

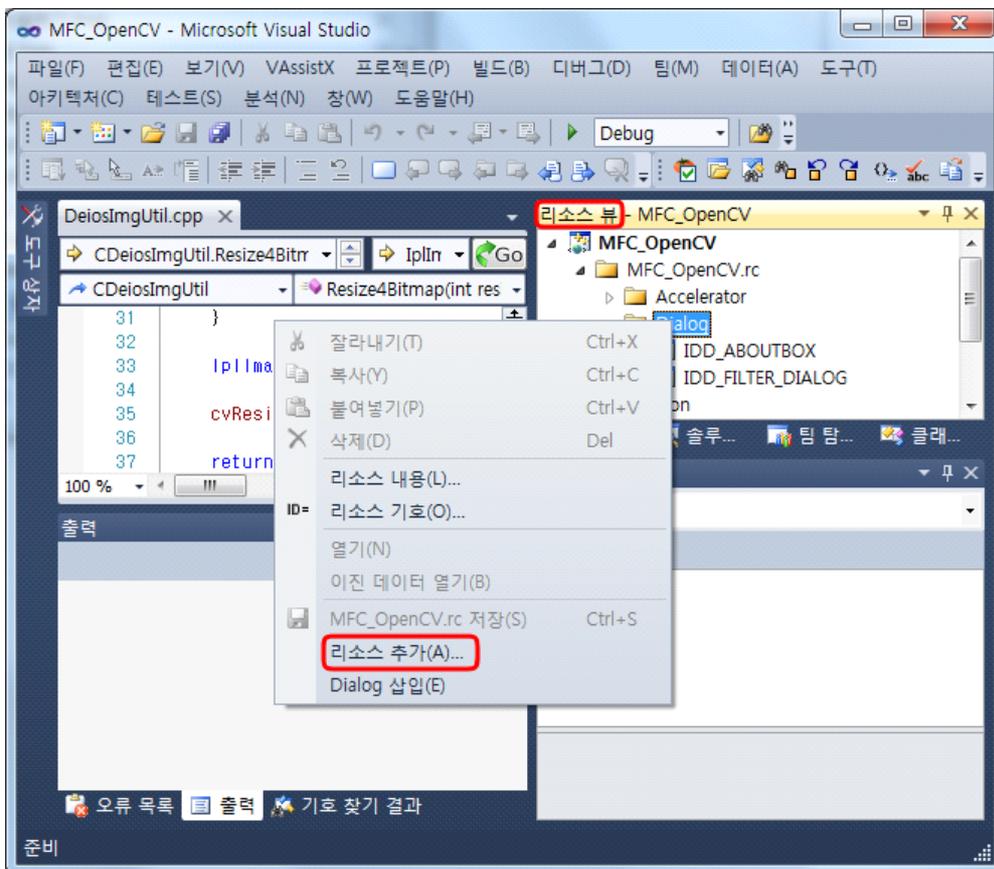
# MFC - Cam 영상 출력해보기

<http://deios.kr>

이번 시간에는 MFC프로젝트에서 Cam 영상을 받아오는 부분을 구현해 보겠습니다.

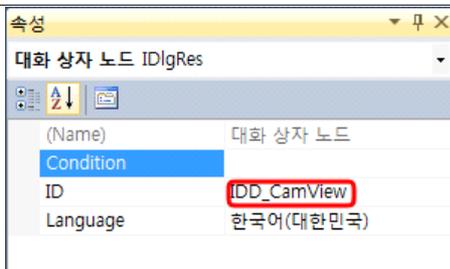
"동영상 처리" → "영상 보기"메뉴를 추가하고, 해당 메뉴를 클릭했을 때, 영상을 뿌려주는 Dialog를 띄웁니다. Dialog에서 "시작" 버튼을 클릭했을 경우 Caption이 "중지"로 바뀌고, Static 영역에 영상을 뿌려주게 만들어 보겠습니다.

먼저 사용할 다이얼로그부터 만들어야겠죠?  
[리소스 뷰]에서 [리소스 추가]를 선택합니다.



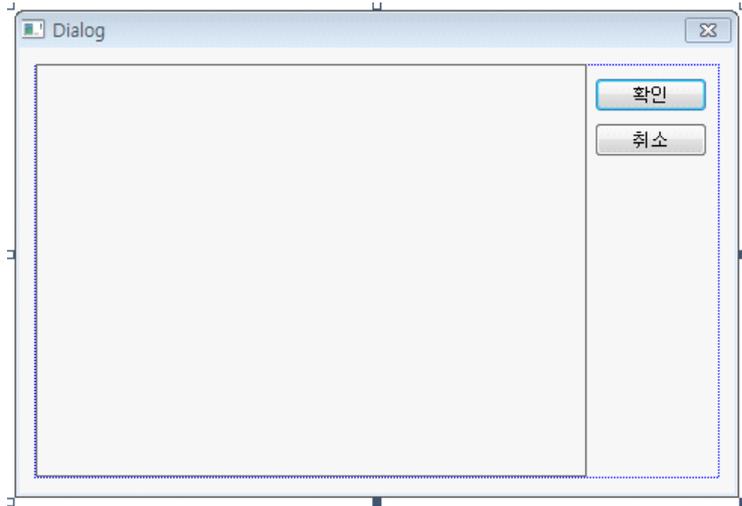
[Dialog]가 선택된 것을 확인하고 [새로 만들기]를 누릅니다.

ID : IDD\_CamView



여기에 [Picture Control]을 추가하겠습니다. 방법은 “[공간 영역 필터링 - Dialog Box 기반 구현](http://deios.kr/409#AddControl)”을 참고하세요.

ID는 “IDC\_STATIC\_VIEW”로 주고 다음과 같이 컨트롤을 적당하게 배열합니다.



이제 “확인”버튼의 Caption을 “시작”으로, “취소”버튼의 Caption을 “닫기”로 변경합니다. Dialog의 Caption도 “Cam View”정도로 바꿔주는 것이 좋겠네요. 이 정도는 눈 감고도 하실 수 있겠죠?

UI의 구성은 대충 끝났고, 이제 이 Dialog에서 무슨 짓을 할지 생각해 봅시다~

먼저, [클래스 추가]를 이용해 이 Dialog에 대응하는 Class를 만들어 줘야겠죠? 그 후 초기화 부분인 “OnInitDialog()”에서 각종 초기화와 “cvCaptureFromCAM”함수를 호출해 줘야겠고, “시작”버튼에 대한 이벤트 처리기에서 화면에 Cam영상을 뿌려주는 코드를 작성해야겠네요... 그리고 “시작”버튼을 “중지”로 만들어 줘야겠죠? 이 작업을 처리하려면 “IDC\_STATIC\_VIEW”와 “IDOK”는 대응하는 멤버 변수를 만들어 줘야겠네요. “OnDestroy()”에서는 “cvReleaseCapture”함수를 호출해 줘야 깔끔하겠죠?

Cam영상을 뿌려주는 부분은 이미지를 연속적으로 받아서 출력해줘야 하니, Timer를 하나 등록해서 “OnTimer”이벤트 처리기에서 작업을 처리해 주는게 좋겠네요. 자... 그럼 작업은 다음과 같이 진행되었습니다.

- [1. Dialog에 대응하는 Class 만들기](#)
- [2. OnInitDialog\(\)에서 초기화 및 cvCaptureFromCAM함수 호출](#)
- [3. OnTimer\(\)구현](#)
- [4. OnOK\(\)구현](#)
- [5. 메뉴에 추가](#)

자~ 그럼 시작해 볼까요?

### 1. Dialog에 대응하는 Class만들기

Class추가 방법이 기억나지 않으시는 분은 "[공간 영역 필터링 - Dialog Box 기반 구현](http://deios.kr/409#DialogClass)(<http://deios.kr/409#DialogClass>)"를 참고하세요~ [클래스 이름]은 "CViewDlg"로 할께요~

이제 다음과 같은 멤버 변수를 추가합니다.

```
CvImage m_CvImage;
CvCapture *capture;
IplImage *image;
bool viewState;
```

그리고 Control에 해당하는 멤버 변수도 만들어 줘야겠죠? 앞으로 Control에 해당하는 멤버 변수는 "Control 객체"라고 부르겠습니다. "IDC\_STATIC\_VIEW"와 "IDOK"에 해당하는 Control 객체를 만들어 줍니다. 역시 기억나지 않으시는 분은 "[공간 영역 필터링 - Dialog Box 기반 구현](http://deios.kr/409#ControlObj)(<http://deios.kr/409#ControlObj>)"를 참고해 주세요~

```
IDC_STATIC_VIEW : m_sView
IDOK : m_btnOk
```

### 2. OnInitDialog()에서 초기화 및 cvCaptureFromCAM함수 호출

OnInitDialog()를 재정의 합니다. 그 후 다음과 같은 코드를 작성합니다.

```
BOOL CViewDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();
    // TODO: 여기에 추가 초기화 작업을 추가합니다.
    capture = cvCaptureFromCAM(0);
    image = 0;
    viewState = FALSE;

    return TRUE; // return TRUE unless you set the focus to a control
    // 예외: OCX 속성 페이지는 FALSE를 반환해야 합니다.
}

```

### 3. OnTimer()구현

WM\_Timer에 대한 이벤트 처리기를 추가하고 구현합니다. 코드는 다음과 같습니다.

```
void CViewDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.
    if(1 == nIDEvent){
        cvGrabFrame( capture );
        image = cvQueryFrame( capture );
        m_CvImage.CopyOf(image);
        CRect viewSize;
        m_sView.GetClientRect(&viewSize);
        m_CvImage.DrawToHDC( m_sView.GetDC()->GetSafeHdc() ,viewSize);
    }
    CDialogEx::OnTimer(nIDEvent);
}

```

#### 4. OnOK()구현

"viewState"가 "FALSE"일 때(영상을 뿌려주고 있지 않을 때)는 Caption을 "중지"로 바꾸고, 타이머를 동작시킵니다. 반대로 "TRUE"일 때(영상을 뿌려주고 있을 때)는 Caption을 "시작"으로 바꾸고, 타이머를 중지시킵니다. 코드는 다음과 같습니다.

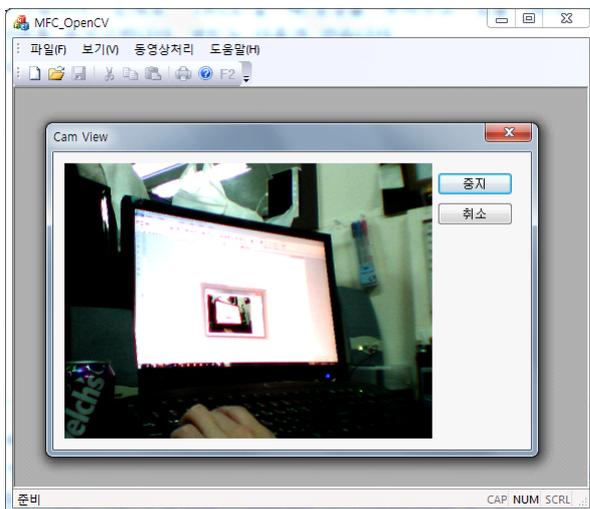
```
void CViewDlg::OnOK()
{
    // TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.
    if(viewState){
        KillTimer(1);
        viewState = FALSE;
        m_btnOK.SetWindowText(TEXT("시작"));
    }else{
        SetTimer(1,((double)1/12)*1000,NULL);
        viewState = TRUE;
        m_btnOK.SetWindowText(TEXT("중지"));
    }
}
```

#### 5. 메뉴에 추가

“동영상 처리” → “영상 보기”메뉴를 추가합니다. 역시 잘 기억나지 않으시는 분은 ["RGB 컬러 공간 분리 - 메뉴 기반 구현\(http://deios.kr/407#AddMenu\)"](http://deios.kr/407#AddMenu)을 참고하세요~

“영상 보기”의 ID는 “ID\_CAM\_VIEW”로 하겠습니다. 아~ 물론 코드는 다음과 같겠죠?  
여기서 중요한 점은 **App Class에서 작성해야 한다**는 것입니다.

```
void CMFC_OpenCVApp::OnCamView()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    CViewDlg dlg;
    dlg.DoModal();
}
```



짹짹~!