

외부 라이브러리

- 라이브러리 요약

1. DLL

1.1 Code

1.1.1 DllMain.cpp

```
#include "pch.h"

static HANDLE s_hInstance = NULL;

#ifdef _MSC_VER
#define WIN32_LEAN_AND_MEAN
#include <Windows.h>
/////////////////////////////////////////////////////////////////
BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    s_hInstance = (HANDLE)hModule;

    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}
#else
/////////////////////////////////////////////////////////////////
__attribute__((constructor)) static void on_load(void)
```

```

{
    s_hInstance = (HANDLE)on_load;
}

////////////////////////////////////
__attribute__((destructor)) static void on_unloaded(void)
{
}
#endif

```

1.1.2 DLL_REPRESENTATION_HEAD.h

```

#pragma once

#include "APIEntry.h"

```

1.1.3 APIEntry.h

```

#pragma once

#ifndef API_IMPORTS
#define API_EXPORTS
#endif

#ifdef _MSC_VER
#ifdef API_EXPORTS
#define API_INTERFACE extern "C" __declspec(dllexport)
#else
#define API_INTERFACE extern "C" __declspec(dllimport)
#endif
#else
#define API_INTERFACE extern "C"
#endif

API_INTERFACE int Add(int& nFir, int&

```

1.1.4 APIEntry.cpp

```
#include "pch.h"
#include "APIEntry.h"

API_INTERFACE int Add(int& nFir, int& nSec)
{
    return nFir + nSec;
}
```

1.1.5 암시적 DLL 링크

```
#include "pch.h"

#include "../DLL_Library/DLL_REPRESENTATION_HEAD.h"
#pragma comment(lib, "DLL_Library.lib")

int main()
{
    int nFir = 10;
    int nSec = 20;

    core::Log_Info(TEXT("%d + %d = %d"), nFir, nSec, Add(nFir, nSec));

    system("pause");
    return 0;
}
```

1.1.6 명시적 DLL 링크

```
#include "pch.h"

#include "../DLL_Library/DLL_REPRESENTATION_HEAD.h"
typedef int(*pFuncPointer)(int&, int&);
```

```
pFuncPointer pFuncPt = NULL;

int main()
{
    int nFir = 10;
    int nSec = 20;

    HANDLE hLib = NULL;
    hLib = core::LoadLibrary(TEXT("DLL_Library.dll"));
    if (NULL == hLib)
    {
        core::Log_Error(TEXT("DLL Load Failed!"));
        return -1;
    }

    pFuncPt = (pFuncPointer)core::GetProcAddress(hLib, "Add");
    if (NULL == pFuncPt)
    {
        core::Log_Error(TEXT("Func Load Failed!"));
        return -1;
    }

    core::Log_Info(TEXT("Explicit Linking"));
    core::Log_Info(TEXT("%d + %d = %d"), nFir, nSec, pFuncPt(nFir, nSec));

    core::FreeLibrary(hLib);

    system("pause");
    return 0;
}
```
