Visual Studio 2010에서 OpenCV MFC 프로젝트 만들기

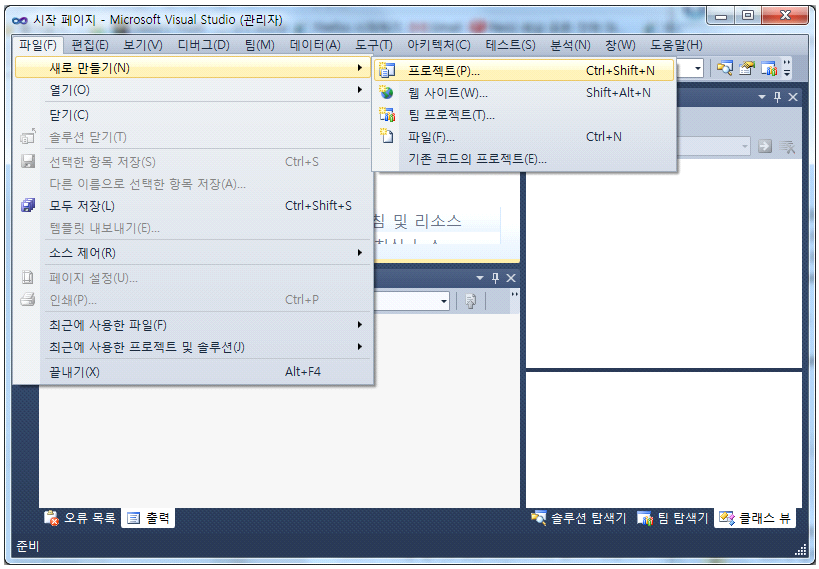
http://deios.kr

이번에는 Visual Studio 2010에서 OpenCV 2.1 라이브러리를 활용하여 MFC프로젝트를 만들고, 이미지를 불러와서 화면에 출력해보겠습니다.

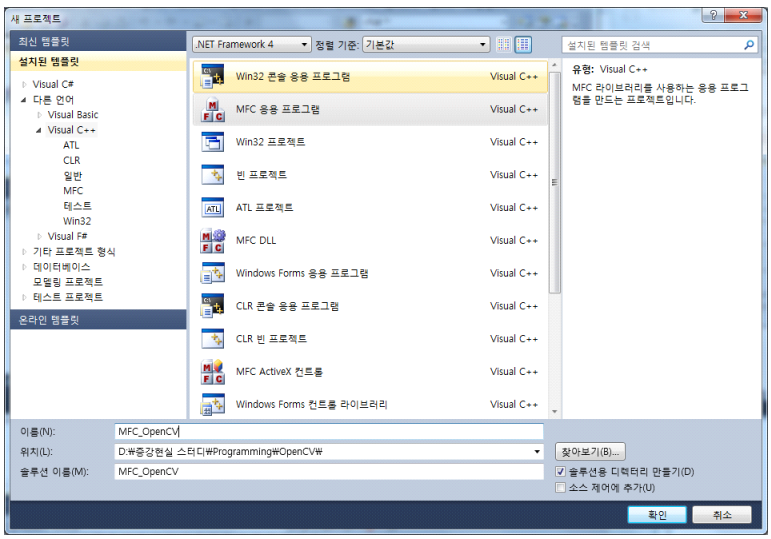
먼저 제 환경은 다음과 같고, DLL파일은 path환경변수에 설정되어 있습니다.

DLL : "D:₩lib₩bin"
Header : "D:₩lib₩include"
Library : "D:₩lib₩lib"
Source : "D:₩lib₩src"

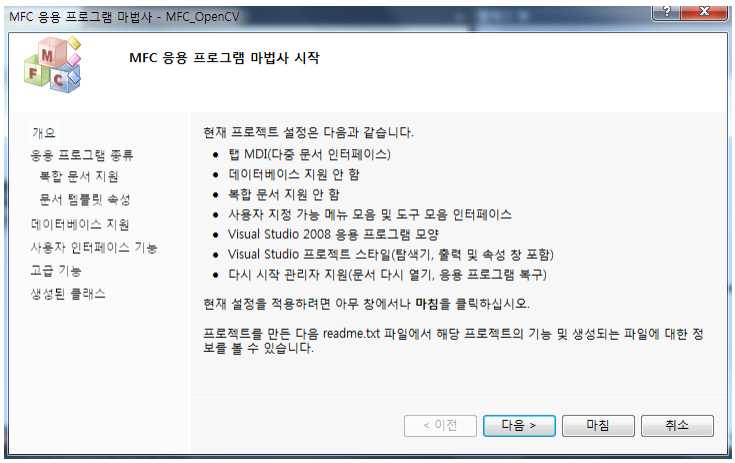
VS2010에서 [파일] → [새로만들기] → [프로젝트]를 선택합니다.



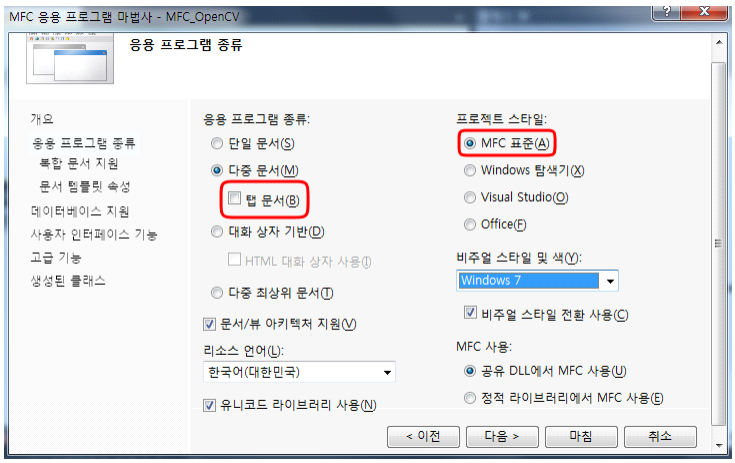
너무나도 당연하게 'Visual C++ MFC 응용 프로그램'을 선택해 주어야겠죠?



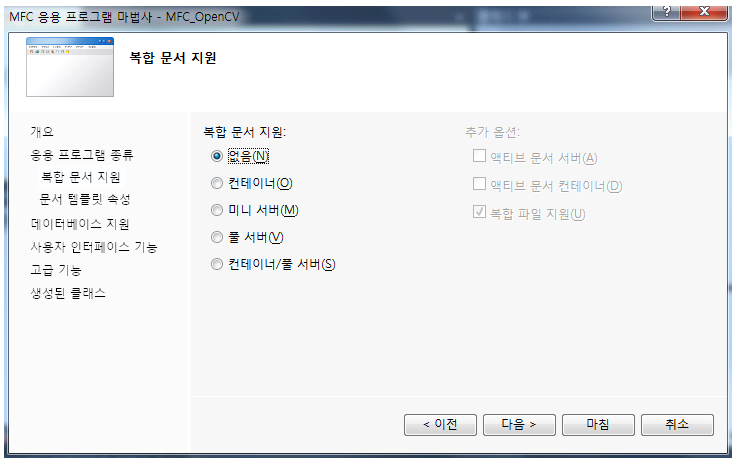
MFC 응용 프로그램 마법사입니다. “다음”을 가뿐하게 눌러 줍니다.



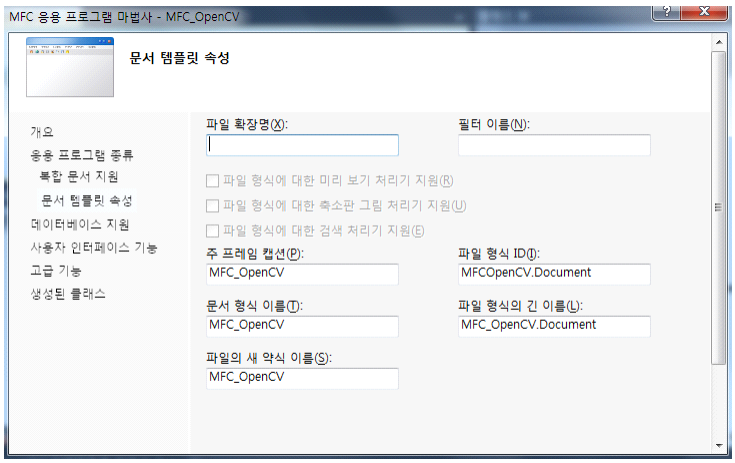
“탭 문서”의 체크를 해제해 주고, “프로젝트 스타일”을 “MFC 표준”으로 지정해 줍니다.



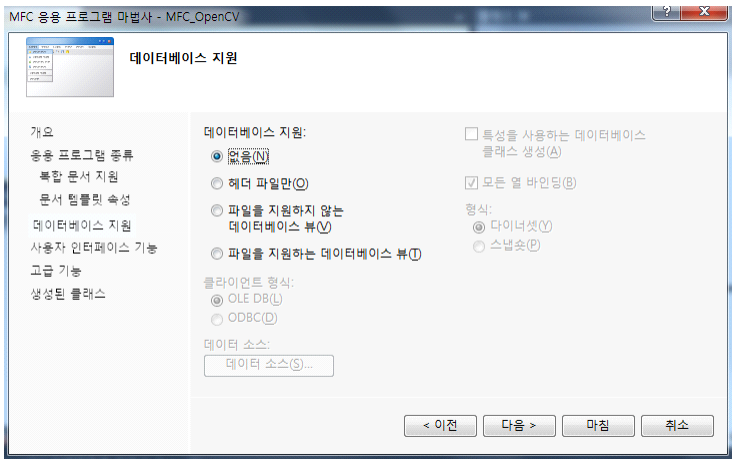
역시 다음을 사정없이 클릭해 줍니다.



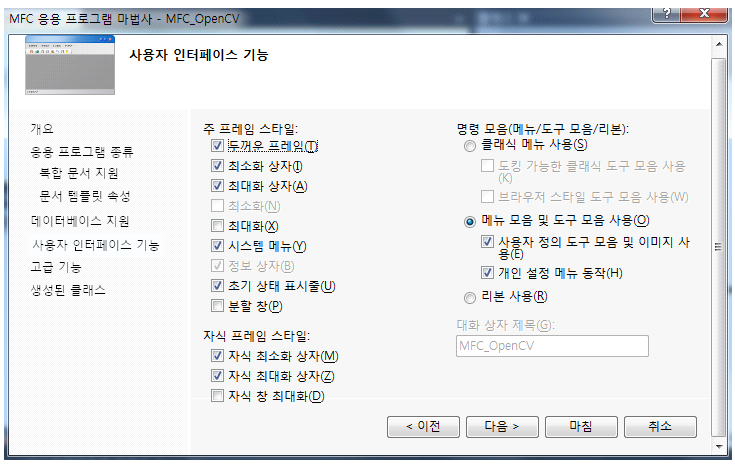
이번에도 그냥 다음



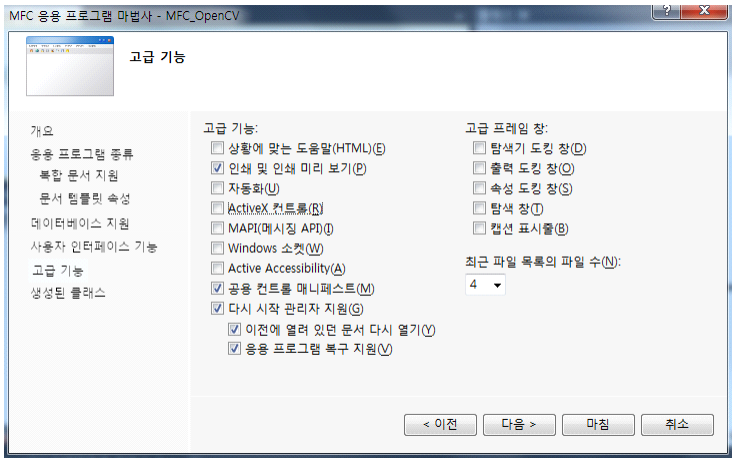
에... 또 다음이네요...



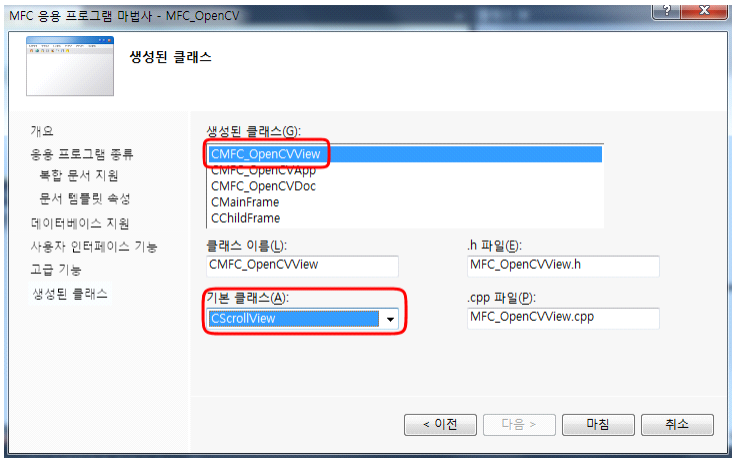
이번에도 별거 없습니다. 이것저것 바꾸셔도 되고요...



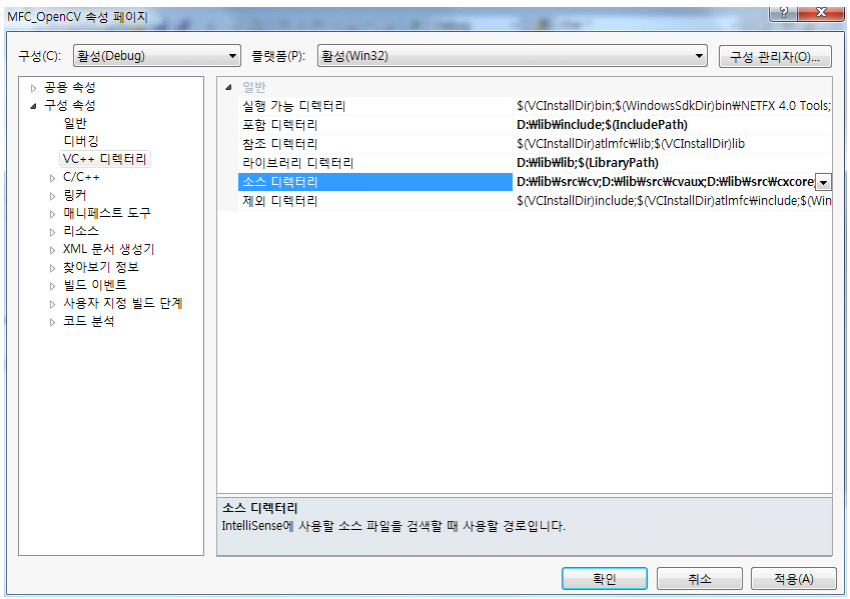
이번에도 그냥 지나가시면 됩니다. 저는 기분나쁜 ActiveX 컨트롤러를 체크 해제 합니다. 사실 별 상관 없습니다.



스크롤을 써먹기 위해서는 'View 클래스'의 기본 클래스를 "CScrollView"로 지정합니다.

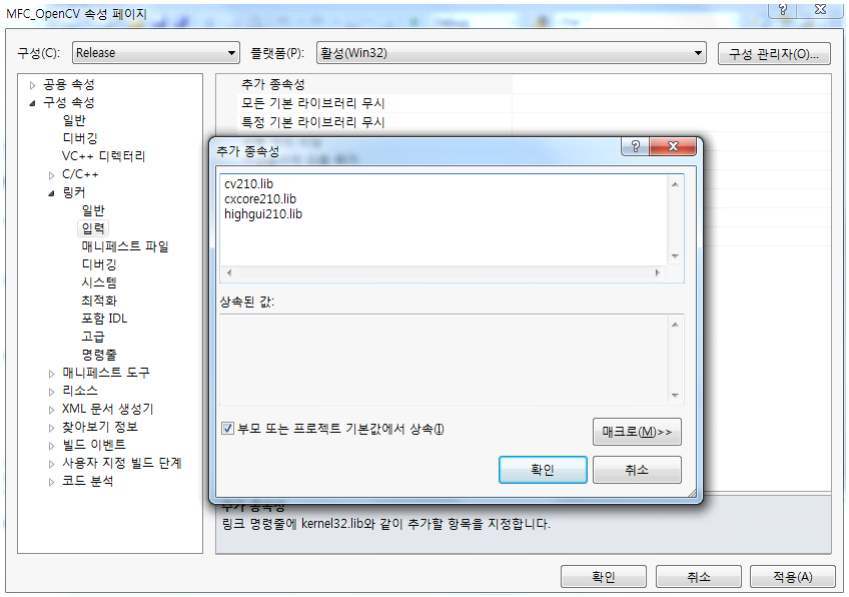


이제 OpenCV를 사용하기 위한 설정을 해줘야 겠죠? 프로젝트 속성을 엽니다.



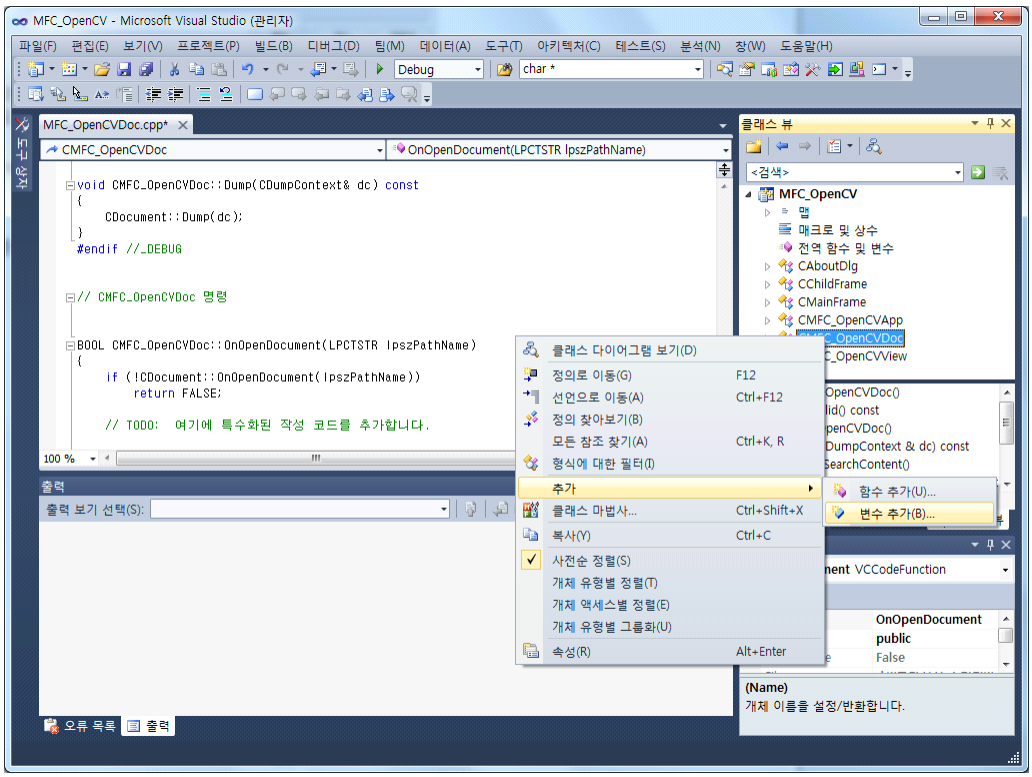
Debug와 Release모두 VC++ 디렉터리를 지정해 줍니다.
자세한 사항은 “Visual Studio 2010에서 OpenCV 이용하기(http://deios.kr/395)”를 참고하세요~

[링커] → [입력]의 “추가 종속성”부분에 lib 파일을 기술합니다.

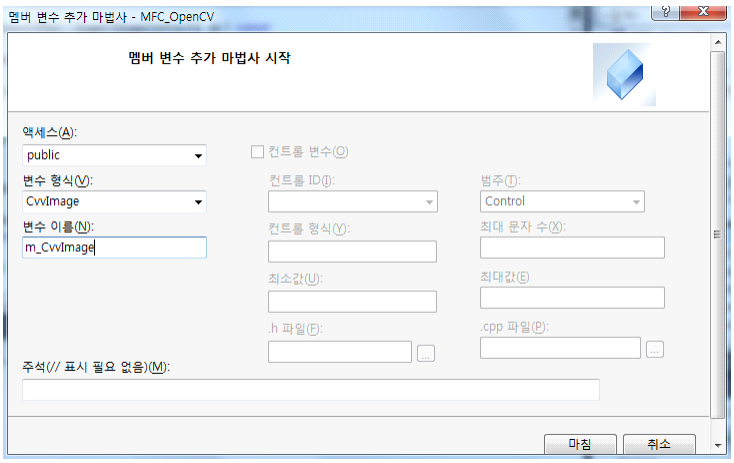


이렇게 하면 OpenCV라이브러리를 사용하기 위한 준비가 끝납니다.
이제 본격적으로 MFC프로젝트에서 OpenCV라이브러리를 사용해 볼까요?

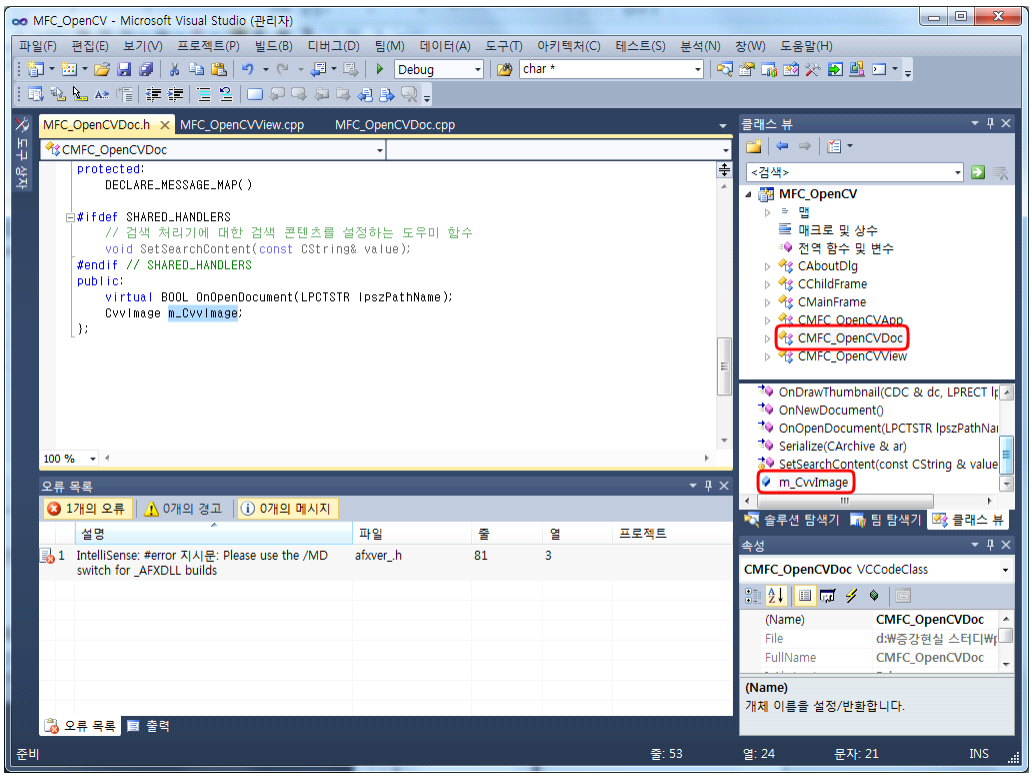
먼저 영상 데이터를 담을 CvvImage클래스의 맴버 변수를 등록해야 합니다.
Doc클래스 위에서 [추가] → [변수 추가]를 눌러 줍니다.



변수 형식은 “CvvImage”로 변수 이름은 “m_CvvImage”로 지정합니다.



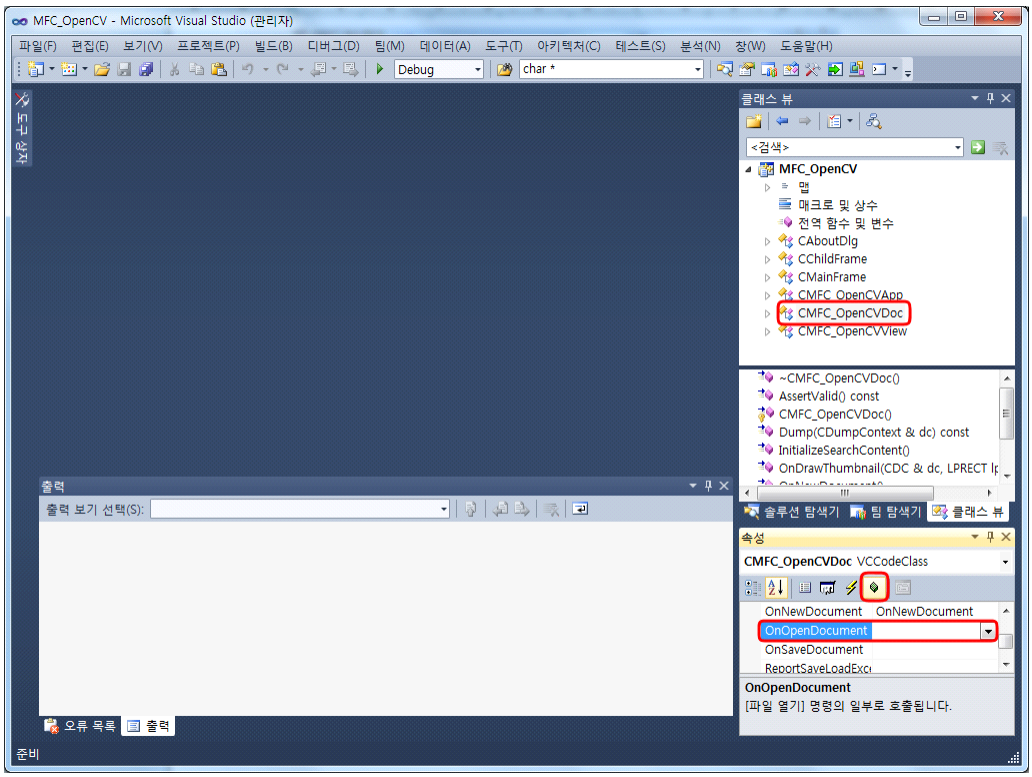
헤더 파일도 추가할겸 해당 맴버 변수가 등록되는 헤더 파일을 열어봅시다.



저 부분에 직접 맴버 변수를 등록해도 됩니다.
상단으로 이동하여 다음과 같은 헤더 파일을 include 합니다.

```
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>
```

다음으로 Doc클래스의 OnOpenDocument를 재정의 합니다.



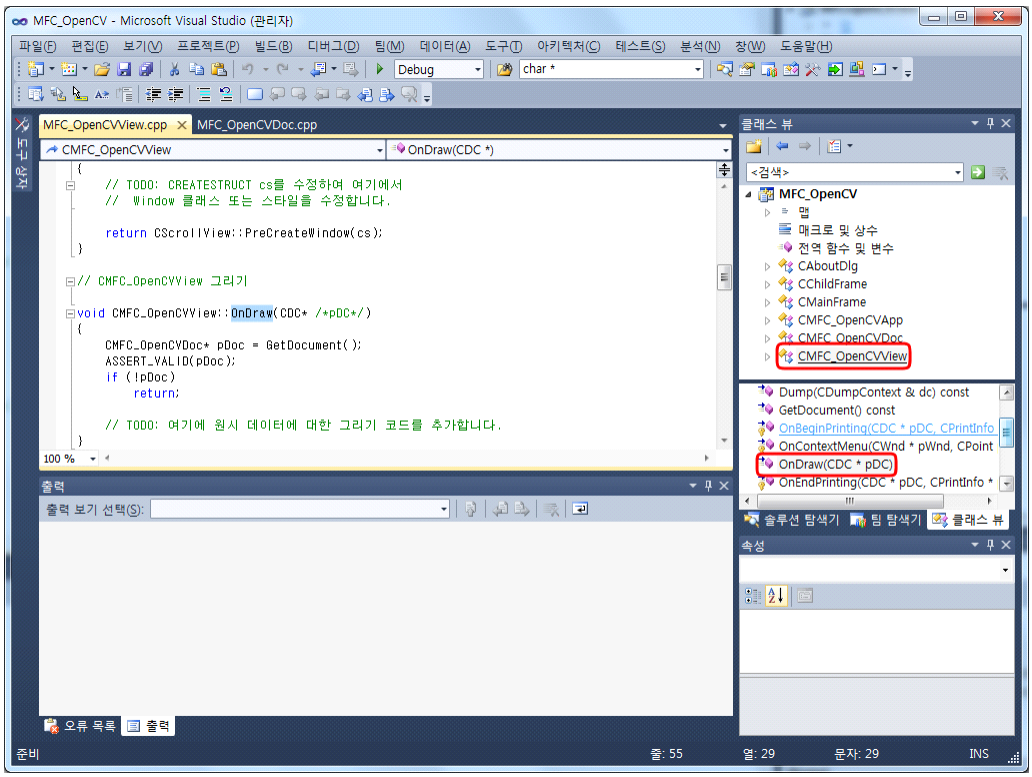
사정없이 클릭해줘야겠죠?

```
BOOL CMFC_OpenCVDoc::OnOpenDocument(LPCTSTR lpszPathName)
{       if (!CDocument::OnOpenDocument(lpszPathName)) return FALSE;

        USES_CONVERSION;
        m_CvvImage.Load(W2A(lpszPathName));

        return TRUE;
}
```

위와 같이 코딩합니다.
이때, OpenCV는 char*형을 사용하고, VS2010의 Dialog는 LPCTSTR형을 사용하기 때문에 FullFilePathName을 받아오기 위해서 W2A라는 매크로를 사용합니다.

이제 화면에 보여주는 일만 남은 것 같네요. 화면에 보여주는 작업은 3단계를 거칠 예정입니다.
첫 번째로 OnDraw를 재정의하여 화면에 출력해 주는 부분을 추가할 것이고,
두 번째로 OnInitialUpdate를 재정의하여 이미지 사이즈에 View크기를 맞출 것입니다.
마지막으로 InitInstance를 재정의하여 프로그램을 처음 시작하면 나오는 빈 화면을 제거할 것입니다.

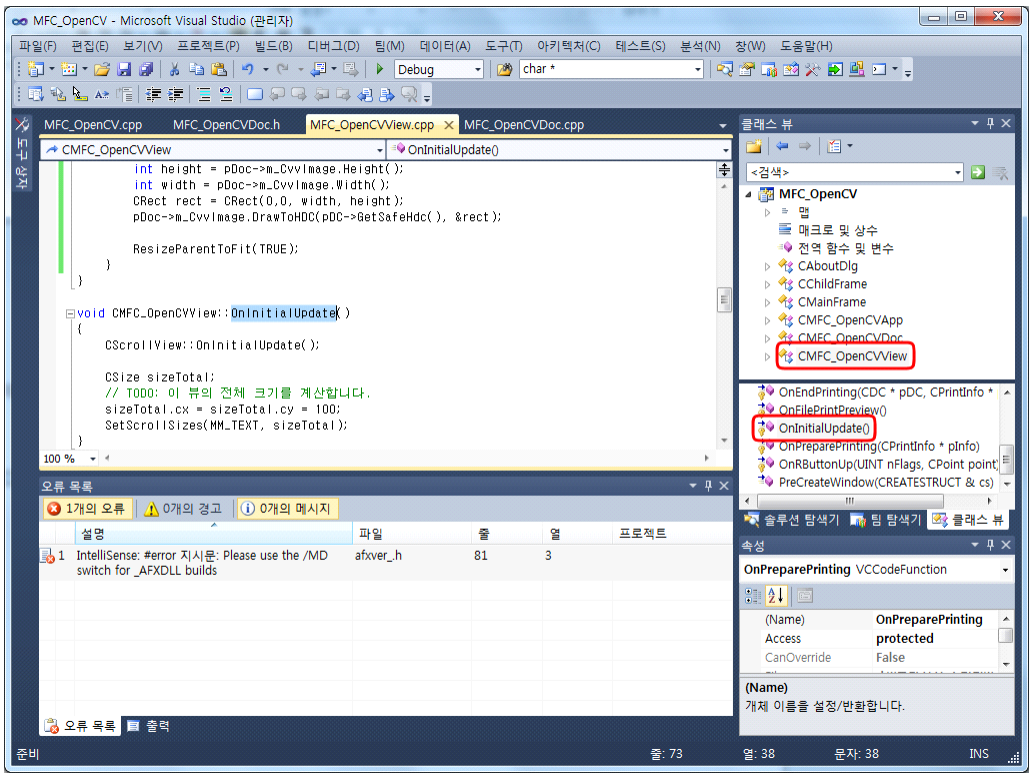
먼저 View클래스의 OnDraw를 사정없이 클릭해 줍니다.



다음과 같이 코딩합니다.

```
void CMFC_OpenCVView::OnDraw(CDC* pDC)
{
	CMFCOpenCVTestDoc* pDoc = GetDocument();
	ASSERT_VALID(pDoc);
	if (!pDoc)
		return;

	// TODO: 여기에 원시 데이터에 대한 그리기 코드를 추가합니다.
	if(pDoc->m_CvvImage.GetImage()){
		int height = pDoc->m_CvvImage.Height();
		int width = pDoc->m_CvvImage.Width();
		CRect rect = CRect(0,0, width, height);
		pDoc->m_CvvImage.DrawToHDC(pDC->GetSafeHdc(), &rect);

		ResizeParentToFit(TRUE);
	}
}
```

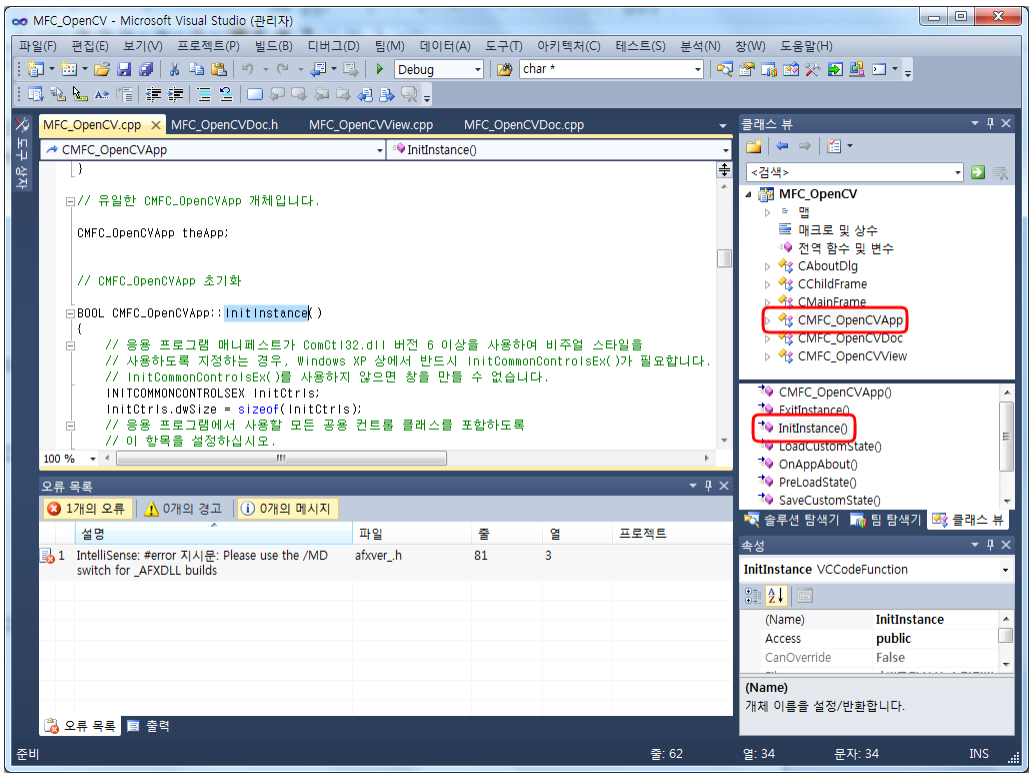
이번에는 OnInitialUpdate를 수정해 보겠습니다.



역시 다음과 같이 코딩합니다.

```
void CMFC_OpenCVView::OnInitialUpdate()
{
	CMFC_OpenCVDoc *pDoc = GetDocument();
	ASSERT_VALID(pDoc);
	CScrollView::OnInitialUpdate();

	CSize sizeTotal;
	// TODO: 이 뷰의 전체 크기를 계산합니다.
	if(pDoc->m_CvvImage.GetImage()){
		int height = pDoc->m_CvvImage.Height();
		int width = pDoc->m_CvvImage.Width();
		sizeTotal = CSize(width, height);
	}else{
		sizeTotal.cx = sizeTotal.cy = 100;
	}
	SetScrollSizes(MM_TEXT, sizeTotal);
	ResizeParentToFit(TRUE);
}
```

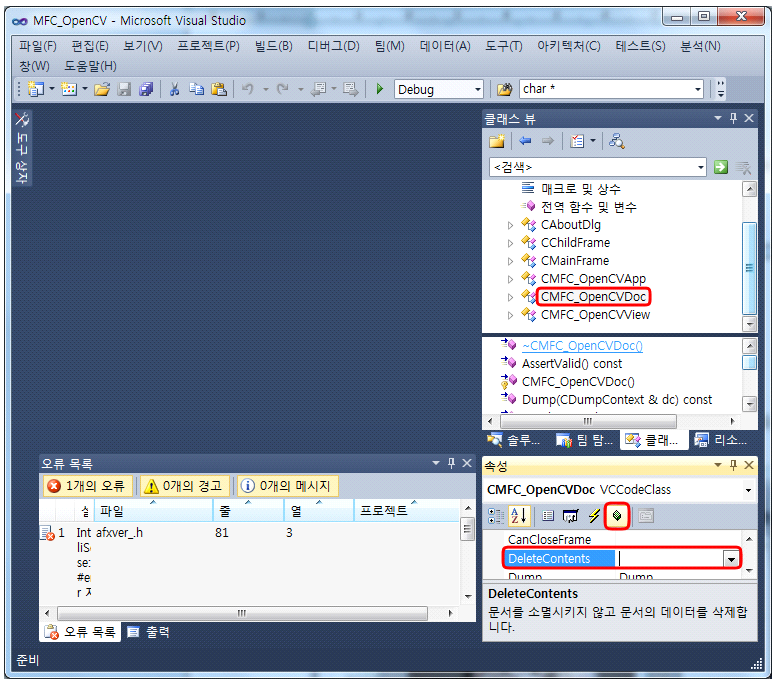
마지막으로 프로그램 처음 실행시 나오는 왠지 기분 나쁜 빈 화면을 제거해 보겠습니다.



다음과 같이 코딩합니다.

```
// 표준 셸 명령, DDE, 파일 열기에 대한 명령줄을 구문 분석합니다.
CCommandLineInfo cmdInfo;
ParseCommandLine(cmdInfo);

if(cmdInfo.m_nShellCommand == CCommandLineInfo::FileNew){
        cmdInfo.m_nShellCommand = CCommandLineInfo::FileNothing;
}
```

마지막으로 m_CvvImage에 할당된 메모리를 해제하겠습니다.



다음과 같이 코딩합니다.

```
void CMFC_OpenCVDoc::DeleteContents()
{
	// TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.
	if(NULL != m_CvvImage.GetImage()) m_CvvImage.~CvvImage();
	CDocument::DeleteContents();
}
```