Dynamic Linking library

DLL 생성 및 활용

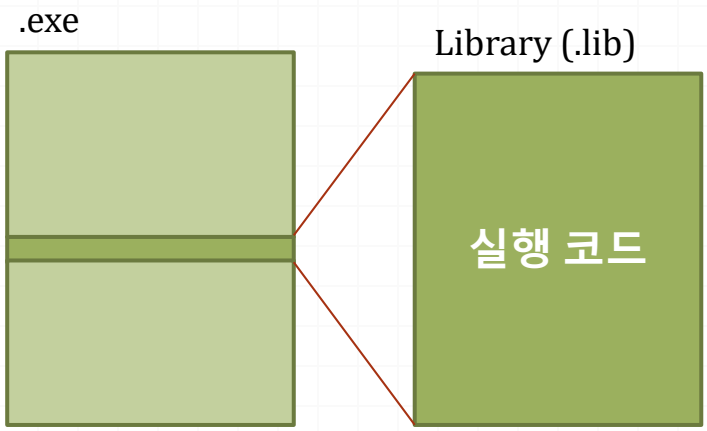
충남대학교 영상시스템 연구실

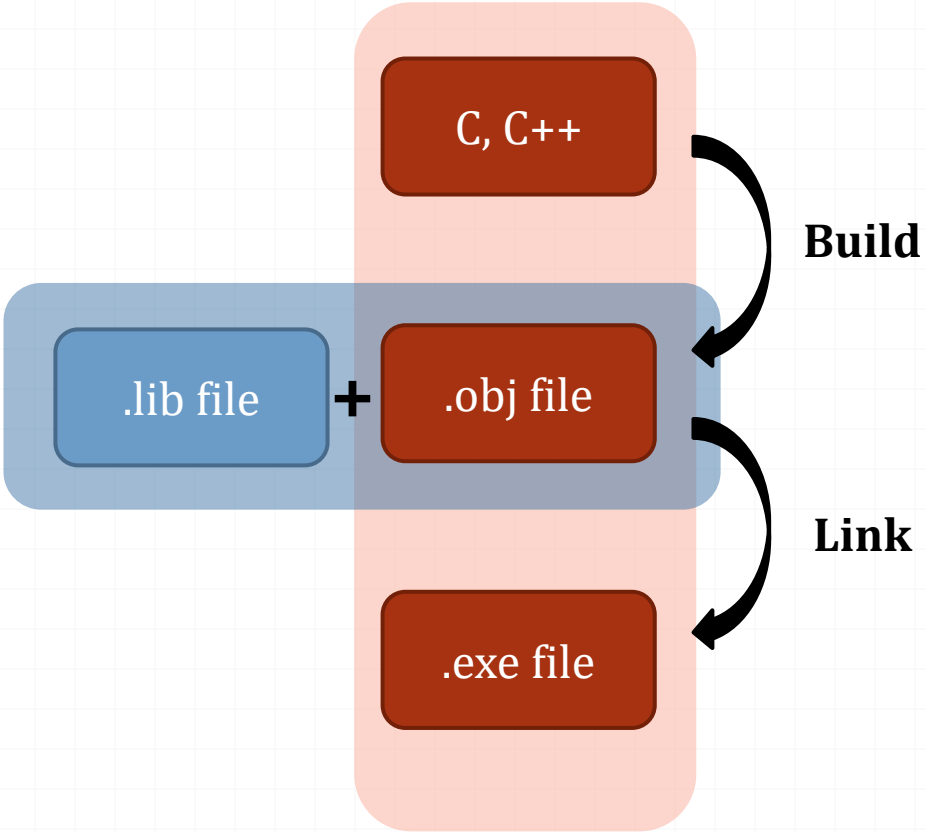
DLL 개요

- 라이브러리(Library)
  - 자주 사용되는 함수를 **미리 작성**하여 재사용할 수 있게 작성한 **모듈**
  - 실행 파일의 크기를 줄이거나 특정코드를 은닉하거나 객체화하는 장점이 있음
- 정적 링크 라이브러리 (Static linking library)
  - 컴파일된 목적 파일(.obj) 과 직접 링크하여 실행 이미지(.exe)를 만드는 방식
  - 링크 시 파일이 복사되므로 실행 프로그램 크기가 크지만 단일 개체로 사용
- 동적 링크 라이브러리 (Dynamic linking library)
  - 프로그램이 실행될 때 외부에 존재하는 라이브러리(DLL)의 함수를 사용하는 방식
  - 실행 파일 크기가 비교적 작지만 실행 시 DLL 파일이 필요
  - 유지보수가 쉬움

DLL 개요

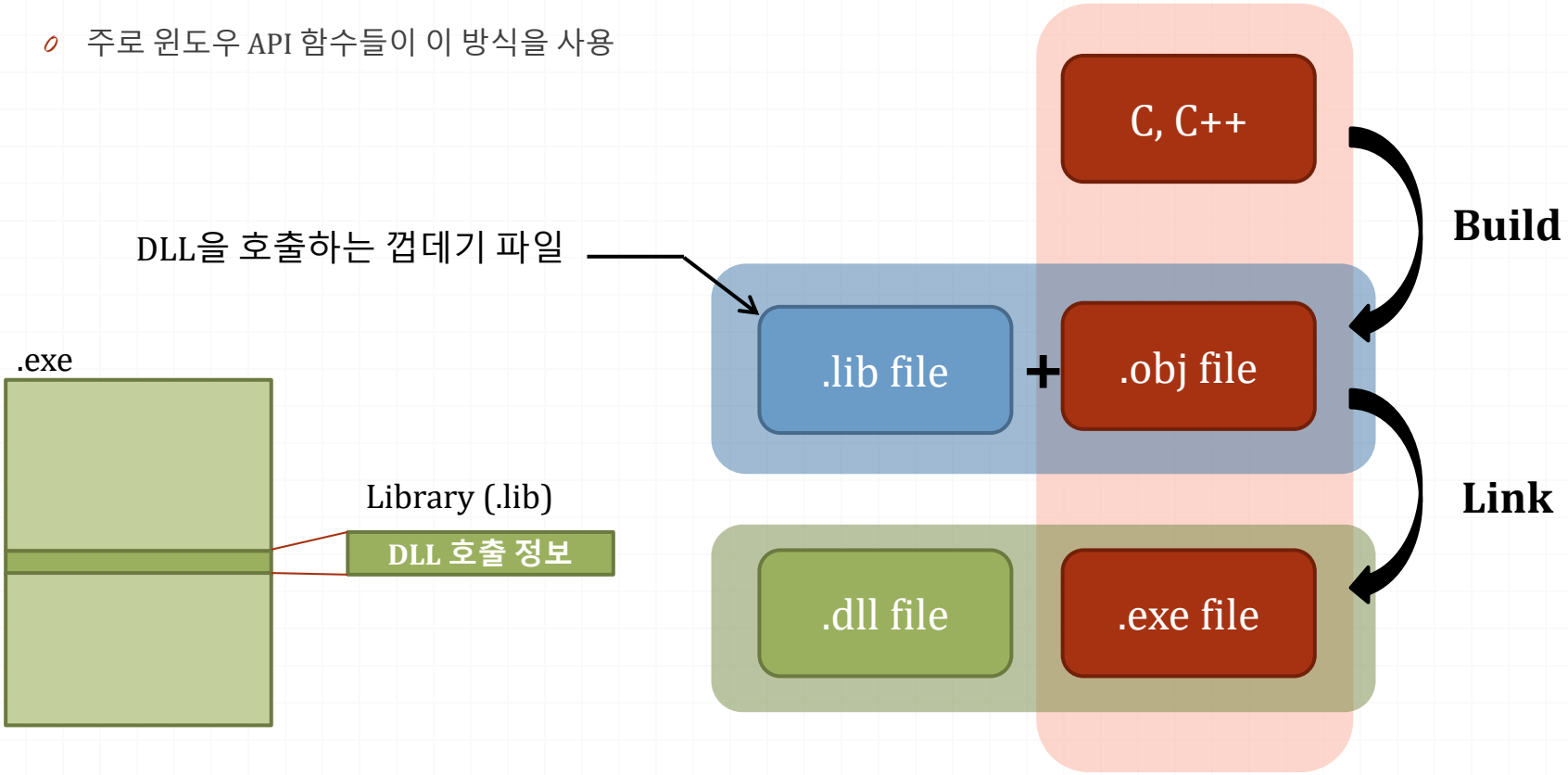
- 정적 링크 라이브러리 (Static linking library)
  - 라이브러리 파일에서 함수의 기능이 구현된 부분이 실행파일에 **추가**됨
  - 실행 파일 크기가 커짐





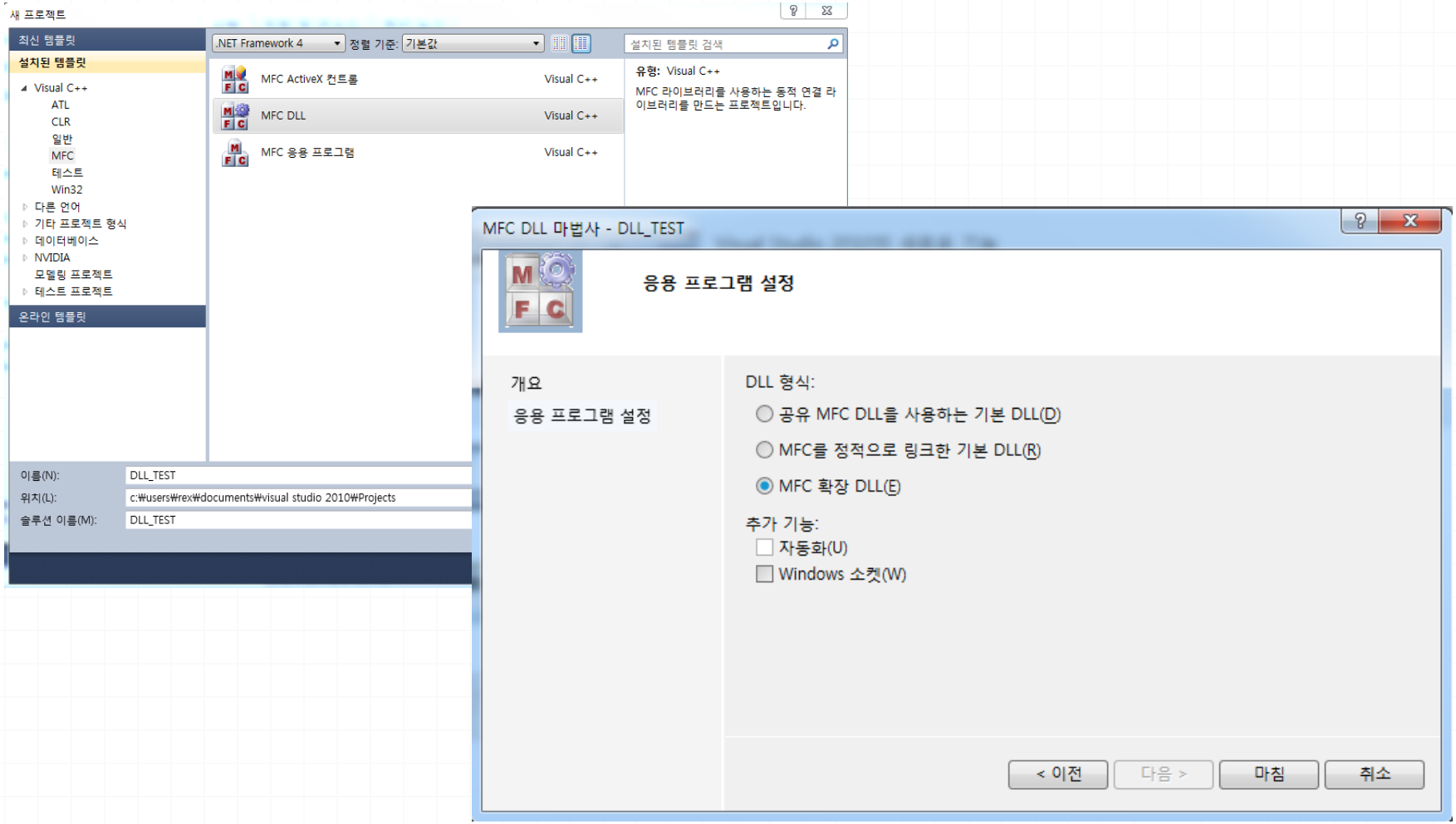
DLL 개요

- 동적 링크 라이브러리 (Dynamic linking library)
  - 컴파일 단계에서 라이브러리 정보만 포함
  - 실행될 때 lib파일 정보에 따라 외부 라이브러리(DLL)의 함수를 로드
  - 주로 윈도우 API 함수들이 이 방식을 사용



DLL 만들기

- DLL 프로젝트 생성



DLL 만들기

DLL 형식

- **공유 MFC DLL을 사용하는 기본 DLL (Using shared MFC DLL)**
  - 배포시 자신의 DLL과 MFC 공유 DLL을 같이 제공
  - Win32 및 MFC 프로그램 모두 DLL에서 함수를 호출 가능

- **MFC를 정적으로 링크한 기본 DLL (Statically Linked MFC DLL)**
  - 배포시 자신의 DLL만 제공 가능
  - Win32 및 MFC 프로그램 모두 DLL에서 함수를 호출 가능
  - DLL 크기가 커지지만 MFC DLL을 다시 배포할 필요가 없음

- **MFC 확장 DLL (Using Shared MFC DLL)**
  - MFC로 작성된 어플리케이션에서만 사용 가능
  - DLL 내부에서 MFC를 사용하는 경우 선택
  - **Class Export 가능**

DLL 만들기

- DllMain 함수

  DLL을 로드하거나 언로드할 때 호출

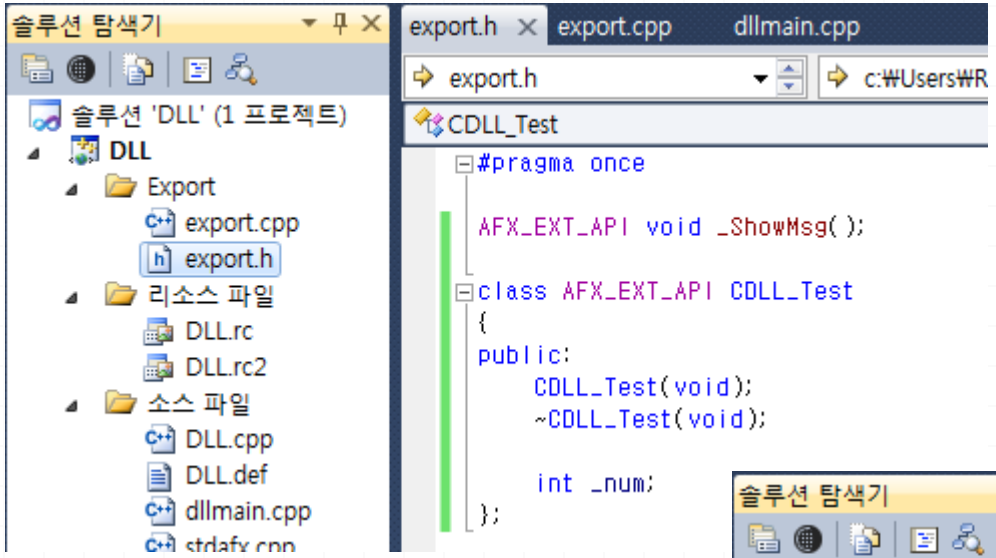
  호출 시점에 따라 dwReason 변수가 달라짐

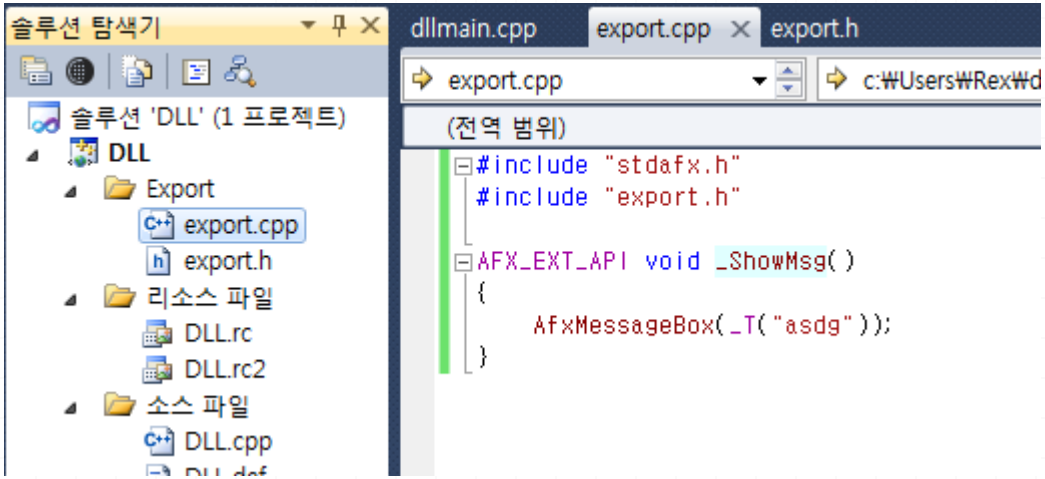
dllmain.cpp

```
#include "stdafx.h"
#include <afxwin.h>
#include <afxdllx.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

static AFX_EXTENSION_MODULE DLLDLL = { NULL, NULL };

extern "C" int APIENTRY
DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)
{
    // lpReserved를 사용하는 경우 다음을 제거하십시오.
    UNREFERENCED_PARAMETER(lpReserved);

    if (dwReason == DLL_PROCESS_ATTACH)          // 라이브러리 로드
    {
        TRACE0("DLL.DLL을 초기화하고 있습니다.\n");

        // 확장 DLL을 한 번만 초기화합니다.
        if (!AfxInitExtensionModule(DLLDLL, hInstance))
            return 0;

        new CDynLinkLibrary(DLLDLL);

    }
    else if (dwReason == DLL_PROCESS_DETACH)     // 라이브러리 언로드
    {
        TRACE0("DLL.DLL을 종료하고 있습니다.\n");

        // 소멸자가 호출되기 전에 라이브러리를 종료합니다.
        AfxTermExtensionModule(DLLDLL);
    }
    return 1;   // 확인
}
```

DLL 만들기

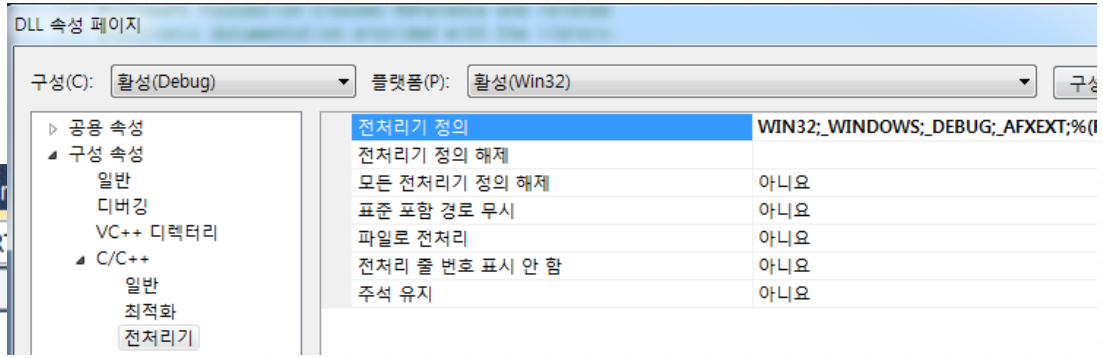
- Export 함수 작성





DLL 만들기

- AFX_EXE_API



afxver_h | afxv_dll.h | export.h | export.cpp | dllmain

AFX_API_EXPORT | #define AFX_API_EXPORT

(전역 범위)

```
// with decorated names.

#ifndef AFX_EXT_DATA
    #ifdef _AFXEXT
        #define AFX_EXT_CLASS       AFX_CLASS_EXPORT
        #define AFX_EXT_API         AFX_API_EXPORT
        #define AFX_EXT_DATA        AFX_DATA_EXPORT
        #define AFX_EXT_DATADEF
    #else
        #define AFX_EXT_CLASS       AFX_CLASS_IMPORT
        #define AFX_EXT_API         AFX_API_IMPORT
        #define AFX_EXT_DATA        AFX_DATA_IMPORT
        #define AFX_EXT_DATADEF
    #endif
#endif
```

afxver_h | afxv_dll.h | export.h | export.cpp | dllmain.cpp

afxver_h

(전역 범위)

```
// for global APIs
#ifndef AFX_API_EXPORT
    #define AFX_API_EXPORT __declspec(dllexport)
#endif
#ifndef AFX_API_IMPORT
    #define AFX_API_IMPORT __declspec(dllimport)
#endif
```

__declspec : 예약어, MS에서 만든 확장 문법

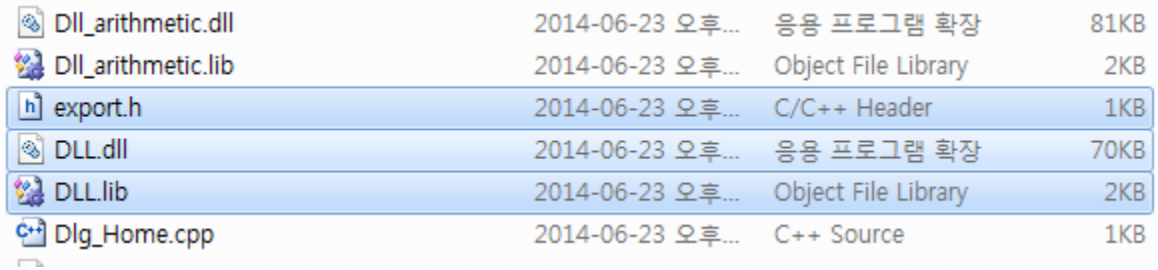
DLL 만들기

- 생성된 파일
  - .dll file : DLL 라이브러리
  - .lib file : 링크 시 사용할 라이브러리 (DLL 호출)
  - .pdb file : 디버깅에 사용하는 프로그램 데이터베이스

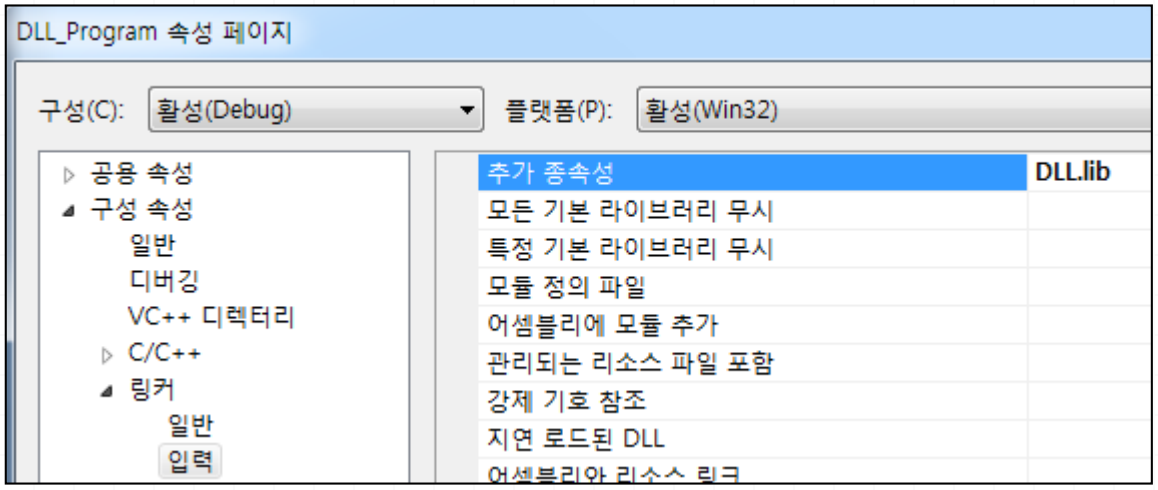
동영상 감상

DLL 사용하기

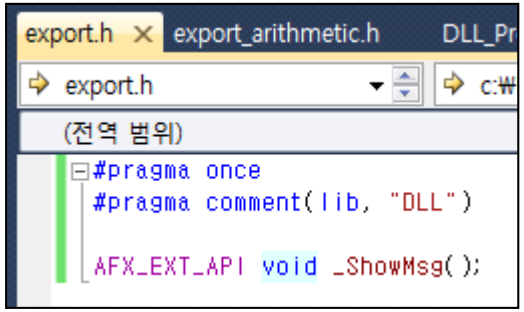
- 프로젝트 폴더에 **.dll, .lib 파일**과 **배포하는 함수 헤더** 복사



- 프로젝트 속성 – 링커 – 입력 –추가 종속성에 라이브러리 추가
  or 전처리기로 라이브러리 로드

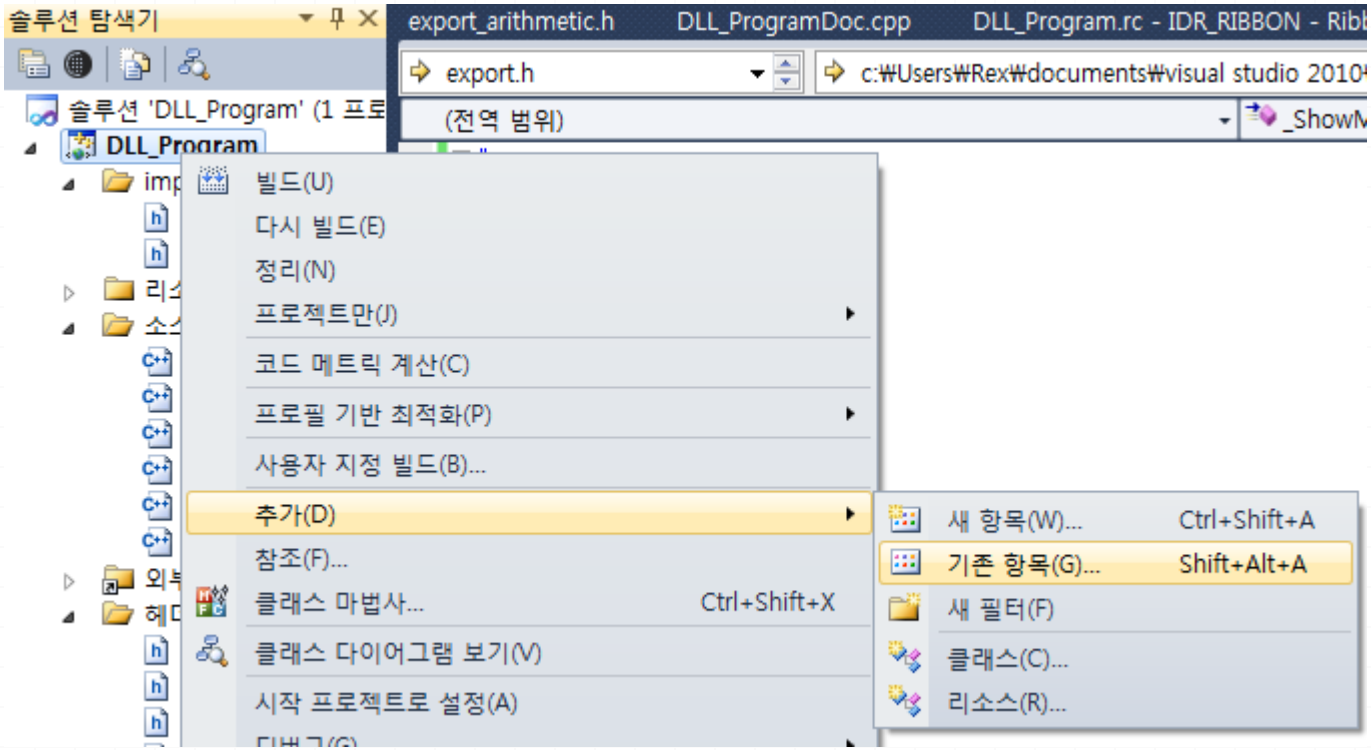


OR



DLL 사용하기

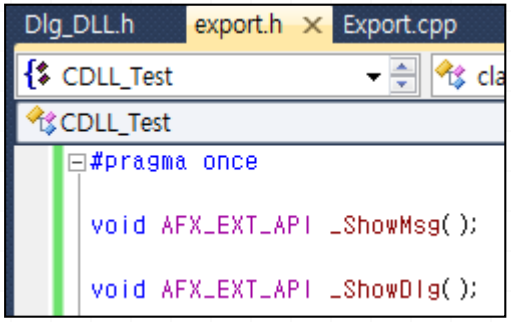
- 배포하는 함수 헤더를 프로젝트에 추가

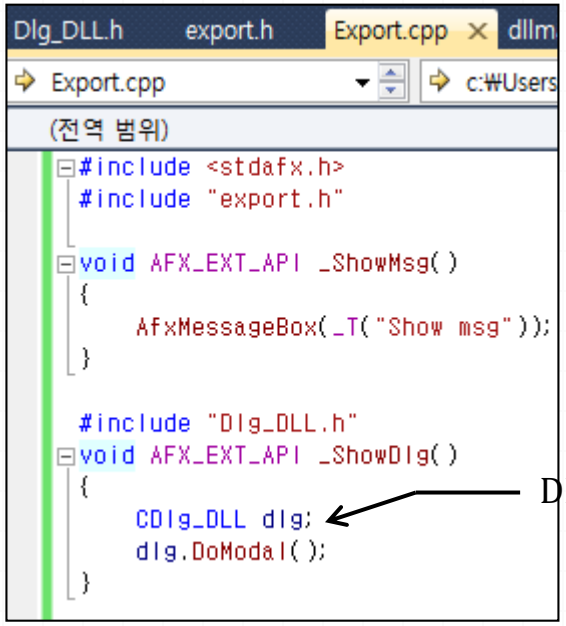


동영상 감상

MFC DLL의 resource 찾는 순서

- MFC extension DLL에서 resource를 가져올 때, EXE file과 DLL file의 중복된 ID가 있는 경우에는 EXE file의 resource를 가져오게 된다. 이것은 MFC에서 resource를 찾는 순서가 Extension DLL 인 경우 EXE file의 resource 를 가장 먼저 살펴 보고 여기에서 발견이 안된 경우에 Extension DLL resource를 그리고 마지막 으로 MFC DLL resource를 찾게 되기 때문이다.





MFC DLL의 resource 찾는 순서

- Solution

AfxSetResourceHandle()를 이용하여 DLL의 instance를 application의 default resource 위치로 잠시 설정

Dlg_DLL.h | export.h | Export.cpp | dllmain.cpp

_ShowDlg | void AFX_EXT_API _ShowDlg()

(전역 범위)

```
void AFX_EXT_API _ShowMsg()
{
    AfxMessageBox(_T("Show msg"));
}

#include "Dlg_DLL.h"
void AFX_EXT_API _ShowDlg()
{
    HINSTANCE hInstResourceClient = AfxGetResourceHandle();
    AfxSetResourceHandle(g_hInstance);

    CDlg_DLL dlg;
    dlg.DoModal();

    AfxSetResourceHandle(hInstResourceClient);
}
```

Dlg_DLL.h | export.h | Export.cpp | dllmain.cpp

dllmain.cpp | c:₩Users₩Rex₩documents₩visual studio 2010

(전역 범위)

```
HINSTANCE g_hInstance;

extern "C" int APIENTRY
DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID lpReserved)
{
    g_hInstance = hInstance;

    // lpReserved를 사용하는 경우 다음을 제거하십시오.
```

stdafx.h | Dlg_DLL.h | export.h | Expor

stdafx.h | c:₩Use

(전역 범위)

```
#endif // _AFX_NO_AFXCMN_SUPPORT


extern HINSTANCE g_hInstance;
```

extern “C”

C 형식으로 네임 맹글링(Name mangling)

export.h | export.cpp | dllmain.cpp

export.cpp — c:₩Users₩Rex₩d

(전역 범위)

```
#include "stdafx.h"
#include "export.h"

extern "C" AFX_EXT_API void _ShowMsg()
{
    AfxMessageBox(_T("asdg"));
}
```

Dependency Walker - [DLL.dll]

File Edit View Options Profile Window Help

DLL.DLL
- MFC100UD.DLL
- MSVCR100D.DLL
- KERNEL32.DLL
- OLEAUT32.DLL
- ADVAPI32.DLL

PI	Ordinal ^	Hint	Function	Entry Point
E	Ordinal ^	Hint	Function	Entry Point
C++	1 (0x0001)	0 (0x0000)	?_ShowMsg@@YAXXZ	0x000112FD

Dependency Walker - [DLL.dll]

File Edit View Options Profile Window Help

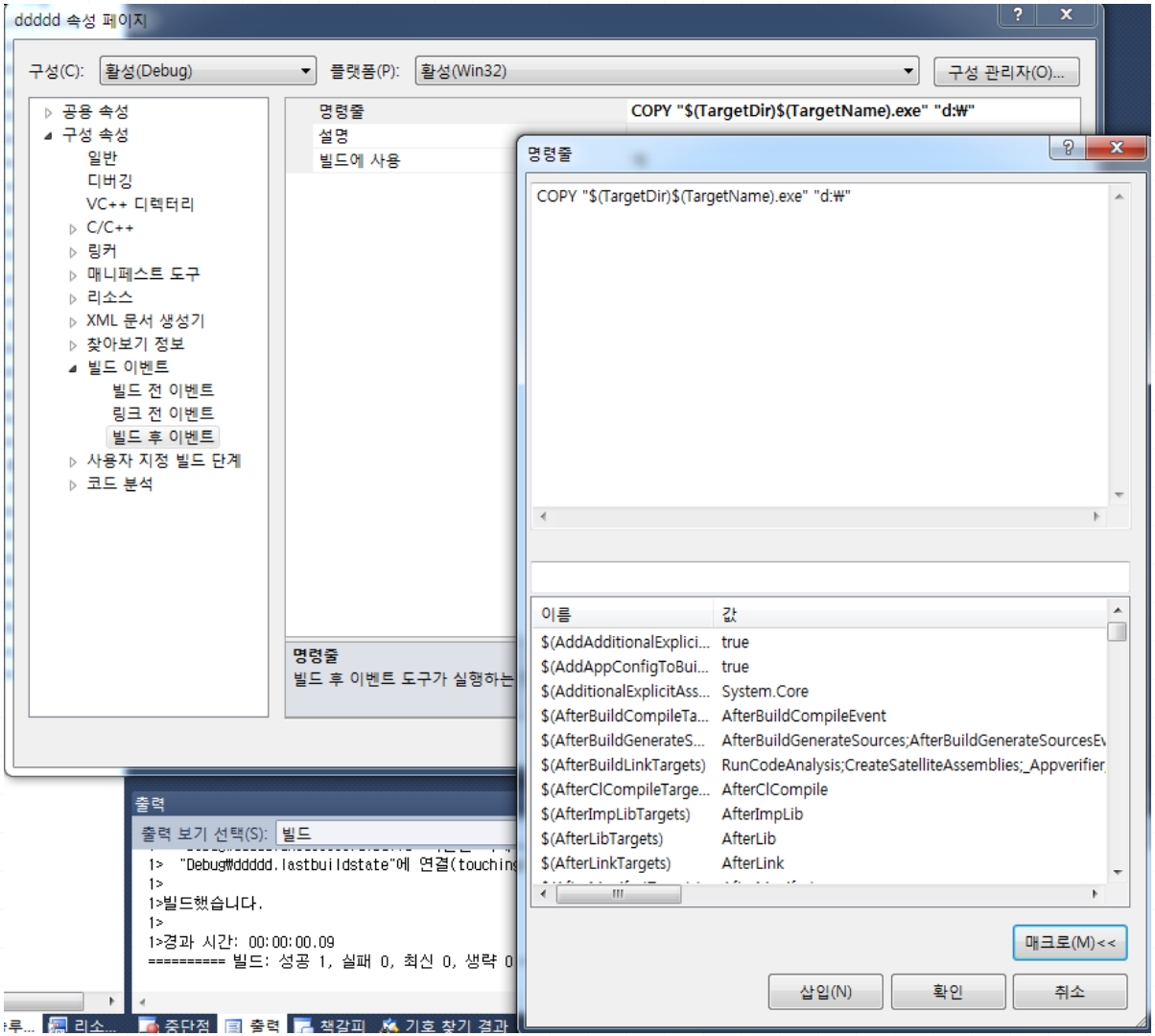
DLL.DLL
- MFC100UD.DLL
- MSVCR100D.DLL
- KERNEL32.DLL
- OLEAUT32.DLL
- ADVAPI32.DLL

PI	Ordinal ^	Hint	Function	Entry Point
E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	0 (0x0000)	_ShowMsg	0x00011398

Module	File Time Stamp ^	Link Time Stamp	File Size	Attr.	Link Check
API-MS-WIN-APPMODEL-RUNTIME-L1-1-0.DLL	Error opening file. 지정된 파일을 찾을 수 없습니다 (2).				
API-MS-WIN-CORE-WINRT-ERROR-L1-1-0.DLL	Error opening file. 지정된 파일을 찾을 수 없습니다 (2).				
API-MS-WIN-CORE-WINRT-L1-1-0.DLL	Error opening file. 지정된 파일을 찾을 수 없습니다 (2).				
API-MS-WIN-CORE-WINRT-ROBUFFER-L1-1-0.DLL	Error opening file. 지정된 파일을 찾을 수 없습니다 (2).				

빌드 후 이벤트

- 프로젝트 속성 – 빌드 이벤트 – 빌드 후 이벤트
- 빌드 된 파일 복사 ex) COPY "$(TargetDir)$(TargetName).dll" " 대상 경로 "



reference

http://support.microsoft.com/kb/600771/ko : MFC DLL의 resource 찾는 순서

http://debugjung.tistory.com/entry/MFC-DLL%EC%97%90%EB%8C%80%ED%95%9C-%EC%A0%95%EB%A6%AC