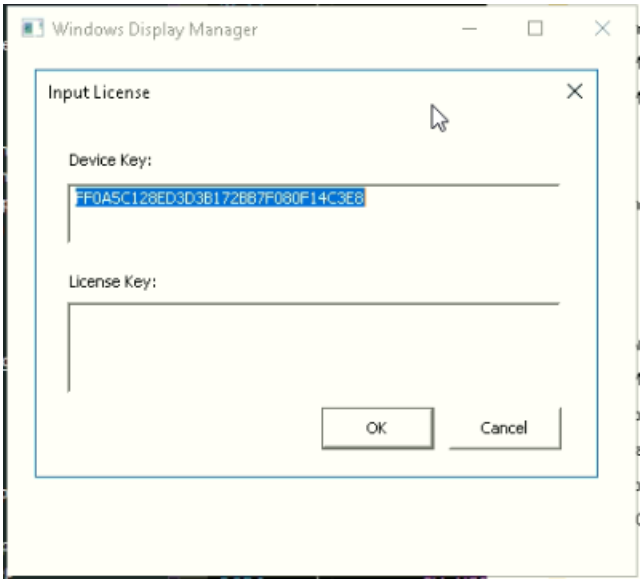
DisplayManager Licensing Manual

1. Getting activation code

- User manual

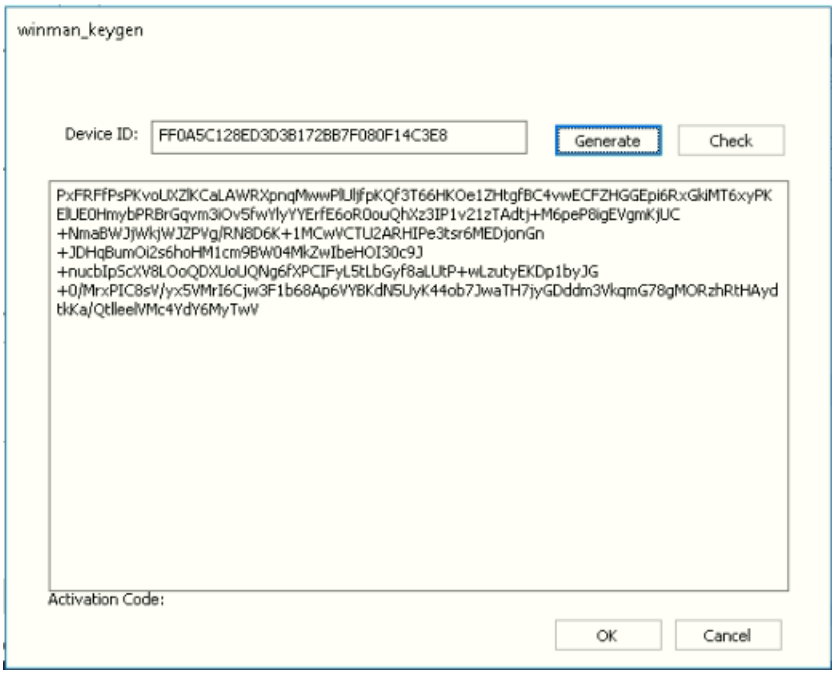
Unlicensed user should get an valid activation code from a provider(it's you Eric).

When unlicensed user runs "displaymanager.exe", then a dialog box appears on the main window of "displaymanager.exe"



Then user copies the Device Key, and sends it to the provider(Eric).

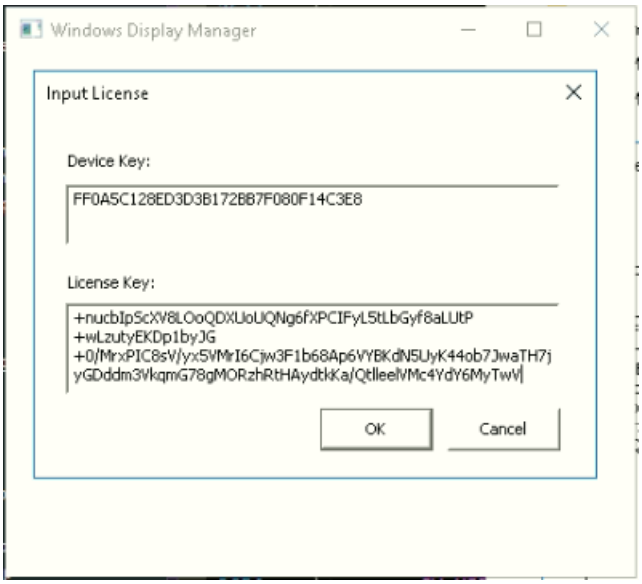
Then the provider(Eric) generates a valid activation code for the Device Key by using the following tools.



winman_keygen_30days.exe	Generate a 30-days activation code.
winman_keygen_full.exe	Generate a full activation code.

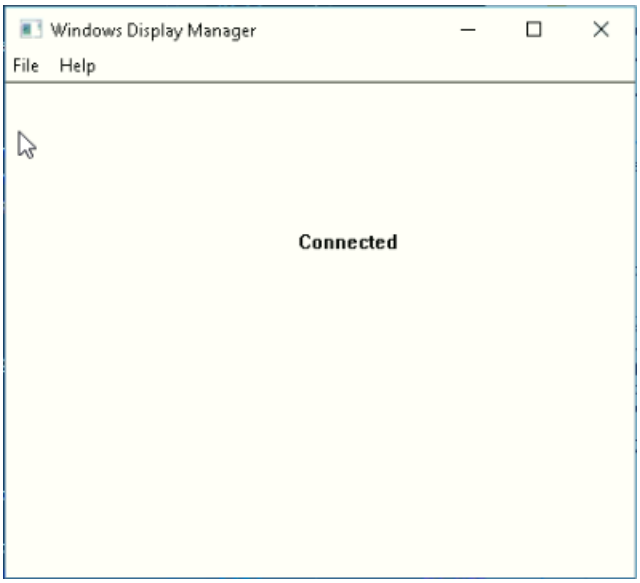
The provider(Eric) copies the proper activation code in the output box of the generator, and sends it back to the user.

Unlicensed user inputs the activation code received into the license dialog box and presses “OK” button.



“displaymanager.exe” checks the validity of the activation code input.

If it is valid, then the dialog box disappears and the main window of “displaymanager.exe” goes on.



Otherwise, the license dialog box continues to be on the main window of “displaymanager.exe”.

The checking validation is mentioned at the next section.

- The structure of activation code

It is already discussed with the provider(Eric).

Plain text = Device hardcodes + Sequence + Issued at + Period + Flags
Signature text = Plain text + CRC16 + RSA2048-signature(private key)
Activation code = Base64(Blowfish(Signature text))

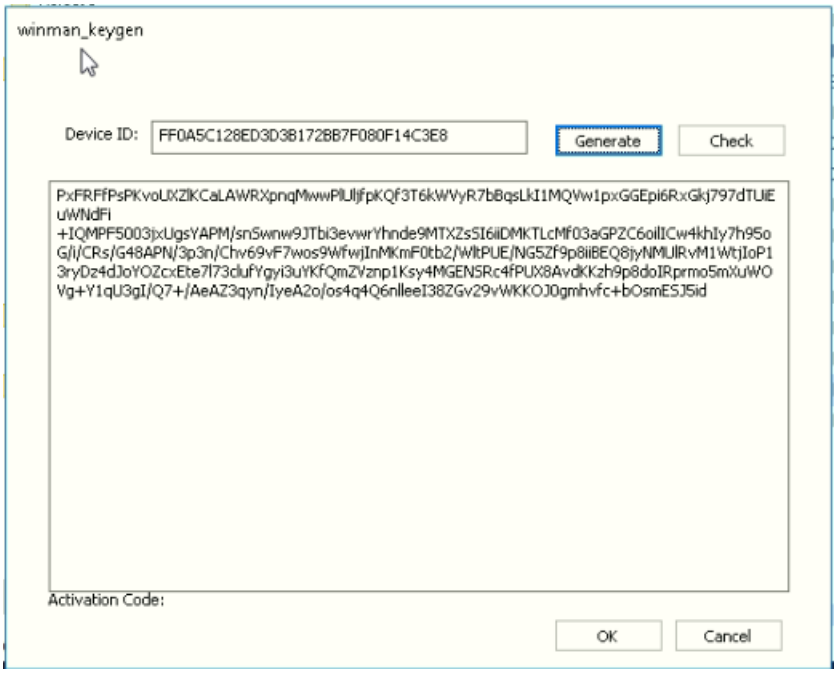
- Activation code validation for a provider

If a user got activation code and some errors occurred, then he would ask a provider for its validity.

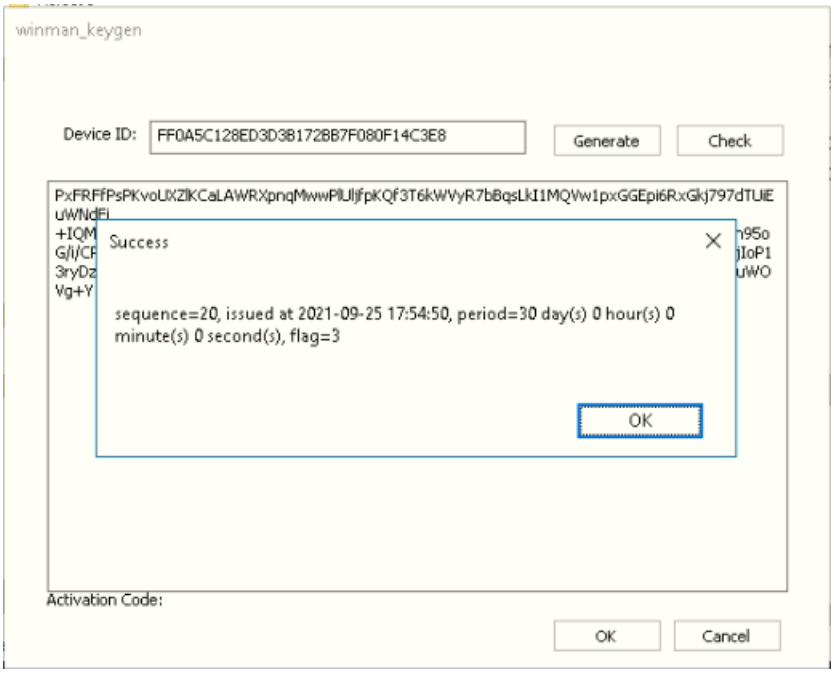
i.e. there will be occasions that some users inquire their activation code is correct and when was it published.

Then the provider(Eric) could check the validity of the activation code by winman_keygen.exe

First, input Device ID and activation code in the text boxes.



Second, press the button “Check” to see additional information about the activation code.



Here the meaning of each field is as follows.

Sequence: whenever provider generates new activation code, sequence increment by 1. So this number shows by which order this activation code was generated. In the picture, this activation code was generated as 20nd activation code.

Issued at: this is the timestamp on generating the activation code. In the picture, this activation code was generated at 2021-09-25 17:54:50 i.e. 5:54:50 pm on the 25th of September in 2021.

Period: This is the expiration period of the activation code since ”issued at”. In this picture, it means that the activation code will expire 30 days later since 2021-09-25 17:54:50, i.e at 2021-10-25 17:54:50.

Flag: this show the method of activation. In the picture, the flag is 3. It means that licensing method consists of RSA2048-signature check, hardware lock check, hidden file check. If the provider invents a new licensing method then flag would be some other value.

Now this winman_keygen.exe generates activation code for “displaymanager.exe”, so flag is always 3.

2. Checking a valid activation code

- Validation algorithm

When the user input an activation code on the input dialog box of “displaymanager.exe”, the app checks it as the following steps.

The app processes an activation code by the reverse order of making activation code, except that RSA2048-signature uses public key. (The app doesn’t know the private key.)

If at least one step of the following meets false, then validation fails.

Compare the “Device hardcodes” in Plain text and the current “Device hardcode”. Check if they are identical.
Check if the value “Flags” is the same of the version. “displaymanager.exe” has the value 3 as its “flags”. If the value “Flags” of the “Plain text” is not 3, then it fails.
Check if the current timestamp is in the range of [“Issued at”, “Issued at” + “Period”].

Check if CRC16 hash value for “Plain text” is equal to the “CRC16”.
Check if RSA2048(public key) signature for “Plain text” + “CRC16” is equal to “RSA2048-signature(private key)”
Check if “Device hardcodes” in Windows Registry is the fixed value. It is called “Hardware Lock”.
Check if hidden file “C:\Windows\cirenc.dll” exists and the whole content is the same as the activation code. It is called “Hidden File Lock”.

- Checking while running “displaymanager.exe”

“displaymanager.exe” app checks the validity of the license at least every 10 seconds.

It walks through the validation algorithm, and if it detects the license is invalid, then it initialize data for the license and opens a notification dialog box and exits.

Running again, activation code dialog box will appear.

3. Project code analysis for licensing

1) displaymanager.exe

- Startup check

To walk through licensing system, Cpp app should get the basic data for it. It is device key.

The first routine is WinMain function when Cpp app runs.

You could see the code like below.

```
// Get several physical device components
GetDeviceKey();
```

This is the function to get hardware device information and calculate the hash of them.

And then Cpp app checks if the activation key is valid.

```
// Get license key from the sqlite db
GetLicenseKey();
GetLicenseInformation();
```

Then basic data for licensing and old activation code were loaded in Cpp app.

The routine to check the validity of the activation is as follows.

```
// Check license key with a hash value made from device information
While (!CheckLicenseKey()) {
…
}
```

If you want to remove the startup check of licensing, remove this “while…{}” code block.

- Hidden file check

This check is done in function “CheckEncFile()”

This function checks if C:\Windows\cirenc.dll exists and loads the content to activation code.

If you want to change the file name, then change the macro.

Current	#define ENC_FILENAME “cirenc.dll”
Change to	#define ENC_FILENAME “changedfile.ext”

If you want to change the directory to C:\Windows\System32, then change the code in CheckEncFile().

Current	GetWindowsDirectory(szWindowsPath, sizeof(szWindowsPath));
Change to	GetSystemDirectory(szWindowsPath, sizeof(szWindowsPath));

If you want change the directory to “D:\mydir\file.ext”, then change the code in CheckEncFile().

Current	sprintf(szEncPath, “%s\\%s”, szWindowsPath, ENC_FILENAME)
Change to	Sprintf(szEncPath, “D:\\mydir\\file.ext”);

When Cpp App gets new activation code, then this hidden file is created by SetEncFile() function.

The content of this function is similar to one in CheckEncFile().

If you don’t want to make hidden file check, then always return true in this function like this.

```
bool CheckEncFile()
{
        return false;
        …
}
```

- Hardware lock check

This check is used to restrict running displaymanager.exe on specified machines.

This is done by CheckDeviceInfo() function.

In this function, hash of all device information is calculated.

```
md5.update(po, strlen(po));
```

This line is to add information to hash context.

5 blocks of information is added to hash context. The list and order of the blocks are

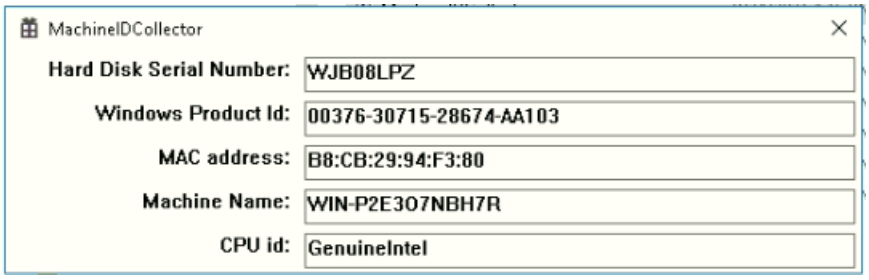
Hard disk serial number
Windows Product ID
MAC address of the first NIC installed on this computer
Computer Name
CPU ID

CheckDeviceInfo() function compares the hash of specified information to the one of Cpp App machine.

If you make this Cpp App on other machine, then you change the code with the information on that machine.

To collect all information of the other machine, please copy MachineIDCollector.exe to that machine and run.

Then the information shows like this.



You change the content of the CheckDeviceInfo() function with this blocks.

After building Cpp App, it can run on that machine.

If you want to make hardware lock check, then always return true in this function like this.

```
bool CheckDeviceInfo()
{
        return true;
        …
}
```

- Periodic license check

Cpp App checks the state of the license periodically. The code is in MainThread() function.

It is a thread for mainly managing presentation but it handles periodic check of license and redundancy data synchronization.

You can find the code in this function below.

```
If (curTime > licTime + 10)
```

If you change the period of license checking, then change the number 10 to 30.

Then the period of checking license will be 30 seconds.

Periodic license checking contains checking expiration routine.

It is CheckLicenseInformation(). This is called in MainThread().

If you want to disable trial version license i.e. any activation code would be full license, then please remove the function call CheckLicenseInformation() like this.

```
// CheckLicenseInformation();
```

- Sqlite db

All licensing information is stored in sqlite db with password/encryption.

The function to open sqlite db file is GetSqliteDbInstance().

Now SQLite db is located at C:\Windows\sysrq1.bin

If you want to change the sqlite db file name, then please change the macro DB_FILENAME.

Current	#define DB_FILENAME “sysrq1.bin”
Change to	#define DB_FILENAME “mylicense.db”

If you want to change the C:\Windows to C:\Windows\System32, then change the content of the function GetSqliteDbInstance().

Current	GetWindowsDirectoryA
Change to	GetSystemDirectoryA

If you want to change the file path, then change the content of the function SqliteDbInstance()

Current	sprint(szDBPath, “%s\\%s”, szSystemPath, DB_FILENAME)
Change to	sprint(szDBPath, “D:\mydir\myfile.db”)

Sqlite db file is encrypted by the password.

The sqlite db password is generated by device id string.

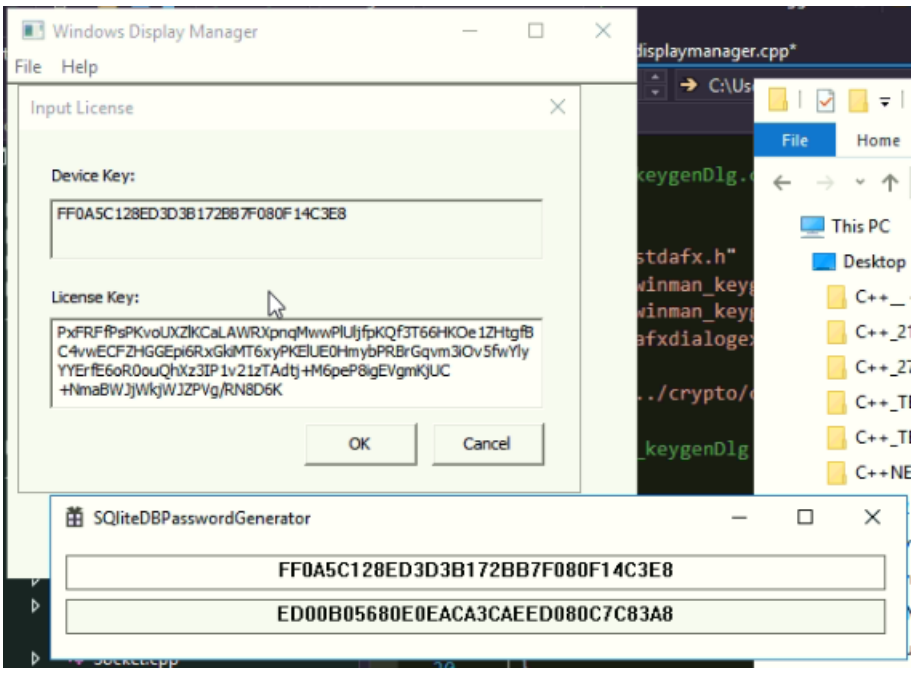
This password is generated by the function GenerateSQLiteDBPassword() in the project code.

So in GetSqliteDBInstance(), there is password generating code as below.

GenerateSQLiteDBPassword(gszDeviceKey, szDBPassword);

If you debug the Cpp App, then you check the content of szDBPassword then you could see the sqlite db file password.

If you want to see the sqlite db file password not in Debug mode at any time, then please run “SQliteDBPasswordGenerator.exe” and input device id string. Then you will see the sqlite password for the machine.



After calling GenerateSQLiteDBPassword(), you could find

```
rc = sqlite3_key(*ppdb, szDBPassword, strlen(szDBPassword));
```

If you manage you sqlite db without password/encryption, then you are easy to do it like commenting or removing.

```
//rc = sqlite3_key(*ppdb, szDBPassword, strlen(szDBPassword));
```

Without password/encryption feature for sqlite db, you sure open sqlite db file C:\Windows\sysrq1.bin file by Navicat, SQLite Expert, or any other database manager to see the content.

Db file content is as below.

textsample varchar(800)	stp int	dtp int
activation code	Reference time for activation	Elapsed time for activation

This db file contains only one record.

If licensing is failed by one of RSA2048-signature check, hardware lock check, hidden file check, then Cpp App deletes sqlite db file.

You can find in bool CreateLicenseDb(const char* lic_code)

```
If (lic_code[0] == '\0')
        Return GetSqliteDbInstance(&db, true) != FALSE;
```

This code deletes sqlite db file from storage. So if you want not to delete sqlite db file when failure of activation, then you could see old sqlite db file because it is not deleted.

2) Winman_keygen.exe

This is an executive for only a provider.

So the project code is private.

There are 2 parts, Generating and Checking.

In winman_keygenDlg.cpp file, you can find the function

```
void Cwinman_keygenDlg::OnBnClickedGenerate()
```

This function is called when you click “Generate” button.

In OnBnClickedGenerate(), you can see like this.

```
activate_code = crypto_keygen(dev_info, devinfo_len, dwSeq, qwIssuedAt, dwPeriod, dwFlags, &privkey);
```

I added comments for using this crypto_keygen() parameters. So you add some code before calling it.

In winman_keygenDlg.cpp, you can find the function

```
void Cwinman_keygenDlg::OnBnClickedCheck()
```

This function checks if the activation code given in the textbox is valid for the device id string given.

The checking is done by the code as below.

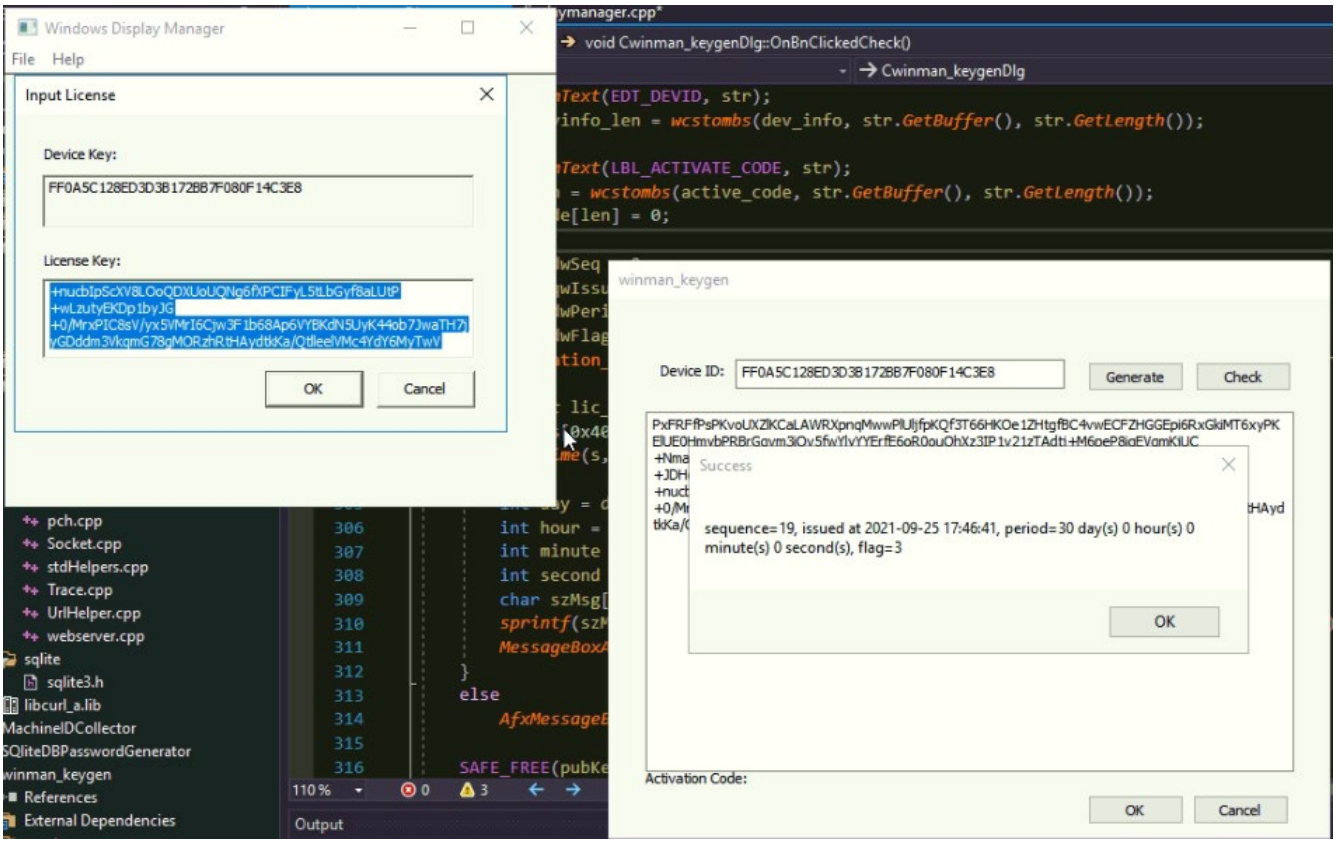
```
If (activation_checkout(active_code, dev_info, devinfo_len, &dwSeq, &qwIssuedAt, &dwPeriod, &dwFlag, &pubKey))
```

activation_checkout function primarily checks if the active_code is valid for dev_info, and takes out activation code parameters like sequence(dwSeq), issued at(qwIssuedAt), expiration period(dwPeriod), flag(dwFlag).

And the following code show you the information by MessageBox.

```
sprintf(szMsg, “sequence=%d, issued at %s, ….);
MessageBoxA(GetSafeHwnd(), szMsg, “Success”, MB_OK);
```

The result is like this picture.



Here the meanings of the parameter is as follows.

sequence: this activation code was generated as 19th one.

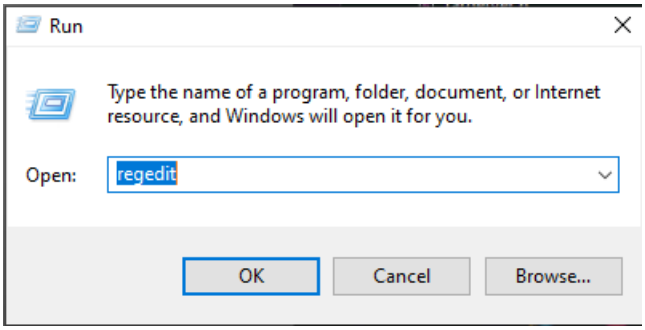
issued at: this activation code was generated at 2021-09-25 17:46:41.

period: this activation will be expired 30 days since 2021-09-25 17:46:41.

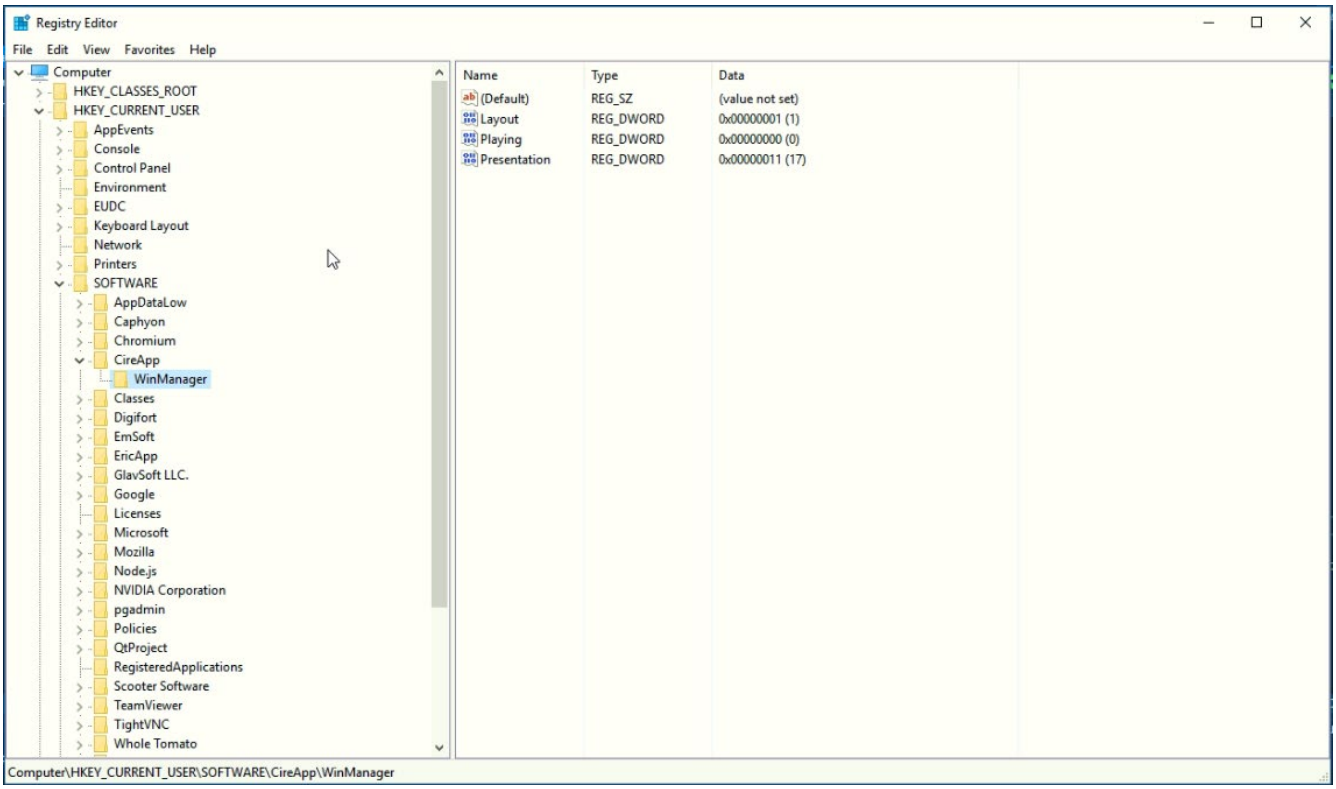
flag: it means 3 so that this activation code is for “displaymanager.exe” which uses RSA2048-signature, hardware lock check, hidden file check.

3) Registry Items

Please press Win+R key to open command window and type “regedit”.



And open the key “HKEY_CURRENT_USER/SOFTWARE/CireApp/WinManager” then you could see 3 items in it.



The meaning of these items are as follows.

Presentation: 0-based presentation number of last played presentation. It is not presentation “id” in json data. It is array index in json data. It is recommended not to modify this value. Here the last played presentation is 18th presentation in json data.(17 is from 0, so counting from 1, 17 means 18th.)

Layout: 0-based layout number of the presentation. Here 1 means the second layout of the presentation.

Playing: it can be either of 0 or 1. If it is 0, then just before closing Cpp App, there was no playing presentation. If it is 1, then there was playing presentation just before closing Cpp App. If you press “stop presentation” on webpage or never press “play layout”, then this value is recorded as 0, If you press “play-layout” on webpage, then it is 1.

In the project code, the reading these values is in function GetLastLayout(). GetLastLayout() is called in WinMain() when starting up Cpp App.

Writing these values is in function SetLastLayout(). It is called when some actions were taken from webpage user.

If you want not to play presentation when newly running displaymanager.exe, then please modify the value “Playing” to be 0 in registry items.

Then you click “displaymanager.exe” on explorer, but no playing presentation is expected.

But if you wanna play presentation just after running “displaymanager.exe”, then modify “Playing” item to be 1 in registry.