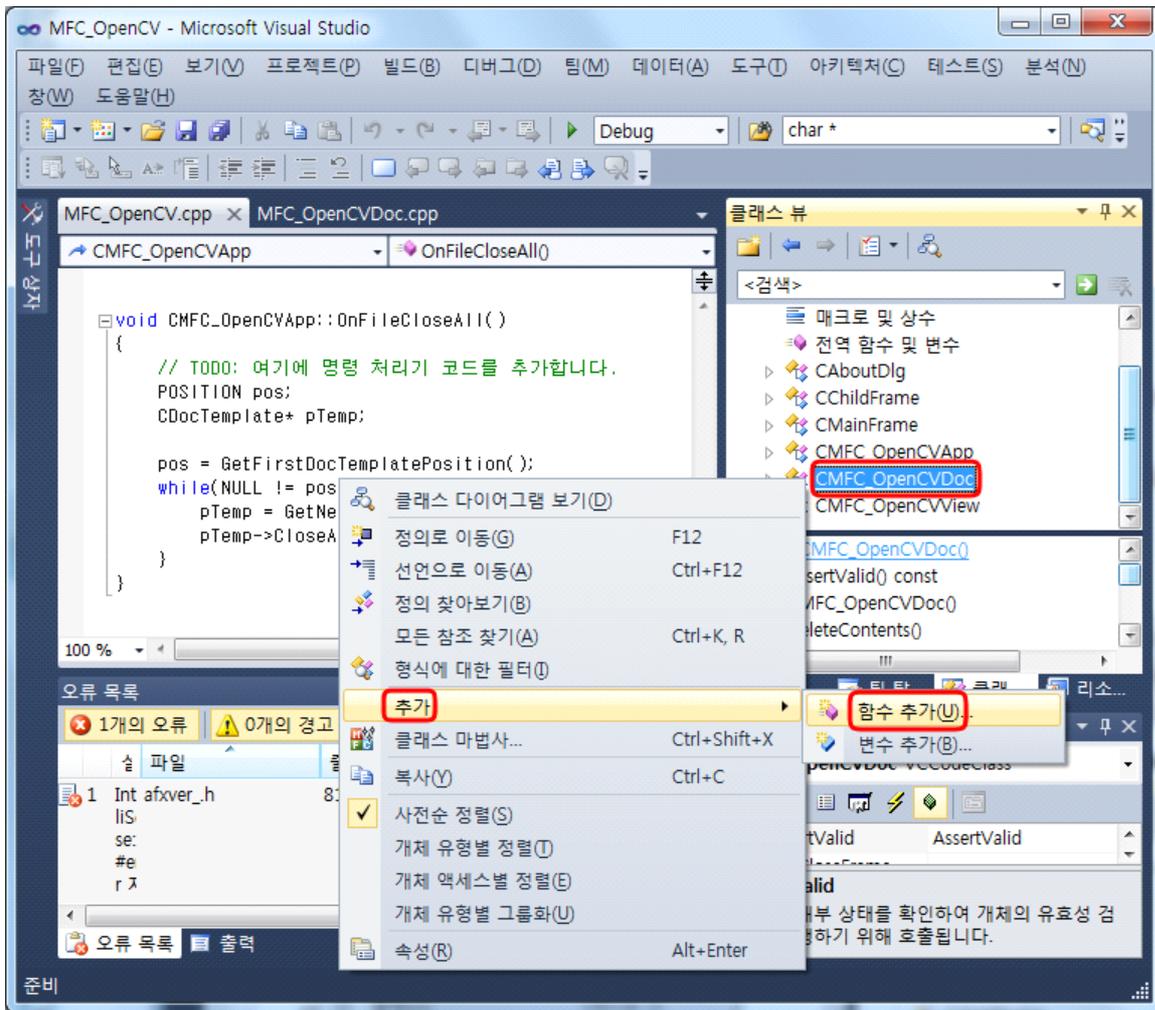


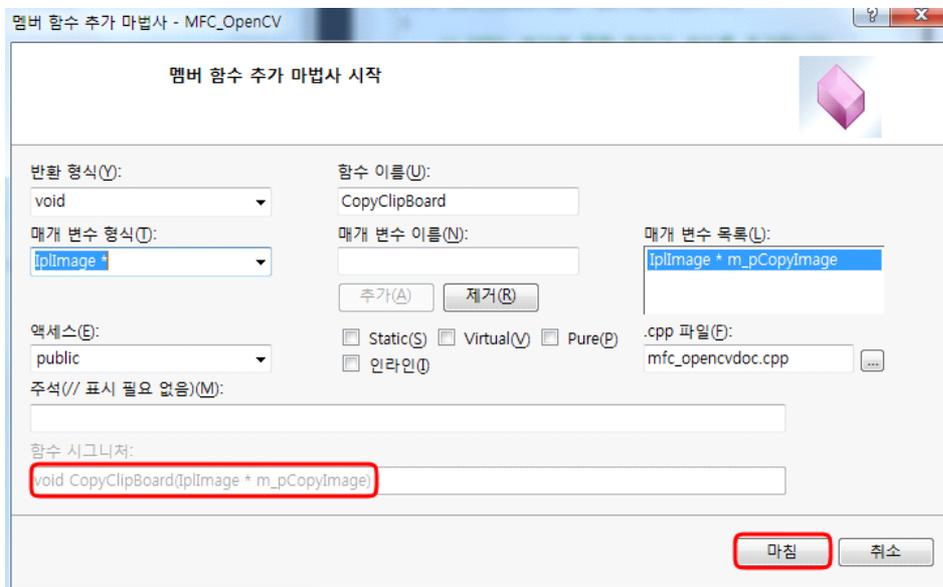
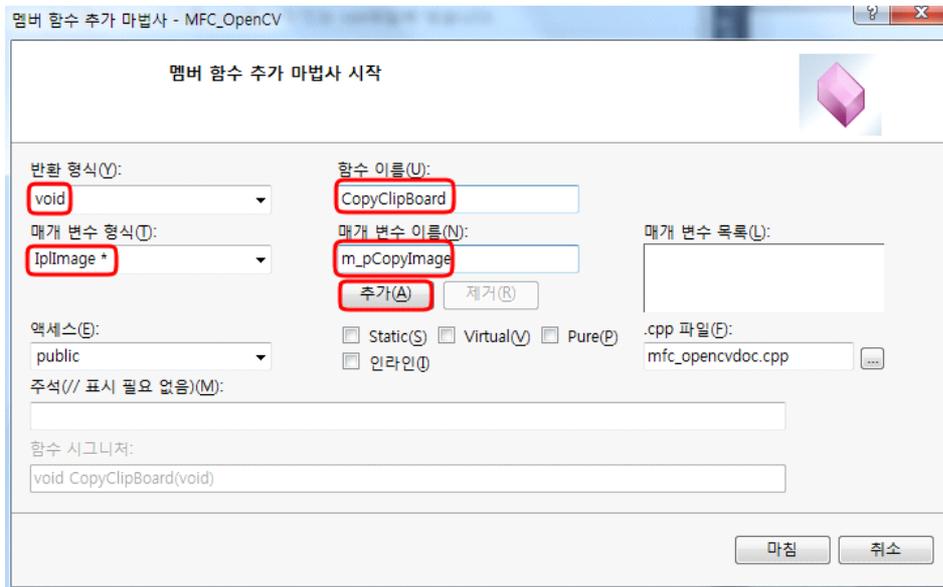
클립보드 기능 구현

<http://deios.kr>

이번에는 자주 사용하는 기능인 C&P(Copy & Paste)를 구현해 보도록 하겠습니다. C&P를 구현하기 위해서는 이미지를 DIB(장치 독립 비트맵;Device Independent Bitmap)으로 변환하여 클립보드 영역에 넘겨주거나, 클립보드 영역에서 데이터를 받아오면 됩니다. 클립보드 영역으로 데이터를 넘겨주기 위하여 CopyClipboard멤버함수, CopyToHandle 멤버 함수를 추가하여, ID_EDIT_COPY이벤트 핸들러에 등록합니다. 먼저 CopyClipboard 멤버 함수를 Doc Class에 작성하겠습니다.



반환 형식 : void
함수 이름 : CopyClipboard
매개 변수 형식 : IplImage *
매개 변수 이름 : m_pCopyImage



다음과 같이 코딩합니다.

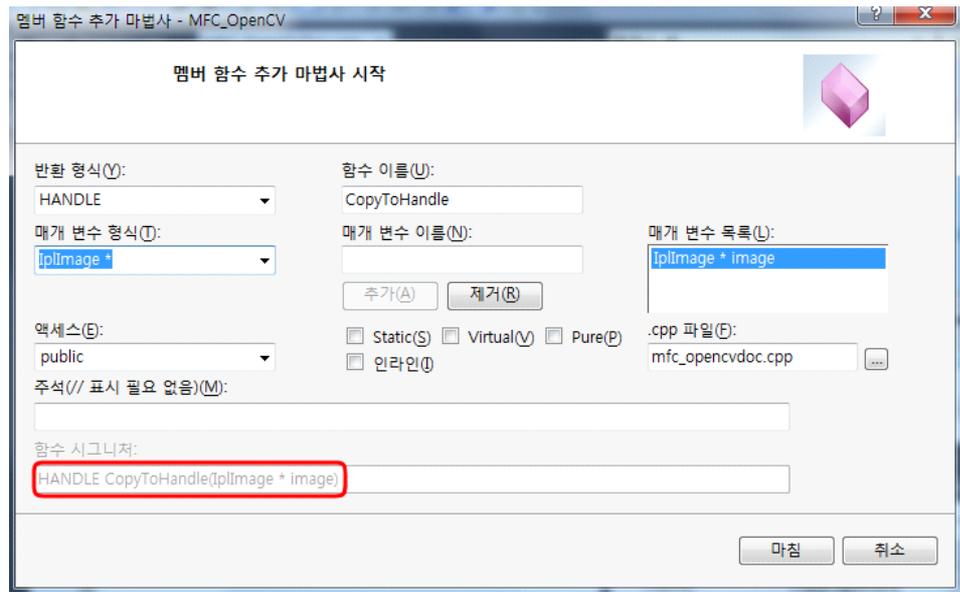
```

HANDLE hDIB = CopyToHandle(m_pCopyImage);
if(::OpenClipboard(AfxGetMainWnd()->GetSafeHwnd())){
    if(::EmptyClipboard()){
        if(NULL == ::SetClipboardData(CF_DIB, hDIB)){
            AfxMessageBox(TEXT("클립보드에 영상을 저장하는 도중 오류가
            발생했습니다.));
        }
    }
}
::CloseClipboard();

```

이번에는 아까와 같은 방법으로 CopyToHandle 멤버 함수를 작성합니다.

반환 형식 : HANDLE
 함수 이름 : CopyToHandle
 매개 변수 형식 : IplImage *
 매개 변수 이름 : image



다음과 같이 코딩합니다. (출처 : 오픈소스 OpenCV를 이용한 컴퓨터 비전 실무 프로그래밍 p237)

```
int nChannels = image->nChannels;
int bpp = 8*nChannels;
IplImage *flip_image = cvCloneImage( image );
cvFlip( image, flip_image, 0 );
int bmpDataSize = flip_image->imageSize;
// 비트맵 헤더 구성
BITMAPINFOHEADER bmpHeader;
bmpHeader.biSize = sizeof(BITMAPINFOHEADER); // 이 구조체의 크기
bmpHeader.biHeight = image->height; // 영상의 높이
bmpHeader.biWidth = image->width; // 영상의 너비
bmpHeader.biPlanes = 1; // 비트 플레인 수 (항상 1임)
bmpHeader.biBitCount = bpp; // 한 화소당 비트 개수
bmpHeader.biCompression = BI_RGB; // BI_RGB : 압축하지 않음
bmpHeader.biSizeImage = bmpDataSize; // 영상의 크기
if(8 == bpp) bmpHeader.biClrUsed = bmpHeader.biClrImportant = 256;
else if(24 == bpp) bmpHeader.biClrUsed = bmpHeader.biClrImportant = 0;
int bmpAllSize = sizeof(BITMAPINFOHEADER) + sizeof(RGBQUAD)*256 +
bmpDataSize*sizeof(char);
HGLOBAL hDIB = (HGLOBAL)::GlobalAlloc(GMEM_MOVEABLE|
GMEM_ZEROINIT,
```

```

bmpAllSize );
    if(NULL == hDIB) return NULL;
    LPSTR pDIB = (LPSTR)::GlobalLock((HGLOBAL)hDIB);
    memcpy( pDIB, &bmpHeader, sizeof(BITMAPINFOHEADER) );

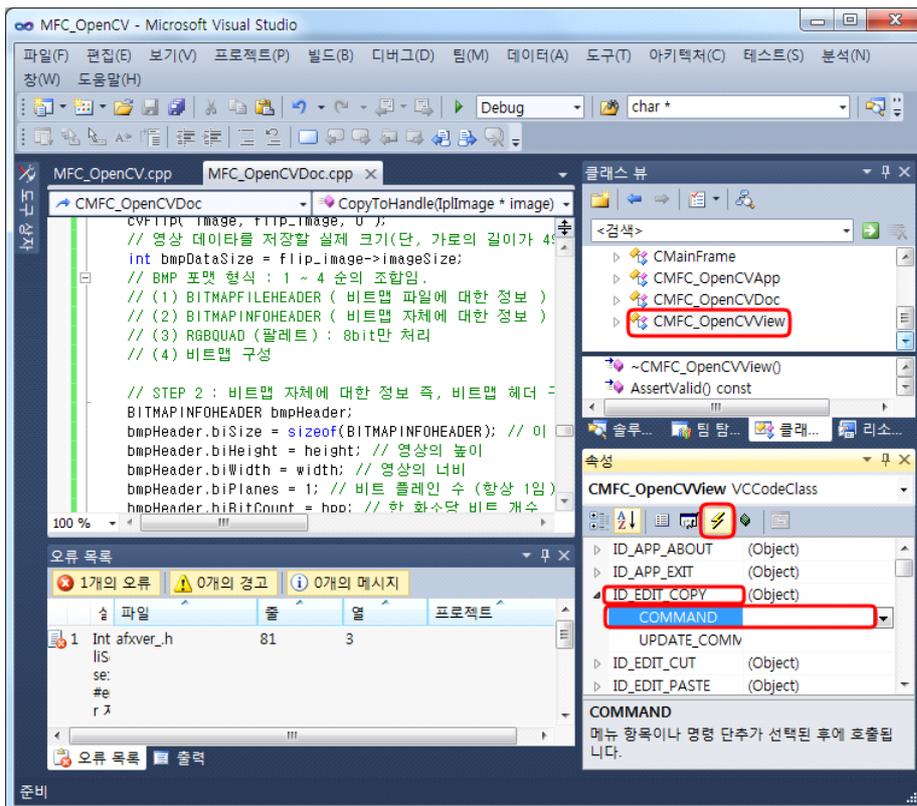
    if(8 == bpp){
        RGBQUAD palette[256];
        memcpy( pDIB+sizeof(BITMAPINFOHEADER), palette, sizeof(RGBQUAD)*256
);

        memcpy( pDIB+sizeof(BITMAPINFOHEADER)+sizeof(RGBQUAD)*256,
                flip_image->imageData, bmpDataSize );
    }
    else if(24 == bpp){
        memcpy( pDIB+sizeof(BITMAPINFOHEADER),
                flip_image->imageData, bmpDataSize );
    }
    ::GlobalUnlock((HGLOBAL)hDIB);
    cvReleaseImage( &flip_image );

    return hDIB;

```

이제 ID_EDIT_COPY에 대한 이벤트 핸들러만 수정해 주면 끝입니다.
이벤트 핸들러를 수정하기 위해 다음과 같이 합니다.

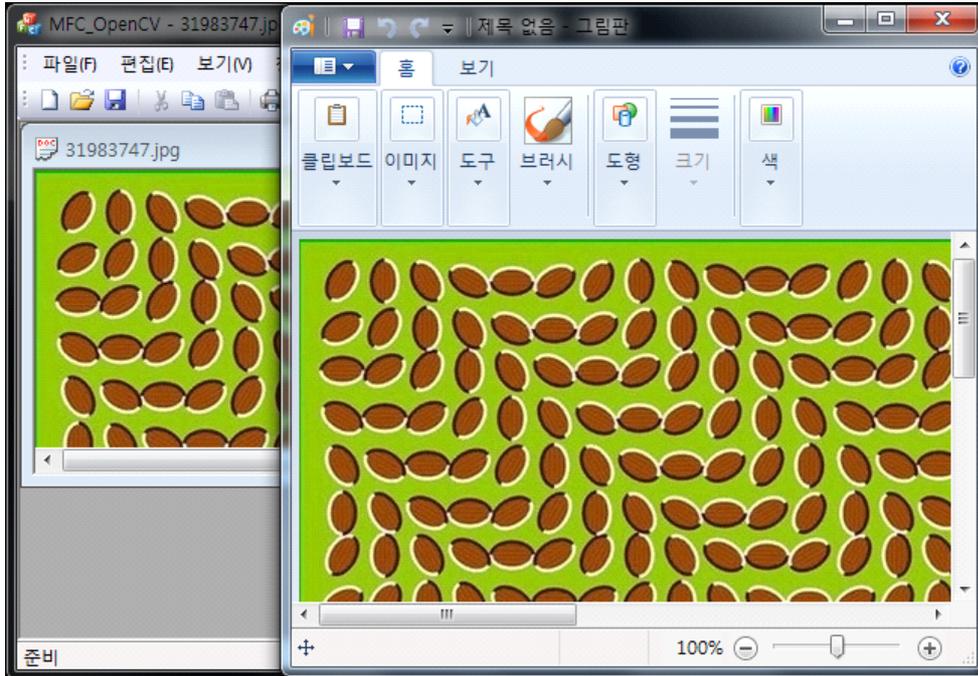


```

void CMFC_OpenCVView::OnEditCopy()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    CMFC_OpenCVDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    pDoc->CopyClipboard( pDoc->m_CvvImage.GetImage() );
}

```

프로그램에서 Copy하여 그림판 같은 곳에 Paste했을 때 정상적으로 동작한다면 성공.



이번에는 Paste기능을 구현해 보도록 하겠습니다. Paste는 Copy의 역순입니다.

CreateFromHandle멤버 함수를 작성하고, ID_EDIT_PASTE에 이벤트 핸들러를 등록하고, 마지막으로 클립보드 영역에 DIB 데이터가 존재할 경우에만 ID_EDIT_PASTE가 동작하도록 하기 위해서 OnUpdateEditPaste 멤버 함수를 작성하겠습니다. 이번에는 모든 함수가 App Class에서 작성될 것이고, 코드만 첨부합니다.

```

void CMFC_OpenCVApp::OnEditPaste()
{
    // TODO: 여기에 명령 처리기 코드를 추가합니다.
    POSITION pos = GetFirstDocTemplatePosition();
    CDocTemplate *pTemplate = GetNextDocTemplate(pos);
    CMFC_OpenCVDoc *pDoc = (CMFC_OpenCVDoc*
)pTemplate->OpenDocumentFile(NULL);

    if( pDoc ){
        HANDLE hDIB = NULL;
        if (::OpenClipboard(AfxGetMainWnd()->GetSafeHwnd())) hDIB =

```

```

::GetClipboardData(CF_DIB);

    if( hDIB ){
        IplImage *image = CreateFromHandle( hDIB );

        pDoc->m_CvvImage.CopyOf( image, image->nChannels*8 );

        POSITION pos = pDoc->GetFirstViewPosition();
        CMFC_OpenCVView *pView = (CMFC_OpenCVView
*)pDoc->GetNextView(pos);

        CSize sizeTotal = CSize(pDoc->m_CvvImage.Width(),
pDoc->m_CvvImage.Height());

        pView->SetScrollSizes(MM_TEXT, sizeTotal);
        pView->ResizeParentToFit(FALSE);
    }

    ::CloseClipboard();
}
}

```

```

IplImage * CMFC_OpenCVApp::CreateFromHandle(HANDLE hDIB)
{
    LPSTR pDIB = (LPSTR)::GlobalLock((HGLOBAL)hDIB);
    BITMAPINFOHEADER bmpHeader;
    memcpy( &bmpHeader, pDIB, sizeof(BITMAPINFOHEADER) );

    int height = bmpHeader.biHeight;
    int width = bmpHeader.biWidth;
    int nChannels = bmpHeader.biBitCount/8;
    int bmpDataSize = bmpHeader.biSizeImage;
    char *imageData = (char *)calloc( bmpDataSize, sizeof(char) );

    if(8 == bmpHeader.biBitCount)
        memcpy( imageData, pDIB+sizeof(BITMAPINFOHEADER)+sizeof(RGBQUAD)*256,
bmpDataSize );
    else if(24 == bmpHeader.biBitCount)
        memcpy( imageData, pDIB+sizeof(BITMAPINFOHEADER), bmpDataSize );

    IplImage *flip_image = cvCreateImage(cvSize(width,height), IPL_DEPTH_8U, nChannels);
    memcpy( flip_image->imageData, imageData, bmpDataSize );

    IplImage *recover_image = cvCloneImage( flip_image );
}

```

```

cvFlip( flip_image, recover_image, 0 );

::GlobalUnlock((HGGLOBAL)hDIB);

free(imageData);
cvReleaseImage( &flip_image );

return recover_image;
}

```

```

void CMFC_OpenCVApp::OnUpdateEditPaste(CCmdUI *pCmdUI)
{
    // TODO: 여기에 명령 업데이트 UI 처리기 코드를 추가합니다.
    if( !IsClipboardFormatAvailable(CF_DIB) ) pCmdUI->Enable( FALSE );
}

```

App Class의 헤더 파일에 #include <cv.h>를 추가해 줘야 합니다.

