing sqlite db instance from local db file.
17
18     int GetSqliteDbInstance(sqlite3** ppdb, bool bDeleteDb)
19     {
20         int rc;
21
22         char szSystemPath[MAX_PATH];
23         memset(szSystemPath, 0, sizeof(szSystemPath));
24         UINT ret = GetWindowsDirectory(szSystemPath, sizeof(szSystemPath));
25         char szDBPath[MAX_PATH];
26         sprintf(szDBPath, "%s\\%s", szSystemPath, DB_FILENAME);
27
28         if (bDeleteDb)
29             return DeleteFile(szDBPath);
30
31         rc = sqlite3_open(szDBPath, ppdb);
32         if (rc != SQLITE_OK)
33             return rc;
34
35         char szDBPassword[0x100] = { 0, };
36         GenerateSqliteDBPassword(gsDeviceKey, szDBPassword);
37         rc = sqlite3_key(*ppdb, szDBPassword, strlen(szDBPassword));
38         return rc;
39     }
40
41     /* sqlite record values callback
42 */
43     /* sqlite3_set_record_values_callback */

```

“sqlite3_key” function sets the encryption to sqlite db file by the password.

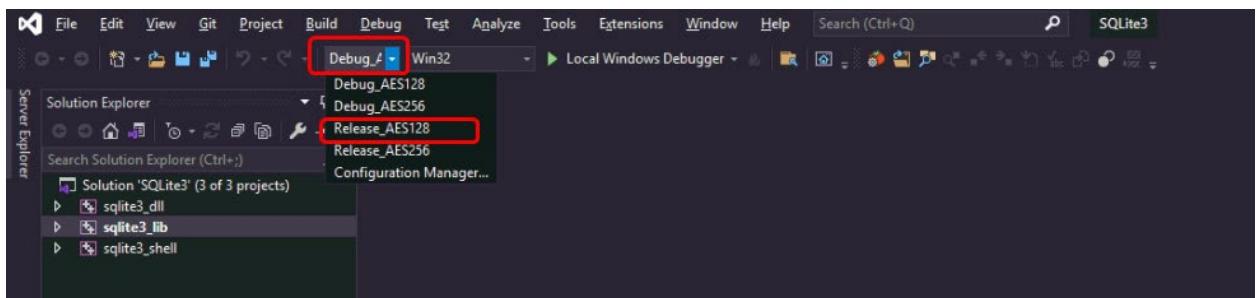
So if you want to make sqlite db file without password, please remove these 3 lines.

- *How to open sqlite db file with password*

By default, sqlite db file is created with password.

Its encryption method is set by sqlite library which is linked to cpp app project.

Because we built sqlite library in Release_AES128, sqlite db file will be encrypted by AES128 algorithm.



(See section “Sqlite library with encryption”)

In “displaymanager” project, “license_db.cpp” file contains GetSqliteDbInstance() function.

Please look at the function codes.

```

16     * Get licensing sqlite db instance from local db file.
17
18     int GetSQLiteDbInstance(sqlite3** ppdb, bool bDeleteDb)
19     {
20         int rc;
21
22         char szSystemPath[MAX_PATH];
23         memset(szSystemPath, 0, sizeof(szSystemPath));
24         UINT ret = GetWindowsDirectory(szSystemPath, sizeof(szSystemPath));
25         char szDBPath[MAX_PATH];
26         sprintf(szDBPath, "%s\\%s", szSystemPath, DB_FILENAME);
27
28         if (bDeleteDb)
29             return DeleteFile(szDBPath);
30
31         rc = sqlite3_open(szDBPath, ppdb);
32         if (rc != SQLITE_OK)
33             return rc;
34
35         char szDBPassword[0x100] = { 0, };
36         GenerateSQLiteDBPassword(szDeviceKey, szDBPassword);
37         rc = sqlite3_key(*ppdb, szDBPassword, strlen(szDBPassword));
38
39     }
40
41     /* sqlite record values callback
42 */
43     /* sqlite record values callback
44 */

```

You can see it calls “GenerateSQLiteDBPassword” function.

This function makes sqlite db file password based on device id string.

So we made a tool to know the password string.

“SQLiteDBPasswordGenerator” project is the one for tool.

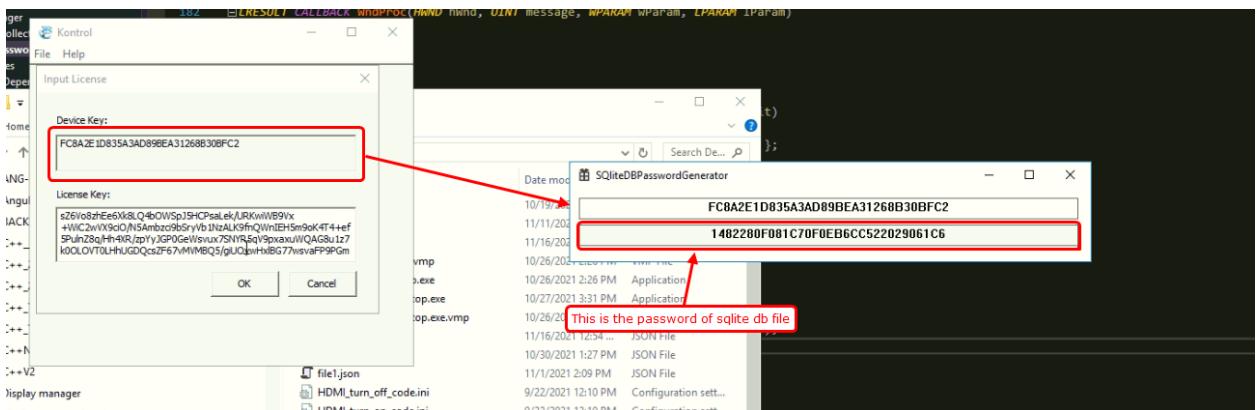
```

181     LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
182     {
183         switch (message)
184         {
185             case WM_COMMAND:
186                 {
187                     if (HIWORD(wParam) == EN_CHANGE && lParam == (LPARAM)ghInputEdit)
188                     {
189                         char szIndText[0x200] = { 0, };
190                         GetWindowText(ghInputEdit, szIndText, 0x1f);
191                         GenerateSQLiteDBPassword(szIndText, szDBPassword);
192                         SetWindowText(ghPassEdit, szDBPassword);
193                     }
194
195                     int wMid = LOWORD(wParam); // Same function in cpp app
196                     // Parse the menu selection
197                     switch (wMid)
198                     {
199                         case IDM_ABOUT:
200                             DialogBox(hInst, MAKEINTRESOURCE(IDD_ABOUTBOX), hWnd, About);
201                             break;
202                         case IDM_EXIT:
203                             DestroyWindow(hWnd);
204                             break;
205                         default:
206                             return DefWindowProc(hWnd, message, wParam, lParam);
207                     }
208                 }
209             }
210         }
211     }

```

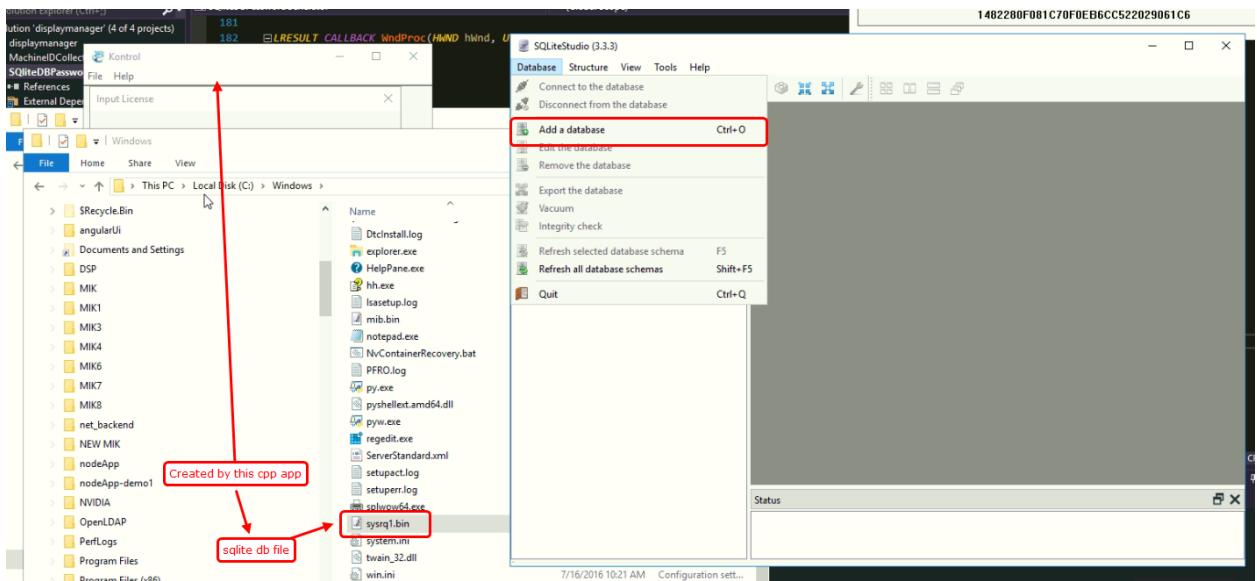
This tool generates the password of sqlite db file.

Please run this tool by pressing F5 key or double-click SQLiteDBPasswordGenerator.exe in Debug/Release folder.

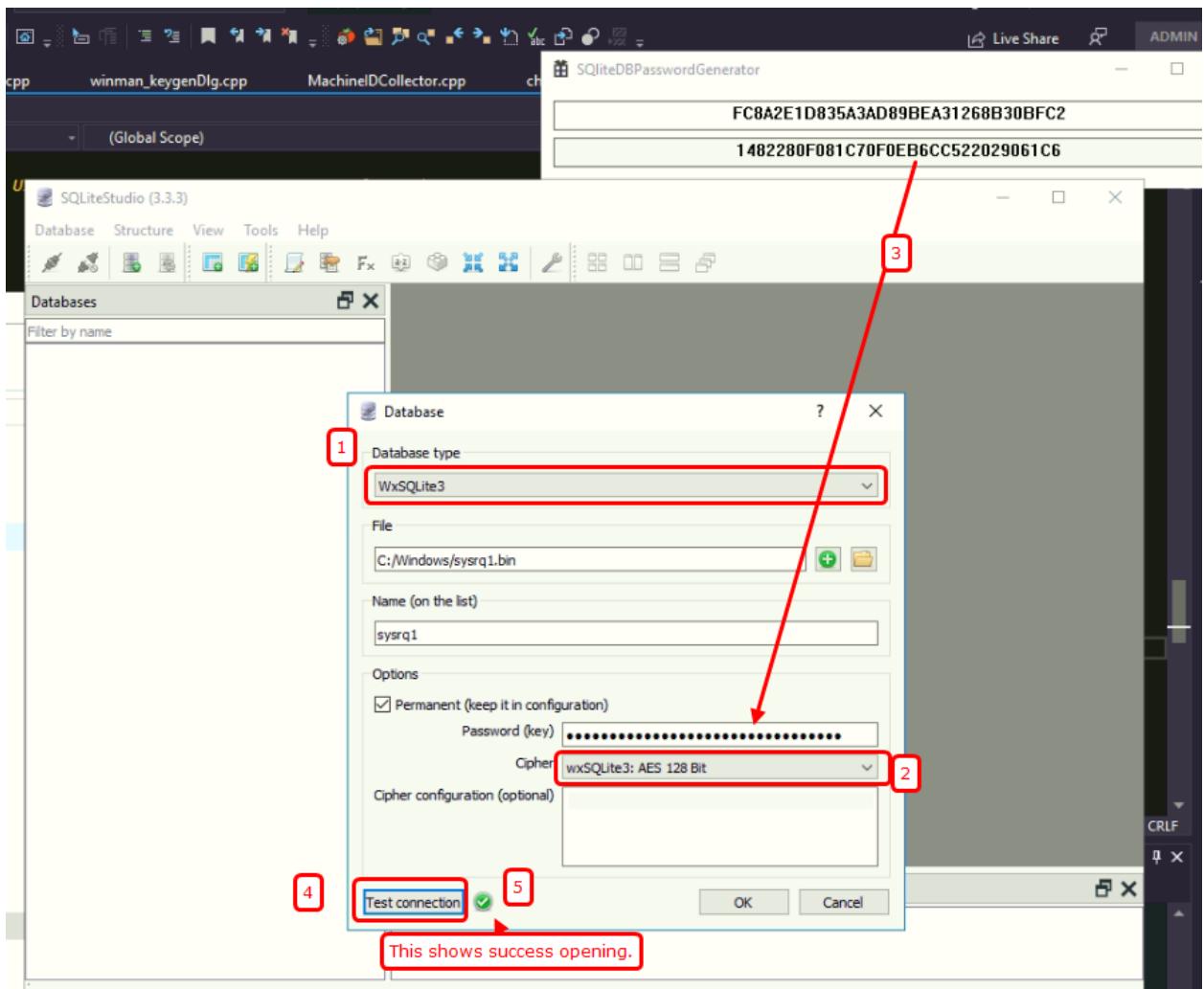


Don't forget. This is the password for sqlite db file which was created by cpp app that has the device id string.

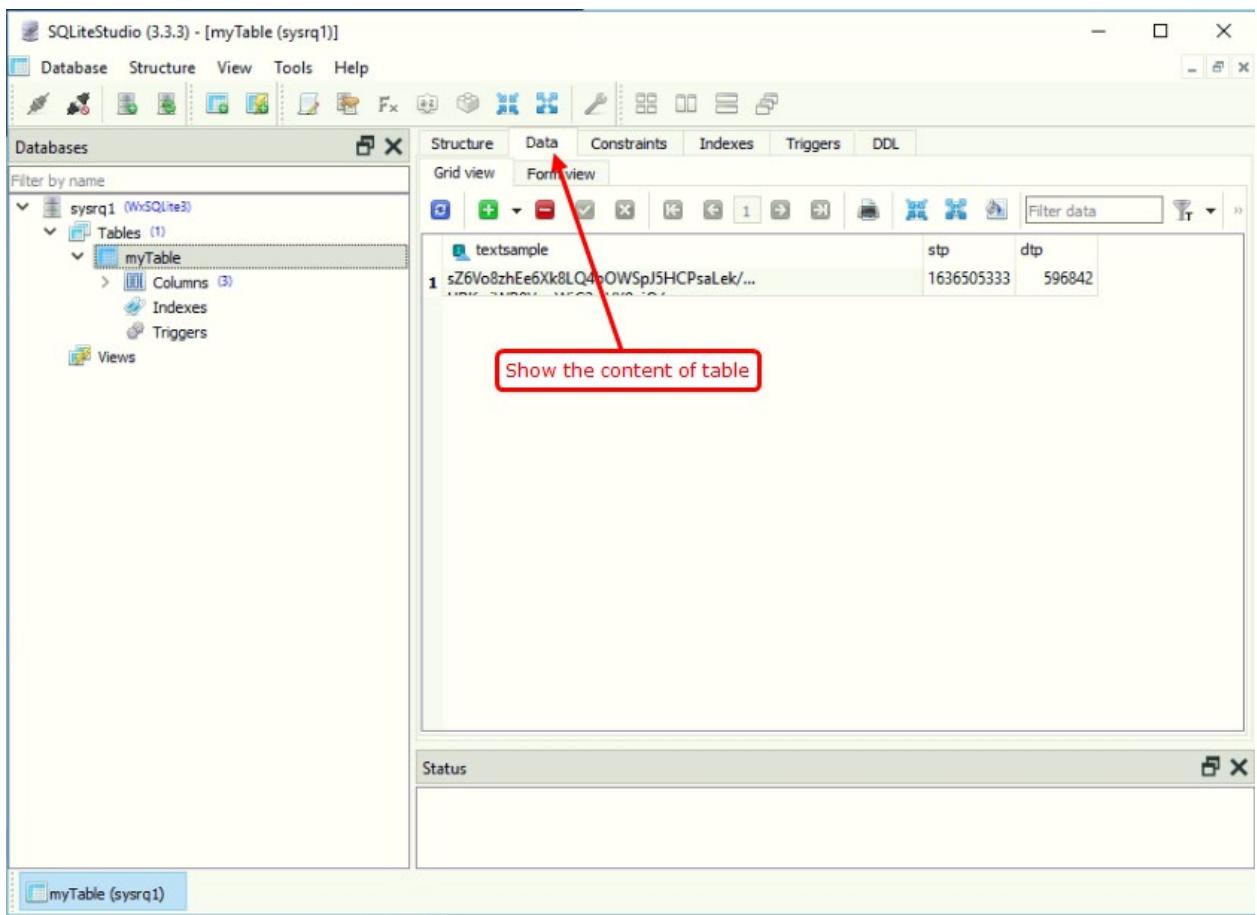
To view the content of sqlite db file, we use SQLiteStudio.



To add a database, select WxSQLite3.



The result is open in the tool like



6) Periodic check

Cpp app checks the validity of license periodically.

It exactly checks the validity of activation code, expiration of license, hidden file check, hardware lock check every 10 seconds.

“displaymanager” project => playback folder => main_process.cpp file has MainThread() function.

This function is the thread function for this application.

This function does periodic check of license.

```

311     MinimizedOtherApps();
312
313     bHttpRequestDontCloseOldLayout = false;
314
315     clearDesktop();
316     // Start a new layout...
317     StartLayout(gPresentationIdPlaying, gnLayoutPlaying);
318
319
320     time_t curTime = GetCurTime();
321     if (lastProcessTime == curtime)
322         continue;
323     // check every second
324     // 10 means the period of the checking license. you can change the value to 30 if you make the period to 30 seconds.
325     if (curTime > licTime + 10)
326     {
327         licTime = curTime;
328
329         HWND hwnd = GetMainWindow();
330
331         DWORD dwMyID = ::GetCurrentThreadId();
332         DWORD dwCurID = ::GetWindowThreadProcessId(hwnd, NULL);
333         ::AttachThreadInput(dwCurID, dwMyID, TRUE);
334
335         do
336         {
337             // trial expiration check...
338             CheckLicenseExpiration();
339             // check the validity of activation code
340
341         } while (true);
342
343         dwCurID = dwMyID;
344         dwMyID = dwCurID;
345
346         ::AttachThreadInput(dwCurID, dwMyID, FALSE);
347
348         dwCurID = dwMyID;
349
350         dwMyID = dwCurID;
351
352         dwCurID = dwMyID;
353
354         dwMyID = dwCurID;
355
356         dwCurID = dwMyID;
357
358         dwMyID = dwCurID;
359
360         dwCurID = dwMyID;
361
362         dwMyID = dwCurID;
363
364         dwCurID = dwMyID;
365
366         dwMyID = dwCurID;
367
368         dwCurID = dwMyID;
369
370         dwMyID = dwCurID;
371
372         dwCurID = dwMyID;
373
374         dwMyID = dwCurID;
375
376         dwCurID = dwMyID;
377
378         dwMyID = dwCurID;
379
380         dwCurID = dwMyID;
381
382         dwMyID = dwCurID;
383
384         dwCurID = dwMyID;
385
386         dwMyID = dwCurID;
387
388         dwCurID = dwMyID;
389
390         dwMyID = dwCurID;
391
392         dwCurID = dwMyID;
393
394         dwMyID = dwCurID;
395
396         dwCurID = dwMyID;
397
398         dwMyID = dwCurID;
399
399     }

```

You can see the code line like above.

If you want to change to period, please change the number. This number is the amount of seconds.

If you change the number to 30, then cpp app will check the validity of the license every 30 seconds.

7) Hardware lock check

“displaymanager” project => license folder => hardware_lock.cpp file has a function named “CheckHardwareLock”.

This function returns true if the hardware lock is true.

Here calls the function GetDeviceHashString which has 5 parameters.

About GetDeviceHashString, I mentioned at “Hash” section.

Here we don’t explain about the GetDeviceHashString.

You can change the parameter values with the parameters which you want.

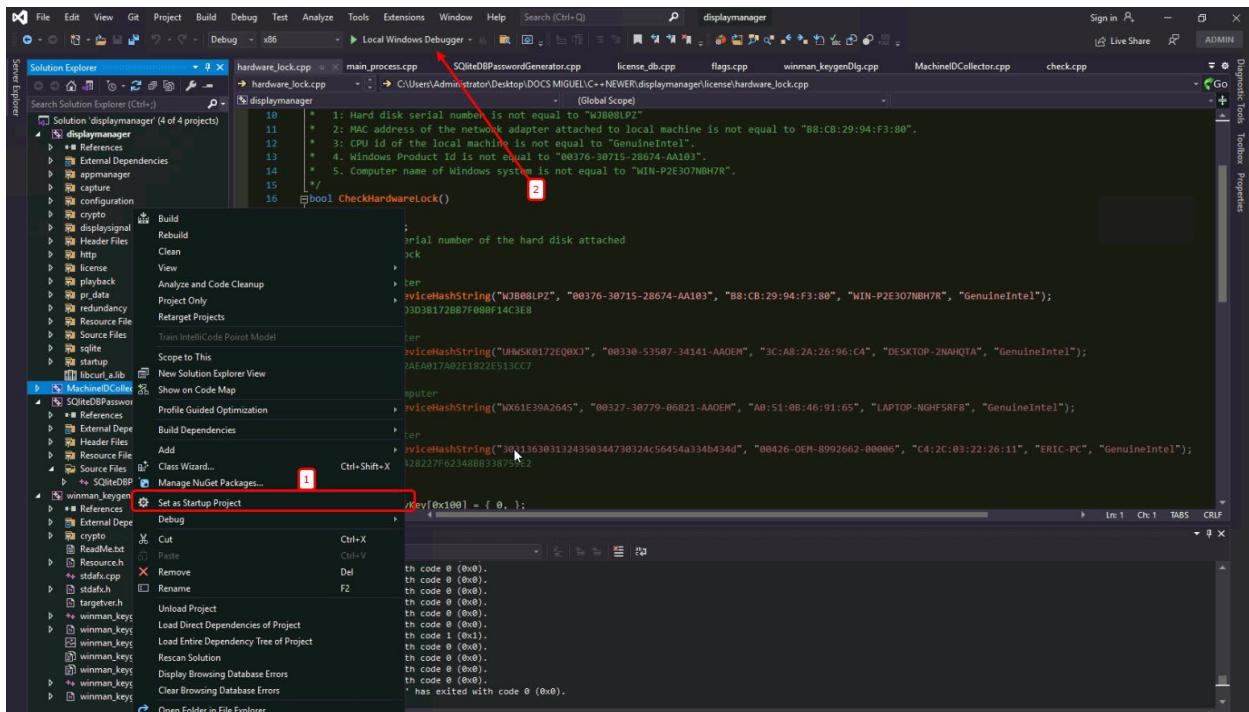
```

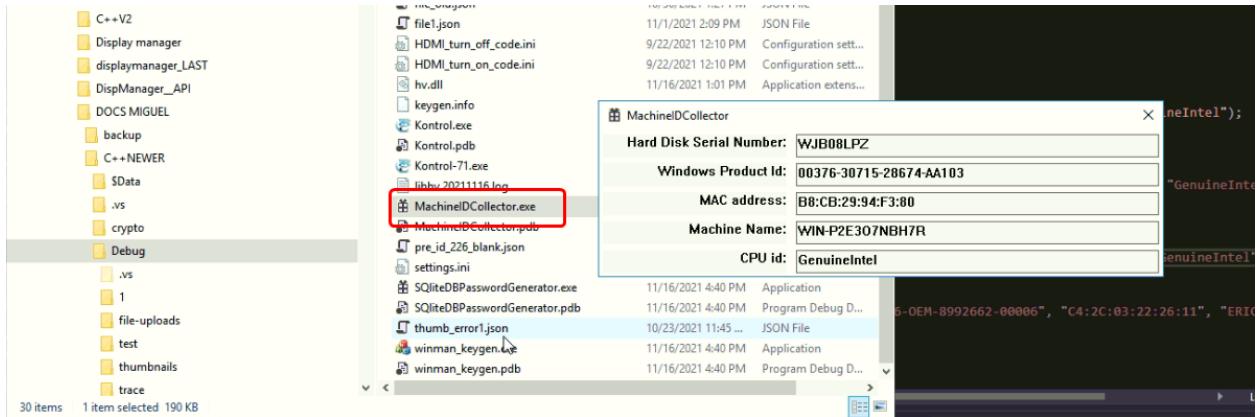
hardware_lock.cpp  main_process.cpp  SQLiteDBPasswordGenerator.cpp  license_db.cpp  flags.cpp  winman_keygenDlg.cpp  MachineIDCollector.cpp  check.cpp
displaymanager
hardware_lock.cpp
displaymanager
10 * 1: Hard disk serial number is not equal to "WJBB8LPZ"
11 * 2: MAC address of the network adapter attached to local machine is not equal to "B8:CB:29:94:F3:80".
12 * 3: CPU id of the local machine is not equal to "GenuineIntel".
13 * 4: Windows Product Id is not equal to "00376-30715-28674-AA103".
14 * 5: Computer name of Windows system is not equal to "WIN-P2E307NBH7R".
15 */
16 bool CheckHardwareLock()
17 {
18     string devid;
19     // get the serial number of the hard disk attached
20     //hardware lock
21 #if 1
22     // .70 computer
23     devid = GetDeviceHashString("WJBB8LPZ", "00376-30715-28674-AA103", "B8:CB:29:94:F3:80", "WIN-P2E307NBH7R", "GenuineIntel");
24     //FFFA05C128ED038172B87F08F14C3E8
25 #elif 1
26     // .71 computer
27     devid = GetDeviceHashString("UHMSK0172EQ0X3", "00330-53507-34141-AA0EM", "3C:AB:2A:26:96:C4", "DESKTOP-ZNAHQTA", "GenuineIntel");
28     //4370AE69L2AE81A2E182E513CC7
29 #elif 1
30     // laptop computer
31     devid = GetDeviceHashString("WX61E39A2645", "00327-30779-06821-AA0EM", "A8:51:0B:46:91:65", "LAPTOP-NGHFSRF8", "GenuineIntel");
32 #else
33     // .10 computer
34     devid = GetDeviceHashString("3031363031324350344730324c56454a334b434d", "00426-OEM-8992662-00006", "C4:2C:03:22:26:11", "ERIC-PC", "GenuineIntel");
35     //0290F070428227F623488B338759E2
36 #endif
37     char fixedDevKey[0x100] = { 0, };

```

The thread 0x32b8 has exited with code 0 (0x0).

To get the hardware information of the computer, please run MachineIDCollector.





You can replace the parameters of GetDeviceHashString in CheckHardwareLock().

But as mentioned at “Hash” section, mac address and computer name don’t take part in calculation of device hash string.

8) Hidden file check

Cpp app also checks the hidden file.

It first checks if the hidden file exists in the supposed path.

If it exists, cpp app checks if content is the valid activation code.

```

1 //include "../framework.h"
2 //include "license.h"
3
4 #define ENC_FILENAME "cirenc.dll"
5
6 extern char gszLicenseKey[0x400];
7
8 /* return value: true if it reads license string from a file "cirenc.dll" in Windows directory, false otherwise.
9 * For instance, "C:\Windows\cirenc.dll" would contain 256-bytes license string.
10 * remark: string is read as a binary data.
11 */
12
13 bool CheckHiddenFile()
14 {
15     char szWindowsPath[MAX_PATH];
16     memset(szWindowsPath, 0, sizeof(szWindowsPath));
17     UINT ret = GetWindowsDirectory(szWindowsPath, sizeof(szWindowsPath));
18     char szEncPath[MAX_PATH];
19     memset(szEncPath, 0, sizeof(szEncPath));
20     sprintf(szEncPath, "%s\\%s", szWindowsPath, ENC_FILENAME);
21     FILE* fp = fopen(szEncPath, "rb");
22     if (fp == NULL)
23         return false;
24     fread(gszLicenseKey, MAX_PATH, 1, fp);
25     fclose(fp);
26     return true;
27 }
28
29 /* return value: true if license string is stored to a file "cirenc.dll" in Windows directory, false otherwise.

```

- How to change the hidden file name?

Hidden file is located in Windows folder.

The name is defined as a macro in hidden_file.cpp.

```

1  #include "../framework.h"
2  #include "license.h"
3
4  #define ENC_FILENAME "cirenc.dll"
5
6  extern char gsLicenseKey[0x400];
7
8  /*
9   * return value: true if it reads license string from a file "cirenc.dll" in Windows directory, false otherwise.
10  * for instance, "C:\Windows\cirenc.dll" would contain 256 bytes license string.
11  * remark: string is read as a binary data.
12 */
13
14 bool CheckHiddenFile()
15 {
16     char szWindowsPath[MAX_PATH];
17     memset(szWindowsPath, 0, sizeof(szWindowsPath));
18     UINT ret = getWindowsDirectory(szWindowsPath, sizeof(szWindowsPath));
19     char szEncPath[MAX_PATH];
20     memset(szEncPath, 0, sizeof(szEncPath));
21     sprintf(szEncPath, "%s\%s", szWindowsPath, ENC_FILENAME);
22     FILE* fp = fopen(szEncPath, "rb");
23     if (fp == NULL)
24         return false;
25     fread(gsLicenseKey, MAX_PATH, 1, fp);
26     fclose(fp);
27     return true;
28 }
29
30 /* return value: true if license string is stored to a file "cirenc.dll" in Windows directory, false otherwise.

```

If you want to change the name of a hidden file, please replace the string with what you want.

Then hidden file will be C:\Windows\<YourHiddenFile>.

4. web commands

All http commands are processed in “http_handler.cpp” file of “displaymanager” project.

```

133 void Request_Handler(webserver::http_request* r)
134 {
135     Inner_Request_Handler(r);
136 }
137
138 void Inner_Request_Handler(webserver::http_request* r)
139 {
140     Socket s = *(r->s_);
141     string title;
142     string body;
143     string bgcolor = "#fffffe";
144     string links = "";
145     string response = "";
146     string status = "";
147
148     string urlPath = r->path_;
149
150     TraceLog("Request: %s", urlPath.c_str());
151
152     // at first, it splits the RUL string into string vector
153     // for example, 192.168.0.70:9001/thumbNail-quality/1 => "192.168.0.70:9001", "thumbNail-quality", "1"
154     vector<string> urlParams = split(urlPath, '/');
155     if (urlParams.size() == 1)
156         return;
157
158     if (urlParams[1] == "") { // checks connectivity of cpp app to webpage.
159         // http://192.168.0.70:9001/
160         if (status == "")
161             {
162                 ...
163             }
164     }
165 }

```

`Inner_Request_Handler()` function processes all http commands from angular webpage.

At first cpp app splits the URL string by ‘/’ character.

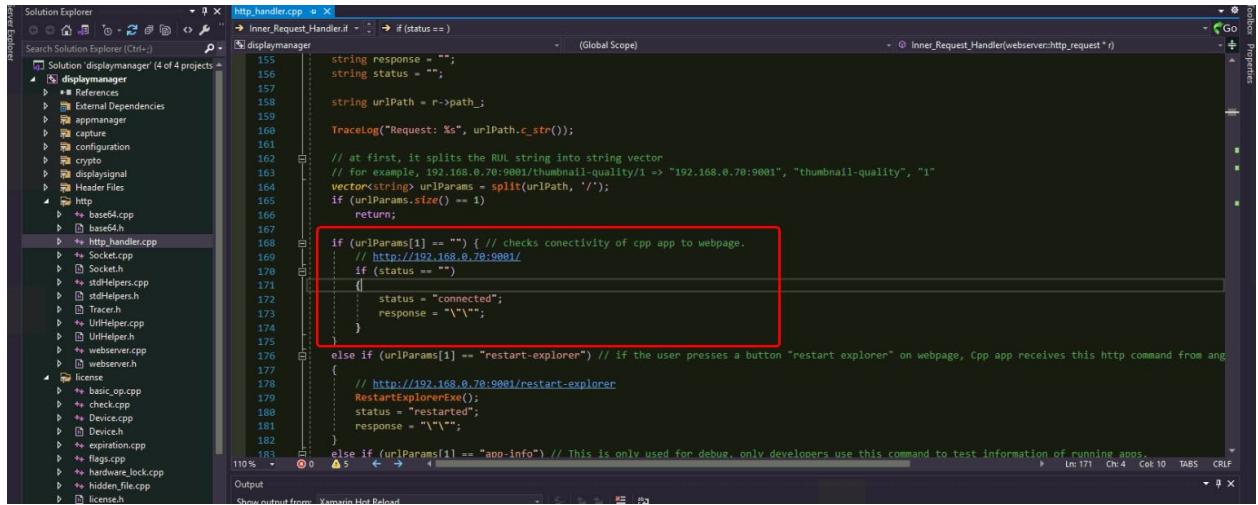
For example, 192.168.0.70:9001/play-layout/223/1/admin/ will be split into "192.168.0.70:9001", "play-layout", "223", "1", "admin", ""

1) Connection

URL is like

```
http://192.168.0.70:9001/
```

This command is not used now.



```
155     string response = "";
156     string status = "";
157
158     string urlPath = r->path_;
159
160     TraceLog("Request: %s", urlPath.c_str());
161
162     // at first, it splits the RUL string into string vector
163     // for example, 192.168.0.70:9001/thumbnail-quality/1 => "192.168.0.70:9001", "thumbnail-quality", "1"
164     vector<string> urlParams = split(urlPath, '/');
165     if (urlParams.size() == 1)
166         return;
167
168     if (urlParams[1] == "") { // checks connectivity of cpp app to webpage.
169         // http://192.168.0.70:9001/
170         if (status == "")
171         {
172             status = "connected";
173             response = "\\\"\\\"";
174         }
175     }
176     else if (urlParams[1] == "restart-explorer") // if the user presses a button "restart explorer" on webpage, Cpp app receives this http command from angular
177     {
178         // http://192.168.0.70:9001/restart-explorer
179         RestartExplorerEx();
180         status = "restarted";
181         response = "\\\"\\\"";
182     }
183     else if (urlParams[1] == "app-info") // This is only used for debug. only developers use this command to test information of running apps.
184     {
185         // http://192.168.0.70:9001/app-info
186     }
187 }
```

Cpp app uses ping instead.

2) Ping

```
http://192.168.0.70:9001/ping
```

Ping is very important http command. Webpage can get the existence of cpp app running by this http command.

And cpp app replies ping command by all running information of the layout.

```

197     char szWindowCaption[0x100] = {0, };
198     GetWindowTextA(info.hWnd, szWindowCaption, sizeof(szWindowCaption));
199
200     char szMsg[0x200];
201     sprintf(szMsg, "window 0x%08X, caption=[%s], process_id=%d", (unsigned long)info.hWnd, szWindowCaption, info.dwProcessId);
202
203     response = szMsg;
204
205 } else if (urlParams[1] == "ping") // angular webpage sends this http command every 10 seconds(I am not sure). Cpp replies with the current playing information
206 {
207     // http://192.168.0.70:9001/ping
208     if (GetTotalPresentationCount())
209     {
210         if (presentationRunning && PermittedRunning())
211         {
212             dm_presentation* pdp = GetPresentationData(gPresentationIdPlaying);
213
214             if (pdp == NULL)
215             {
216                 status = "idle";
217                 response = "\\No presentation found\\";
218             }
219             else
220             {
221                 char layoutbf[33];
222                 _itoa(gnLayoutPlaying, layoutbf, 10);
223                 // before fully loaded, Cpp replies with "loading"
224                 // after fully loaded, Cpp replies with "playing"
225             }
226         }
227     }
228 }

```

If count of total presentations is zero, it returns “idle”.

```

197     char szWindowCaption[0x100] = {0, };
198     GetWindowTextA(info.hWnd, szWindowCaption, sizeof(szWindowCaption));
199
200     char szMsg[0x200];
201     sprintf(szMsg, "window 0x%08X, caption=[%s], process_id=%d", (unsigned long)info.hWnd, szWindowCaption, info.dwProcessId);
202
203     response = szMsg;
204
205 } else if (urlParams[1] == "ping") // angular webpage sends this http command every 10 seconds(I am not sure). Cpp replies with the current playing information
206 {
207     // http://192.168.0.70:9001/ping
208     if (GetTotalPresentationCount())
209     {
210         if (presentationRunning && PermittedRunning())
211         {
212             dm_presentation* pdp = GetPresentationData(gPresentationIdPlaying);
213
214             if (pdp == NULL)
215             {
216                 status = "idle";
217                 response = "\\No presentation found\\";
218             }
219             else
220             {
221                 char layoutbf[33];
222                 _itoa(gnLayoutPlaying, layoutbf, 10);
223                 // before fully loaded, Cpp replies with "loading"
224                 // after fully loaded, Cpp replies with "playing"
225             }
226         }
227     }
228 }

```

- *When is total count of presentations zero?*

When you want to clean all running environment and start from empty data, then you delete file.json and run cpp app.

Cpp app loads file.json content at its startup so it will load nothing.

The total count of presentations is zero.

But to play a presentation on webpage, it should publish a presentation first and count of presentations becomes at least 1. So this case only happens on error.

To prevent errors while replying ping command, this relation is needed.

- *When is active ping-reply information delivered to webpage?*

As mentioned above, total count of presentations should not be zero.

And active ping-reply information could be delivered when it is playing a presentation with master right.

```

200     char szMsg[0x200];
201     sprintf(szMsg, "window 0x%08X, caption=[%s], process_id=%d", (unsigned long)info.hWnd, szWindowCaption, info.dwProcessId);
202
203     response = szMsg;
204 }
205
206 else if (urlParams[1] == "ping") // angular webpage sends this http command every 10 seconds(I am not sure). Cpp replies with the current playing info
207 {
208     // http://192.168.0.70:9001/ping
209     if (GetTotalPresentationCount())
210     {
211         if (presentationRunning & PermittedRunning())
212         {
213             dm_presentation* pdp = GetPresentationData(gPresentationIdPlaying);
214
215             if (pdp == NULL)
216             {
217                 status = "idle";
218                 response = "\\"No presentation found\\\"";
219             }
220             else
221             {
222                 char layoutbf[33];
223                 _itoa(gLayoutPlaying, layoutbf, 10);
224                 // before fully loaded, Cpp replies with "loading"
225                 // after fully loaded, Cpp replies with "playing"
226                 if (loadingLayoutStatus == LOADING_STARTED)
227                     status = "loading";
228                 else
229                 {
230                     char layoutbf[33];
231                     _itoa(gLayoutPlaying, layoutbf, 10);
232                     // before fully loaded, Cpp replies with "loading"
233                     // after fully loaded, Cpp replies with "playing"
234                     if (loadingLayoutStatus == LOADING_STARTED)
235                         status = "loading";
236                     else
237                         status = "playing";
238
239             }
240
241             response = ("\\computer_id\\": " + to_string(pdp->computer_id) + "\\"
242             ", \\presentation_id\\": " + to_string(gPresentationIdPlaying) + "\\"
243             ", \\layout\\": " + layoutbf +
244             ", \\info\\": "[// presentations[gPresentationPlaying][\"info\"][[gLayoutPlaying][\"windows\"]].dump() + "
245             FreePresentationData(pdp);
246
247             // ping reply data should contain the live data of currently running apps.
248         }
249     }
250 }

```

When any kind of error such as presentation data loading error happens, it will not send active ping-reply command to the webpage.

- When is reply status “loading” or “playing”?

When Cpp app receives ping command while it is loading a layout and not fully loaded, it replies with “loading” instead of “playing”.

```

211     {
212         dm_presentation* pdp = GetPresentationData(gPresentationIdPlaying);
213
214         if (pdp == NULL)
215         {
216             status = "idle";
217             response = "\\"No presentation found\\\"";
218         }
219         else
220         {
221             char layoutbf[33];
222             _itoa(gLayoutPlaying, layoutbf, 10);
223             // before fully loaded, Cpp replies with "loading"
224             // after fully loaded, Cpp replies with "playing"
225             if (loadingLayoutStatus == LOADING_STARTED)
226                 status = "loading";
227             else
228                 status = "playing";
229
230             response = ("\\computer_id\\": " + to_string(pdp->computer_id) + "\\"
231             ", \\presentation_id\\": " + to_string(gPresentationIdPlaying) + "\\"
232             ", \\layout\\": " + layoutbf +
233             ", \\info\\": "[// presentations[gPresentationPlaying][\"info\"][[gLayoutPlaying][\"windows\"]].dump() + "
234             FreePresentationData(pdp);
235
236             // ping reply data should contain the live data of currently running apps.
237         }
238     }
239 }

```

As you can see, loadingLayoutStatus variable shows the status of loading layout.

Before loading all channel windows, it is set to “LOADING_STARTED”.

```

http_handler.cpp
StartLayout -> void StartLayout(int nPresid, int nLayoutIdx)
397     startTime = GetCurTime();
398
399     /* Launch all exe apps containing 'presentation', 'layout' */
400     TraceLog("StartLayout: %d/%d %d\n", nPresid, nLayoutIdx, totalWin);
401
402     // set loading status to one that represents the layout is starting...
403     loadingLayoutStatus = LOADING_STARTED;
404     bool bClosedOldApps = false;
405
406     // load every channel in a layout
407     // if the loading is interrupted (loadingLayoutInterrupted becomes true), then stops loading channels.
408     for (int x = 0; x < totalWin && !loadingLayoutInterrupted; x++)
409     {
410         if (presentationRunning)
411         {
412             dm_window& dw = pdl->pplate[0].channels[x].windows[0];
413
414             if (dw.app_location && dw.app_location[0] != '\0')
415             {
416                 MatchChannelX1WindowHandle();
417                 PositionOpenChannels();
418                 StartChannel(nPresid, nLayoutIdx, x, NULL);
419             }
420
421             // at the moment of 20% of loading, close old channel windows.
422             if (!bClosedOldApps)
423             {
424                 if (x * 100 > totalWin * 20)
425                 {
426                     CloseAllBackupRunningApps();
427                 }
428             }
429         }
430     }
431
432     // UpdateTimeAllRunningApps();
433
434     if (!bClosedOldApps) // if not closed old channel windows, close them all.
435     {
436         CloseAllBackupRunningApps();
437     }
438
439     loadingLayoutInterrupted = false;
440     loadingLayoutStatus = LOADING_FINISHED;
441 }

```

After fully loaded all channel windows, it is set to “LOADING_FINISHED”.

```

http_handler.cpp
StartLayout -> void StartLayout(int nPresid, int nLayoutIdx)
440     UpdateTimeAllRunningApps();
441
442     if (!bClosedOldApps) // if not closed old channel windows, close them all.
443     {
444         CloseAllBackupRunningApps();
445     }
446
447     loadingLayoutInterrupted = false;
448     loadingLayoutStatus = LOADING_FINISHED;
449 }

At the end of the function. It means that all channels are fully loaded.

```

http command, and playing a presentation proceed concurrently(multithreading).

So we have to introduce some specific synchronization mechanism here.

Cpp app solved this issue by loadingLayoutStatus variable.

When cpp app receives “ping” command and loadingLayoutStatus is “LOADING_STARTED”, it replies with “loading” and when loadingLayoutStatus is “LOADING_FINISHED”, it replies with “playing”.

- *Cpp app replies “ping” command by active running information. What is active running information?*

Webpage configures presentation data and publishes it to cpp app.

But as related to closing some channels or move/resize action, its geometry changes by time and even some channels could be deleted.

So cpp app replies with information of only running channels(not closed ones).

```

236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285

```

displaymanager
FreePresentationData(pdp);
// ping reply data should contain the live data of currently running apps.
mutex_locker<raw_mutex> locker(app_mutex);
string strInfo;
int k;
for (k = 0; k < (int)RunningApps.size(); k++)
{
 AppInfo& info = RunningApps[k];

 if ((info.nPrid != gPresentationIdPlaying || info.nLayoutId != gnLayoutPlaying) || ::IsWindow(info.hWind))
 continue;

 info.data["x"] = info.x;
 info.data["y"] = info.y;
 info.data["width"] = info.width;
 info.data["height"] = info.height;
 info.data["channel_no"] = info.n channelId;

 string strChannelText = info.data.dump();
 strInfo += strChannelText + ":";

 if (strInfo.length() > 0)
 strInfo = strInfo.substr(0, strInfo.length() - 1);

 response += strInfo;
}

Originally configured presentation data is attached to at the starting channels.

But at the moment of receiving ping command, it updates information with live geometry.

Ping-reply text will contain live geometry of all running channels.

- *Is there a way to check ping-reply text in cpp app?*

After making ping-reply text, we rarely have to check its validity of json format.

So I added some code for it and now disabled it.

If you enable the checking part, when an error happens, you will see “ping_error.json” file created in the cpp app folder.

```

257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285

```

displaymanager
strInfo += strChannelText + ":";
if (strInfo.length() > 0)
 strInfo = strInfo.substr(0, strInfo.length() - 1);

response += strInfo;
response += "}"
// the following lines used for debugging.
#if 0
 json resData;
 try
 {
 resData = json::parse(response);
 }
 catch (json::parse_error)&
 {
 if (1)
 {
 string strJson = getexecfolder() + "\\ping_error.json";
 FILE* fp = fopen(strJson.c_str(), "wb");
 if (fp)
 {
 fwrite(response.c_str(), 1, response.length(), fp);
 fclose(fp);
 }
 }
 }
#endif

Please make the code line to

```
#if 1
```

After you checking ping-reply text, please make it to original one.

```
#if 0
```

3) Play/Play-layout

```
http://192.168.0.70:9001/play/presentation_id/layout_index/owner
```

```
http://192.168.0.70:9001/play-layout/presentation_id/layout_index/owner
```

“play” http command is similar to “play-layout” command.

Cpp app does not use “play” command.

When a user presses “Play Layout 1”, then cpp app receives “play-layout” command.

So here we explain about “play-layout”.

This http command only works if the presentation exists (published already), and cpp app works as a master.

```
364     }
365     else if (urlParams[1] == "play-layout" && urlParams.size() > 4) // this http command is received when a user press "Play Layout x"
366     {
367         // http://192.168.0.70:9001/play-layout/presentation_id/layout_index/owner
368         if (
369             #ifdef MULTIMASTER_ACCESS
370             1,
371         #endif
372         ownership == "" || ownership == urlParams[4]),
373         if (PermittedRunning())
374         {
375             if (gbPresDataPending)
376             {
377                 gbPresDataPending = false;
378                 LoadPresentationDataFromFile();
379             }
380         }
381     }
382     int presentationId = atoi(urlParams[1].c_str()); // Presentation data exists (already published)
383     dm_presentation* pdp = GetPresentationData(presentationId);
384     BOOL prefound = (pdp == NULL) ? FALSE : TRUE;
385     FreePresentationData(pdp);
386     if (prefound) // if this presentation is published from angular.
387     {
388         InterruptLoadingPresentation();
389     }
390 }
```

Here the remarkable part is loading presentation data if pending.

```

364     }
365     else if (urlParams[1] == "play-layout" && urlParams.size() > 4) // this http command is received when a user press "Play Layout x"
366     {
367         // http://192.168.0.78:9001/play-layout/presentation_id/layout_index/owner
368         if (defined MULTIUSER_ACCESS)
369         {
370             if (ownership == "" || ownership == urlParams[4]) // if presentation data is pending to be loaded, load it.
371             {
372                 if (PermittedRunning())
373                 {
374                     if (gbPresDataPending)
375                     {
376                         gbPresDataPending = false;
377                         LoadPresentationDataFromFile();
378                     }
379                 }
380             }
381         }
382     }
383     int presentationId = atoi(urlParams[2].c_str());
384     dm_presentation* pdp = GetPresentationData(presentationId);
385     BOOL prefound = (pdp == NULL) ? FALSE : TRUE;
386     FreePresentationData(pdp);
387     if (prefound) // if this presentation is published from angular.
388     {
389         InterruptLoadingPresentation();
390     }
391 }

```

VA Find References Results

- TriggerLoadingPresentation (47): while (loadingLayoutStatus == LOADING_FINISHED) // wait until fully loaded
- InterruptLoadingPresentation (60): if (loadingLayoutStatus == LOADING_STARTED)
- StopLoading (37): while (loadingLayoutStatus == LOADING_INIT)
- StartLayout (403): loadingLayoutStatus = LOADING_STARTED;
- StartLayout (440): loadingLayoutStatus = LOADING_FINISHED;
- C:\Users\Administrator\Desktop\DOCS\MIQUEL\c+ +NEVER\displaymanager\appmanager\appmanager.cpp
 - MatchChannelWindowHandle (68): if (loadingLayoutStatus == LOADING_STARTED)
- C:\Users\Administrator\Desktop\DOCS\MIQUEL\c+ +NEVER\displaymanager\playback\main_process.cpp
 - MainThread (414): loadingLayoutStatus = LOADING_INIT;
 - InnerMainThread (304): loadingLayoutStatus = LOADING_INIT;
 - InnerMainThread (302): loadingLayoutStatus = LOADING_INIT;

For further information, please see “Unpublish” or “Get-presentation section”.

At first, cpp app checks if the presentation exists in presentation data.

```

375     if (PermittedRunning())
376     {
377         if (gbPresDataPending)
378         {
379             gbPresDataPending = false;
380             LoadPresent(); // if the presentation exists (already published)
381         }
382     }
383     int presentationId = atoi(urlParams[2].c_str());
384     dm_presentation* pdp = GetPresentationData(presentationId);
385     BOOL prefound = (pdp == NULL) ? FALSE : TRUE;
386     FreePresentationData(pdp);
387     if (prefound) // if this presentation is published from angular.
388     {
389         InterruptLoadingPresentation(); // Interrupt if layout is not fully loaded.
390         if (ownership != urlParams[4])
391             TraceLog("play-layout:ownership (%s->%s)\n", ownership.c_str(), urlParams[4].c_str());
392         ownership = urlParams[4];
393         layoutPlay = TRUE;
394         int layoutNo = atoi(urlParams[3].c_str());
395         gHTTPRequestPresId = presentationId;
396         gHTTPRequestLayoutId = layoutNo - 1;
397         bHTTPRequestDontCloseOldLayout = false;
398     }
399 }

```

4) Unpublish

5) Schedule

It is not used in this project.

It is made by old dev, but we do not use it here.

6) File-upload

On file-upload, there is a problem while processing URL strings.

For example, if the file name is “不 tt.mp4”, angular send the URL to cpp app like

```
%E4%B8%8D%20tt.mp4
```

Offset	0	1	2	3	4	5	6	7	8	A	B	C	D	E	F
00000000	E4	B8	8D	20	74	74	2E	6D	70	34					

So cpp app receives “%E4%B8%8D%20tt.mp4” from webpage for “不 tt.mp4”.

Cpp app should interpret the URL string to real one.

Now Eric did not agree to interpret all the non-ASCII characters so that cpp app is now interpreting only ‘%20’ character to ‘ ’.

Cpp app will work in case of “ATT%20see.mov” to “ATT see.mov”.

But will not work in case of “%E4%B8%8D%20tt.mp4” because cpp app has no interpretation engine for non-ASCII characters.

I did interpretation for non-ASCII characters, but he disliked so I removed again... pity.

Exact code part here (screenshot)

- *What is the solution for file name to upload?*

So the solution for this problem now is to upload files of name with only ASCII-characters + space character.

The available file name includes

English lower/upper case characters
Digits
~`!@#\$%^&()=+-_

7) get-presentation

this http command is dispatched to cpp app from angular webpage when a user “publish”es a presentation on webpage.

At first angular webpage sends cpp app the URL only which contains the presentation id.

- *What are “publish” steps between angular and cpp app?*

This has the most complex steps in this cpp project.

First, a user presses “publish” button on webpage.

Second, webpage sends <http://192.168.0.70:9001/get-presentation/225> to cpp app. Here port number 9001 is hardcoded in cpp code and angular code.

Third, cpp app interprets the URL and get the presentation number from it. It is 225 here. And it sends the data request to a webserver by curl.

webserver URL is configured in settings.ini.

For example, 192.168.0.71:5010.

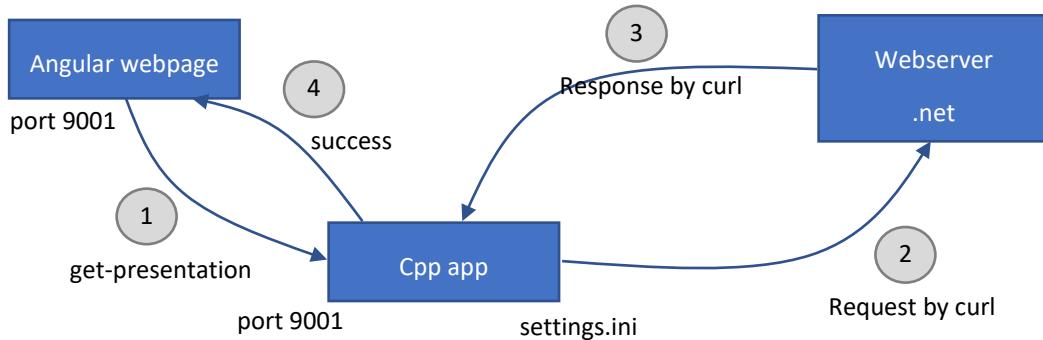
Pay attention to a webserver. It is not angular webpage. It is a .net server.

Fourth, .net server transmits presentation data on curl requested by cpp app.

Fifth, cpp app receives presentation data from a webserver by curl.

After receiving presentation data from a .net server, cpp app replies to a angular webpage with “success”.

If it fails to receive from a .net server, it replies with “fail”.



8) View-control

This http command is received when a user presses a button on angular webpage “live-feed”, or “kontrol”.

When this command is received, cpp app brings the app selected on top of other apps.

9) Section-control

This category has 4 http commands “max”, “def”, “close”, “app/flag”.

This http command is received when a user presses a button on angular webpage “live-feed”, or “kontrol”.

10) Stop-presentation

- 11) Close-app
- 12) Restart-explorer
- 13) Shutdown
- 14) Get-device-info
- 15) sendEvent
- 16) sendNotice
- 17) thumbnail-quality
- 18) picture/1

5. Synchronization of presentation data

- 1) Retrieval from webserver
- 2) Saving on local storage
- 3) Interacting with playing presentation

6. Playing/stopping layouts

- 1) Play presentation
 - *Webpage command*
 - *Sequencer automation*
 - *Scheduler automation*
- 2) Play layout
 - *Rewinding*
 - *Overlay of layouts*
- 3) Play channel
 - *Launching an external app*
 - *Recovering app window handle*
- 4) Stopping presentation
- 5) Closing a specified window.

7. Move/resize by web users

- 1) Creation of a websocket server
- 2) move/resize command process thread

8. Thumbnails for web users

- 1) Creation of a websocket server
- 2) Capturing thumbnail images
- 3) Thumbnail quality

9. Basic features for running application windows

- 1) Location of window geometry
- 2) Slicing window
- 3) Full-screen/maximize
- 4) Replacing app window

10. Redundancy

- 1) Algorithm
- 2) Creation
- 3) switchover

11. RS232 commands for signal input

- 1) Command file format
- 2) Transmit command to RS232

12. Log

- 1) Log file path
- 2) Logging

13. Startup windows OS.

- 1) Registry setting
- 2) Launching presentation at start up of windows OS.

14. Splash screen

15. Additional FAQs:

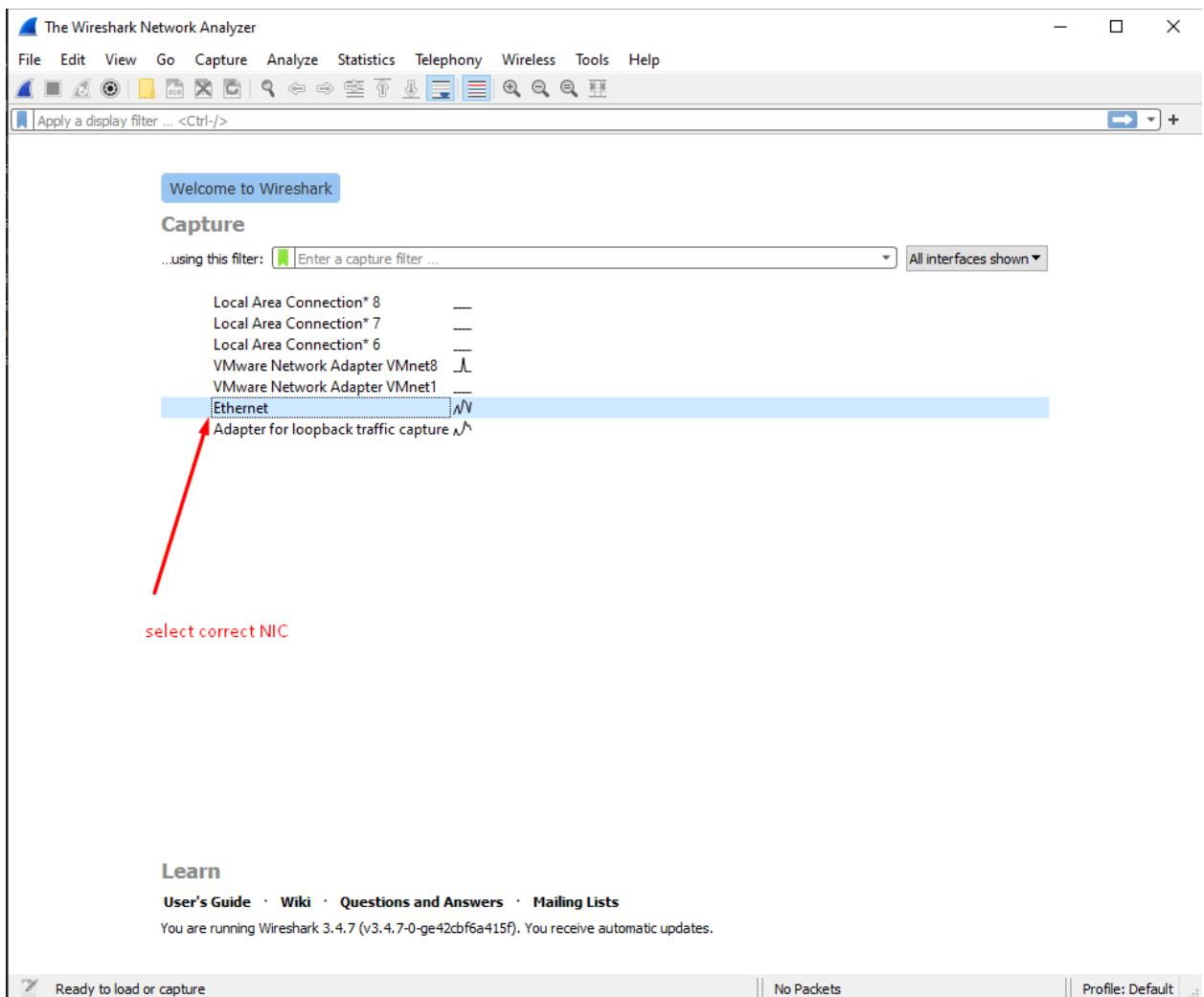
- 1) How can we check the TCP connection is normal or broken?

You can check it by wireshark.

<https://www.wireshark.org/download.html>

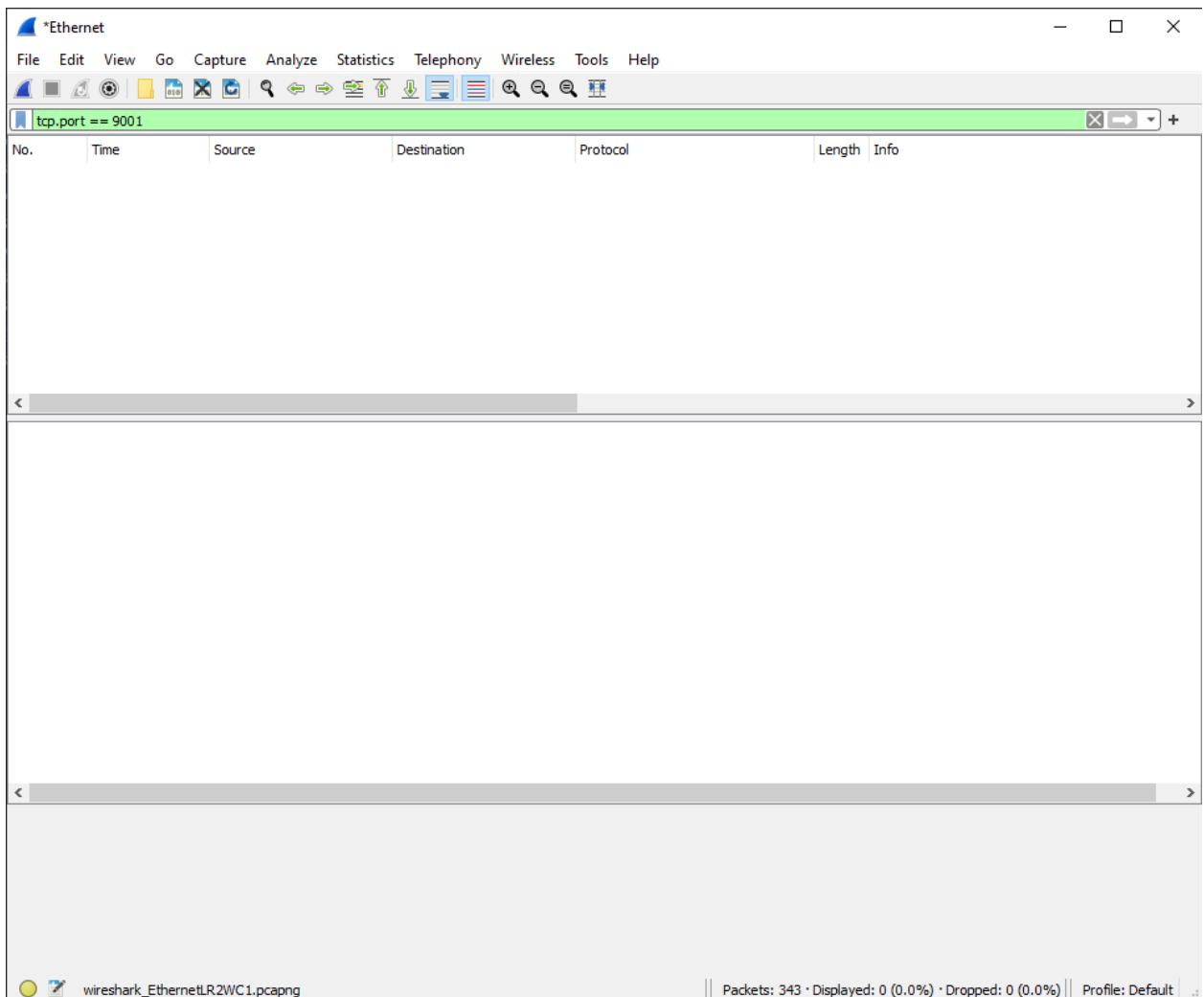
After installing wireshark for windows, please run it.

Select correct NIC and double click it.



Let's assume that a user runs angular webpage at 192.168.0.71 and cpp app runs at 192.168.0.70.

Angular webpage sends http commands by port 9001. So we can check it by filtering $\text{tcp.port} == 9001$



If you can see any packets there and source IP, destination IP are the ones of cpp app and angular webpage, TCP connection is working.

Let's assume that you configured move/resize+KVM port 6001, and thumbnail port 6002.

To check the port 6001, or 6002, you can input in filter box

```
tcp.port == 6001
```

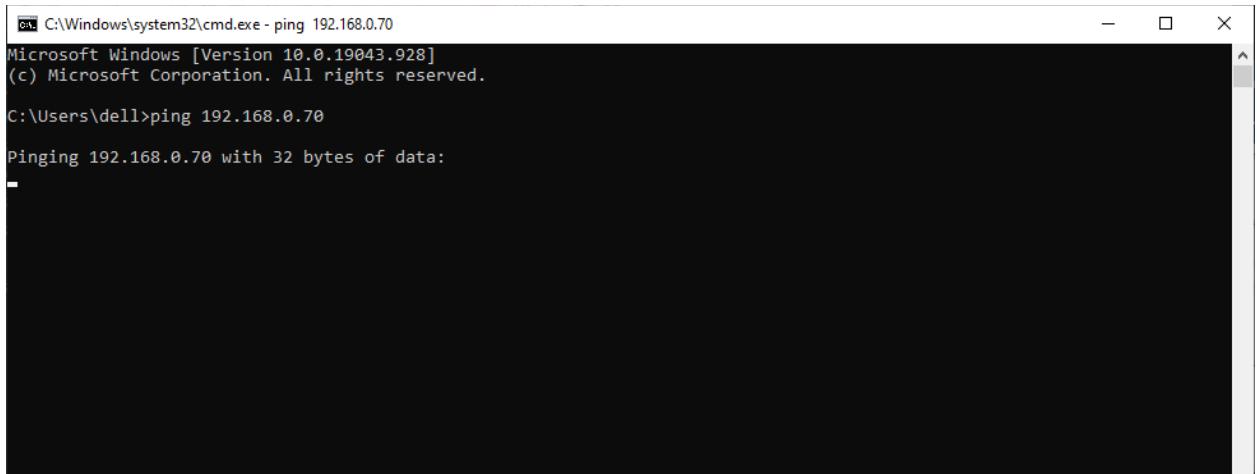
```
tcp.port == 6002
```

```
...
```

You can check source IP, and destination IP for your cpp app and .net app or angular webpage.

If nothing shows, the connection is not existing.

So you should check the connection state by “ping” command.



```
C:\Windows\system32\cmd.exe - ping 192.168.0.70
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dell>ping 192.168.0.70

Pinging 192.168.0.70 with 32 bytes of data:
```

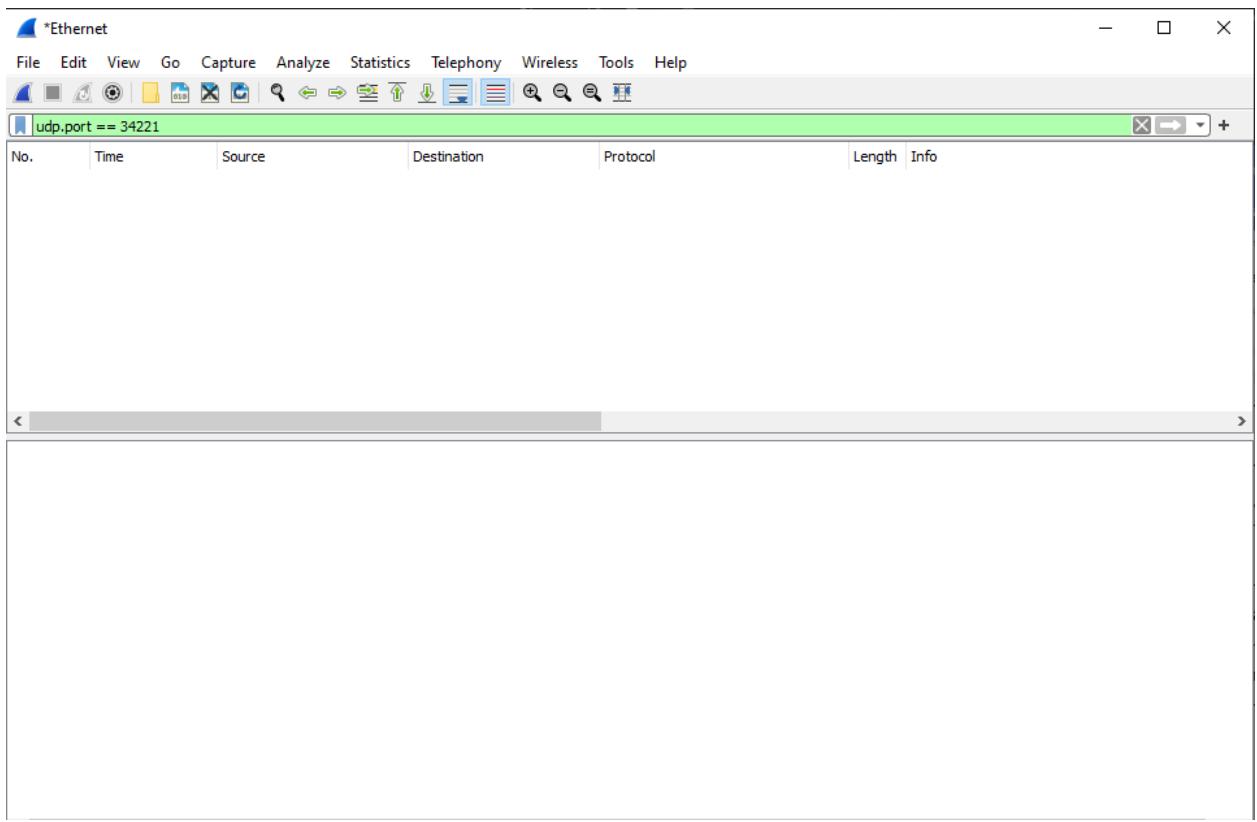
If “ping” works good, you should check settings.ini and webpage working.

In our cpp project, there are 4 types of TCP ports and 1 UDP port.

Connection	Port	Mission	Modification
TCP	9001	Angular webpage	Can not modify
TCP	6001	Move/resize + KVM websocket	Configured in settings.ini
TCP	6002	Thumbnail websocket	Configured in settings.ini
TCP	5010	.net webserver for presentation data request	Configured in settings.ini
UDP	54332	Redundancy UDP communication	Written in CMateComm.cpp

To check the UDP port, input in filter box like

```
udp.port == 34221
```



Also here, if some packets are shown, it works good. But don't forget to check the source IP and destination IP.

If nothing is shown, no communication is working here.