



# Image Processing

---

## ❖ 이미지 BMP, JPG 로딩

담당교수: 김 은 이

연락처 : 450-4135, [eykim@konkuk.ac.kr](mailto:eykim@konkuk.ac.kr)

# 목차

---

## □ 프로젝트 생성 및 설정

### ■ VS2015 기준

## □ 이미지 로드

### ■ BMP

### ■ JPG

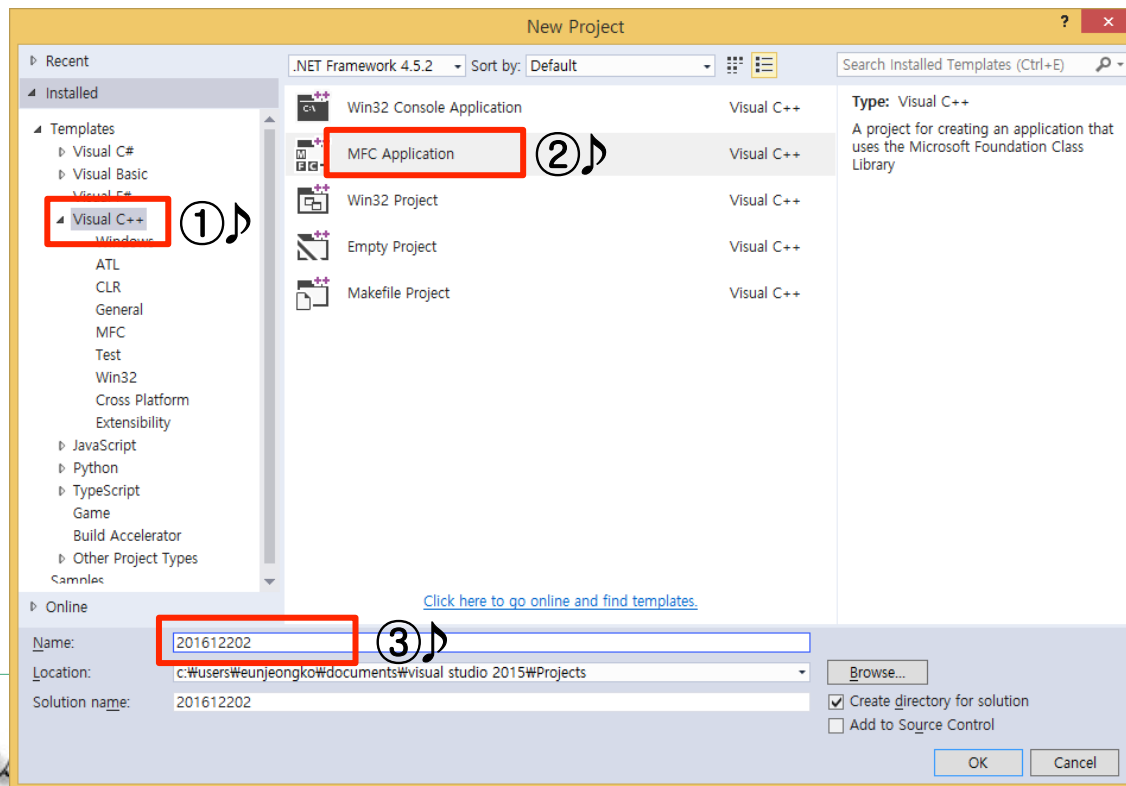
## □ RGB to HSI

---



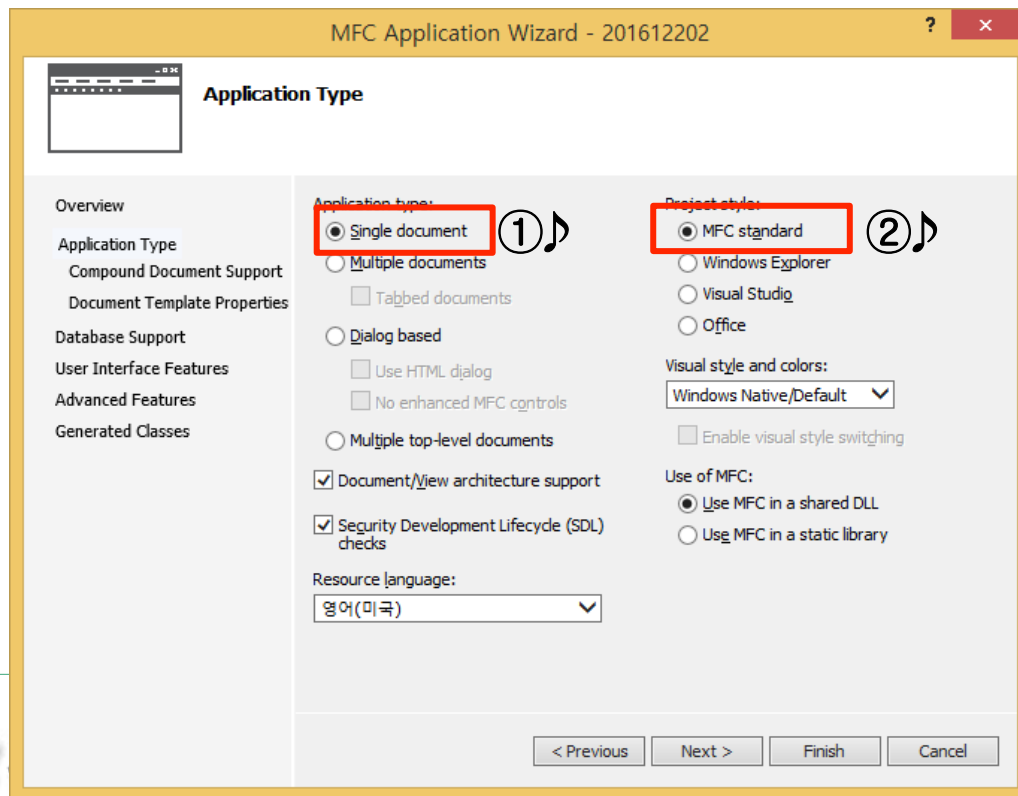
# 프로젝트 생성 및 설정

- Visual Studio 2015에서 프로젝트 생성
  - MFC 응용 프로그램으로 생성



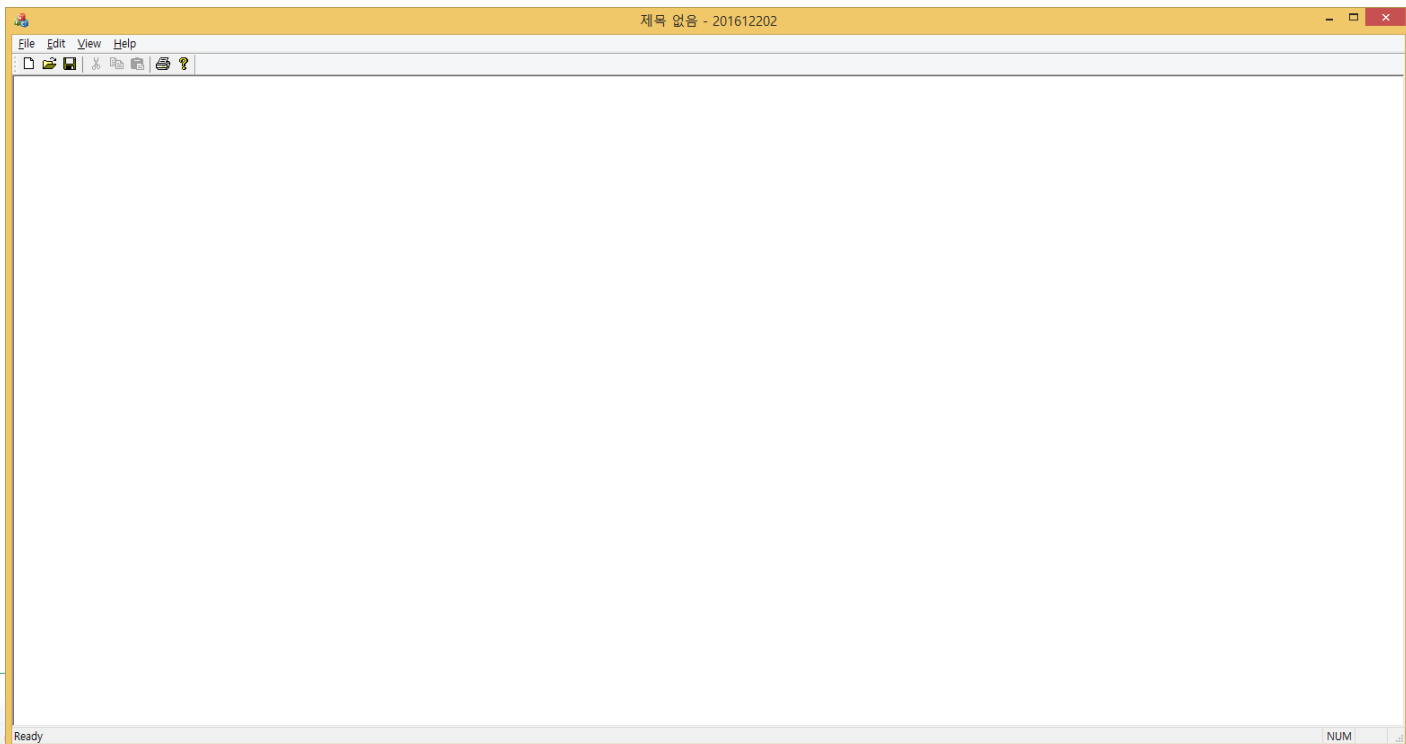
# 프로젝트 생성 및 설정

- Visual Studio 2015에서 프로젝트 생성
  - MFC 응용 프로그램으로 생성



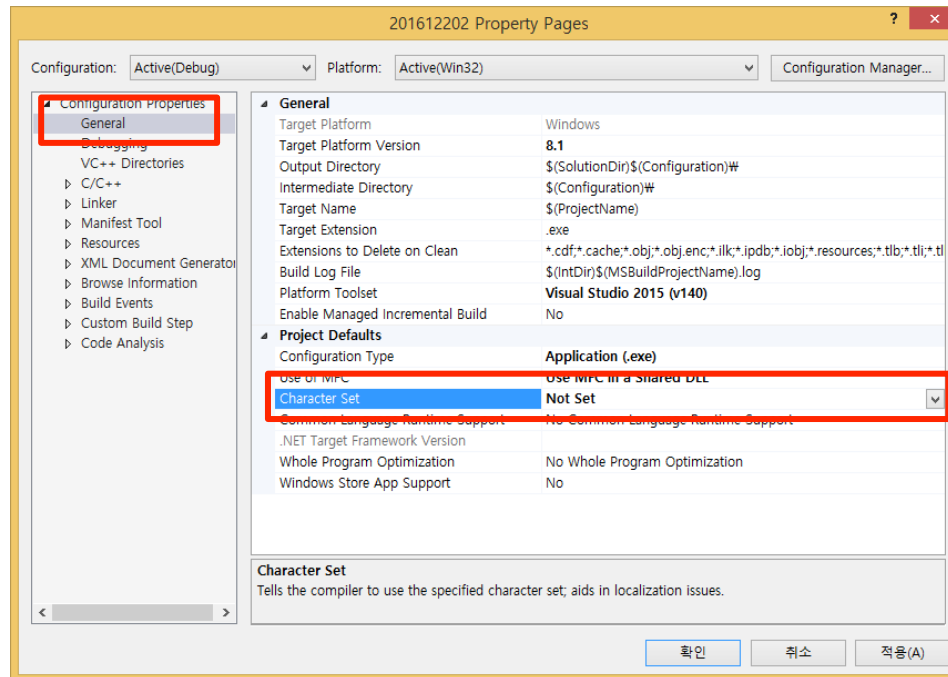
# 프로젝트 생성 및 설정

- Visual Studio 2015에서 프로젝트 생성
  - 실행 시, 아래와 같이 비어있는 프로그램 실행됨



# 프로젝트 설정

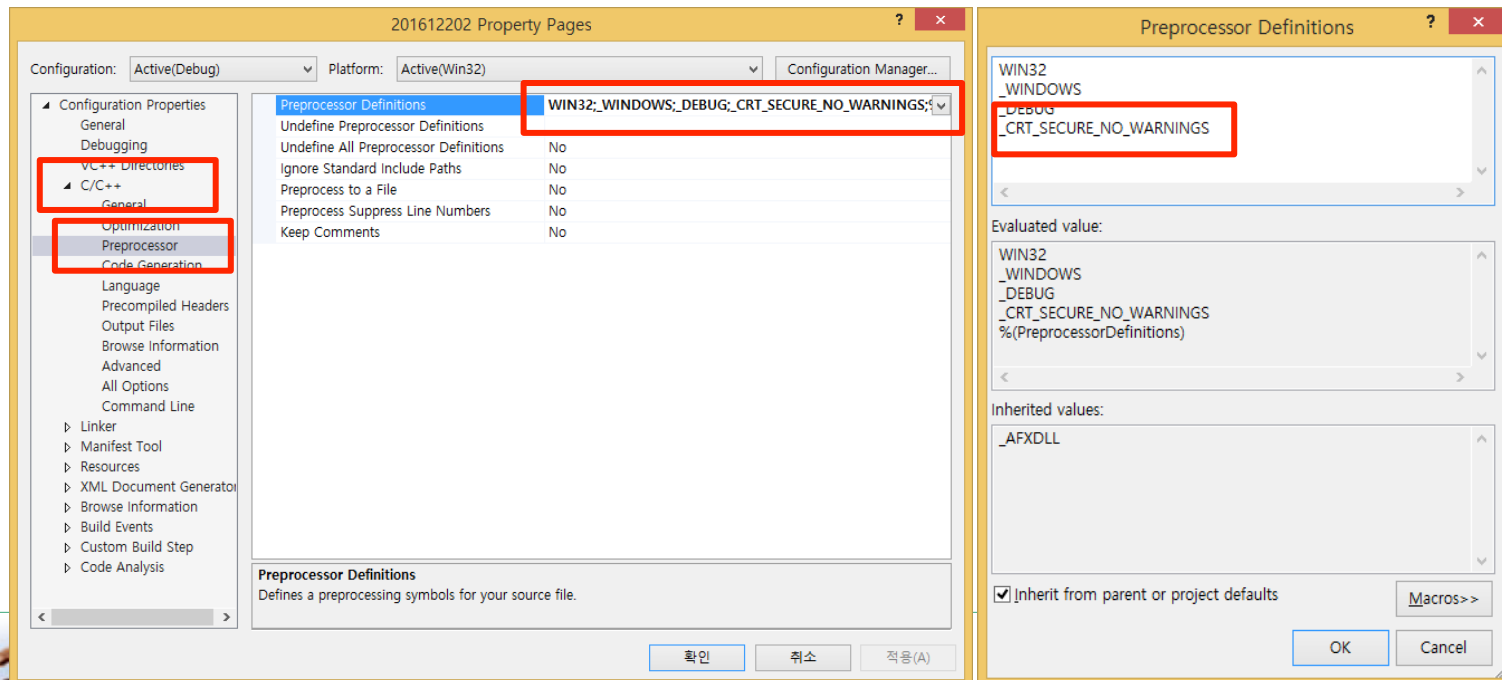
- 상단 메뉴에서 “프로젝트” → 속성
  - 일반 → 유니코드 사용 안함



# 프로젝트 설정

□ 상단 메뉴에서 “프로젝트” → 속성

■ C/C++ → 전처리기 → ‘\_CRT\_SECURE\_NO\_WARNINGS’ 입력



# 이미지 로드, BMP 로드 메뉴 생성

## □ 리소스뷰

### ■ 이미지 로드, BMP로드 메뉴 생성

The image shows a sequence of three screenshots from Visual Studio illustrating the steps to create a menu item for loading a BMP image.

**Screenshot 1: Solution Explorer**  
The 'Solution Explorer' window shows the project structure. The 'Resource Files' folder is expanded, and the file '201612202.ico' is highlighted with a red box.

**Screenshot 2: Resource View**  
The 'Resource View' window shows the resource files. The 'Menu' folder is expanded, and the item 'IDR\_MAINFRAME [English (United States)]' is highlighted with a red box.

**Screenshot 3: Menu Editor**  
The 'Menu Editor' window shows the 'File' menu. A new menu item '이미지 로드' (Load Image) is being added. The 'BMP' type is selected, and the text '타이핑하여 입력' (Type here) is shown with an arrow pointing to the input field.

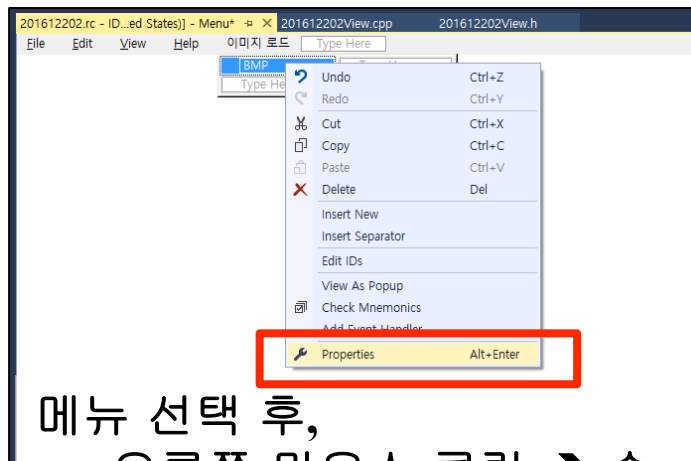




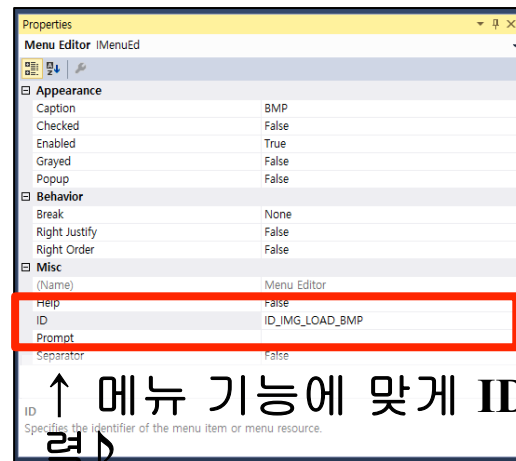
# 이미지 로드, BMP 로드 메뉴 생성

## □ 리소스뷰

### ■ 메뉴 속성 설정



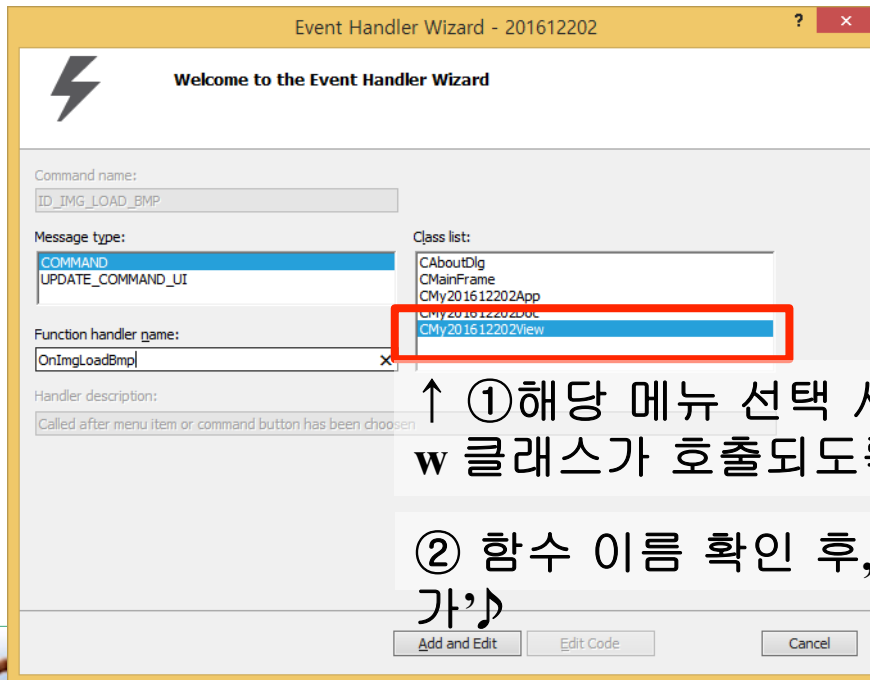
메뉴 선택 후,  
오른쪽 마우스 클릭 → 속  
성



# 이미지 로드, BMP 로드 메뉴 생성

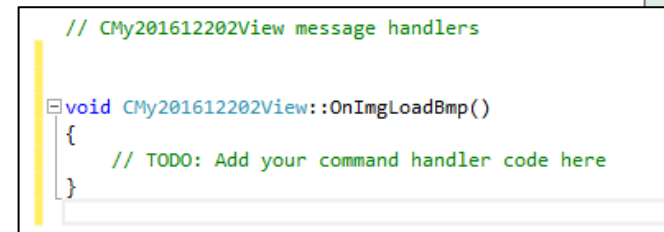
## □ 리소스뷰

- 이벤트 처리기 추가: 메뉴 클릭 시 실행되는 함수 추가
  - View 클래스에 실행되는 함수 추가



↑ ① 해당 메뉴 선택 시, **\*\*View** 클래스가 호출되도록 설정♪

② 함수 이름 확인 후, '추가'♪



↑ 생성된 멤버 함수 확인♪

# 이미지 정보를 저장할 변수 추가

## □ View 클래스에 멤버 변수 추가

- RGBQUAD\*\* rgbBuffer; //이미지를 저장할 변수
- Int imgHeight; //이미지 높이 정보
- int imgWidth; //이미지 너비 정보
- BITMAPINFOHEADER bmpInfo; //BMP 정보 헤더
- BITMAPFILEHEADER bmpHeader; //BMP 파일 헤더

→ 동적 할당하는 포인터 배열은 반드시 '생성자'와 '소멸자'에 NULL로 초기화 및 해제를 수행

```

CMy201612202View::CMy201612202View()
{
    // TODO: add construction code here
    rgbBuffer = nullptr;
}

CMy201612202View::~CMy201612202View()
{
    if (rgbBuffer != nullptr)
    {
        for (int i = 0; i < imgHeight; i++)
            delete[] rgbBuffer[i];
        delete[] rgbBuffer;
    }
}
    
```



# BMP 로드 함수 작성 (1/3)

```
//메뉴 클릭 시 호출되는 함수에 작업
void Viewclass이름::OnBmpLoad()
{
    //1. 파일 다이얼로그로부터 BMP 파일 입력
    CFileDialog dlg(TRUE, ".bmp", NULL, NULL, "Bitmap File (*.bmp)|*.bmp|");
    if(IDOK != dlg.DoModal())
        return;
    CString filename = dlg.GetPathName();
    if( rgbBuffer!=NULL ) { //이미 할당된 경우, 메모리 해제
        for(int i = 0; i<imgHeight; i++)
            delete [] rgbBuffer[i];
        delete [] rgbBuffer; }

    //2. 파일을 오픈하여 영상 정보 획득
    CFile file;
    file.Open( filename, CFile::modeRead);
    file.Read( &bmpHeader, sizeof( BITMAPFILEHEADER ) );
    file.Read( &bmpInfo, sizeof( BITMAPINFOHEADER));
    imgWidth = bmpInfo.biWidth;
    imgHeight = bmpInfo.biHeight;
```



# BMP 로드 함수 작성 (2/3)

```
//3. 이미지를 저장할 버퍼 할당 (2차원 배열) [이미지 높이 * 이미지 너비 만큼 할당]
rgbBuffer = new RGBQUAD*[imgHeight];
    for(int i = 0; i<imgHeight; i++)
        rgbBuffer[i] = new RGBQUAD[imgWidth];
//4. 이미지의 너비가 4의 배수인지 체크
// BMP조건 가로는 4byte씩 이어야 한다.
// 한 픽셀이3바이트(R,G,B)씩이니깐. 가로(m_width) * 3이4의 배수인가 아닌가를 알아야한다.
// b4byte : 4byte배수인지 아닌지를 안다.
// upbyte : 4byte배수에 모자라는 바이트다.
bool b4byte = false;
int upbyte = 0;
if( (imgWidth*3) % 4 == 0) {
    // 4의배수로떨어지는경우.
    b4byte = true;
    upbyte = 0; }
else {
    // 4의배수로떨어지지않는경우.
    b4byte = false;
    upbyte = 4 - (imgWidth*3) % 4; }
```



# BMP 로드 함수 작성 (3/3)

```
//5. 픽셀 데이터를 파일로부터 읽어옴
BYTE data[3];
for(int i = 0 ; i < imgHeight; i++) {
    for(int j = 0 ; j < imgWidth; j++) {
        file.Read( &data, 3 );
        //이미지가 거꾸로 저장되어 있기 때문에 거꾸로 읽어옴
        rgbBuffer[imgHeight-i-1][j].rgbBlue = data[0];
        rgbBuffer[imgHeight-i-1][j].rgbGreen = data[1];
        rgbBuffer[imgHeight-i-1][j].rgbRed = data[2];
    }
    if( b4byte == false )
    {
        // 가로가4byte배수가아니면쓰레기값을읽는다.
        file.Read( &data, upbyte );
    }
}
file.Close(); //파일 닫기
Invalidate(TRUE); //화면 갱신
```



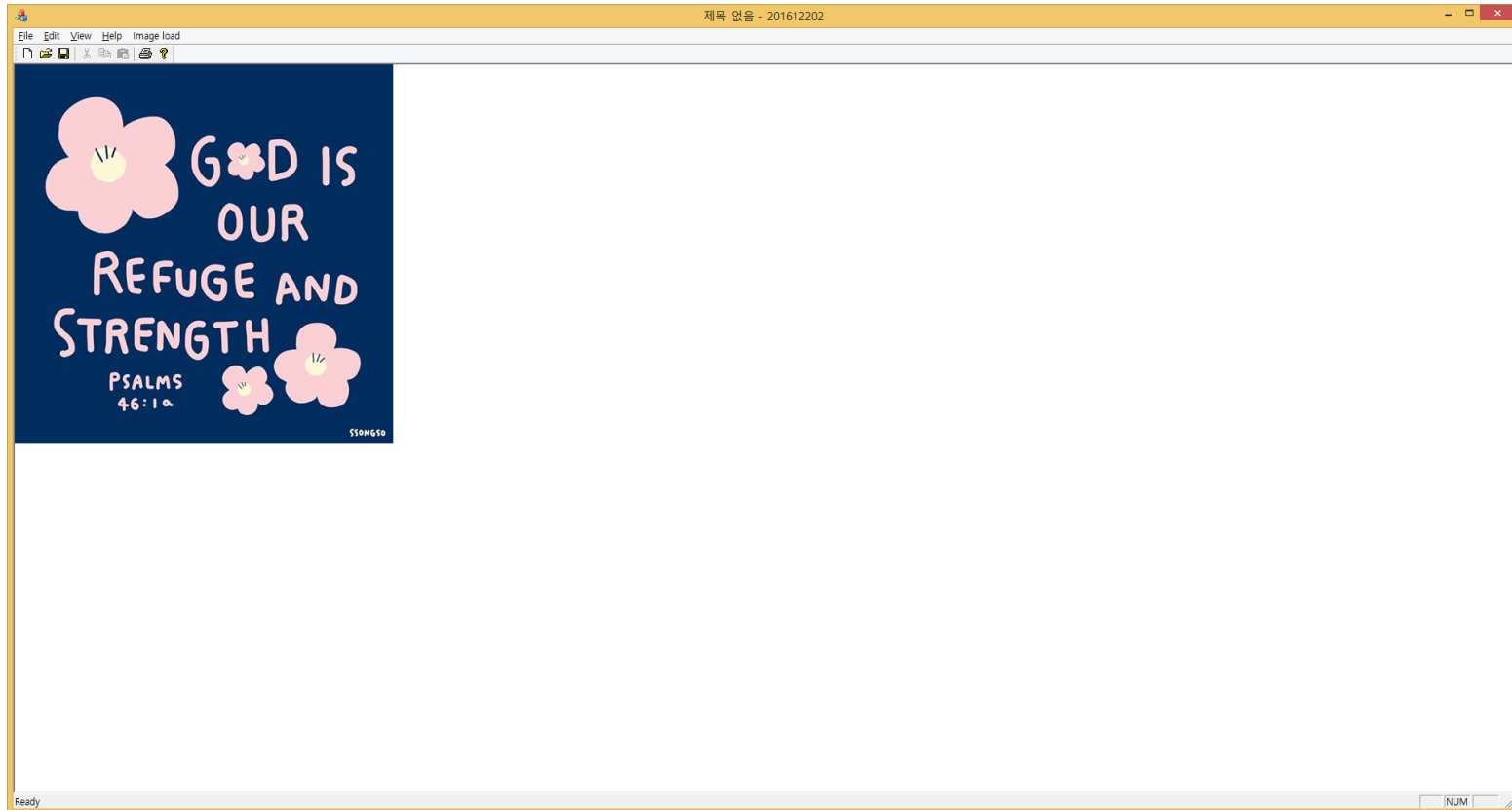
# 이미지 화면에 출력

## □ View 클래스 OnDraw함수를 다음과 같이 수정

```
void View클래스 이름::OnDraw(CDC* pDC)
{
    CIM_LECTUREDDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc); //기존의 것을 그대로 사용, 수정할 필요 없음
    if (!pDoc)
        return;
    if(rgbBuffer != NULL){ //이미지가 들어 있는 경우에 출력
        for(int i = 0; i<imgHeight; i++){ //이미지 사이즈 만큼 돌면서 한 픽셀씩 출력
            for(int j = 0; j<imgWidth; j++){
                POINT p; //출력할 픽셀의 위치 지정
                p.x = j;
                p.y = i;
                pDC->SetPixel(p, RGB(rgbBuffer[i][j].rgbRed, rgbBuffer[i][j].rgbGreen, rgbBuffer[i][j].rgbBlue));
                //픽셀 위치 p에 RGB값을 출력
            }
        }
    }
}
```



# BMP 로드 결과





# RGB to HSI 함수 작성 (1/3)

1. rgbBuffer에 이미지가 들어있는 지 여부 확인

```
if(rgbBuffer == NULL)
```

```
    OnBmpLoad(); //rgbBuffer에 데이터가 없는 경우, 로드 함수를 호출하여 이미지 획득
```

2. 변수 메모리 할당

```
hueBuffer = new float*[imgHeight];
```

```
satuBuffer = new float*[imgHeight];
```

```
intenBuffer = new float*[imgHeight];
```

```
for(int i = 0; i<imgHeight; i++)
```

```
{
```

```
    hueBuffer[i] = new float[imgWidth];
```

```
    satuBuffer[i] = new float[imgWidth];
```

```
    intenBuffer[i] = new float[imgWidth];
```

```
}
```



# RGB to HSI 함수 작성 (2/3)

## 3. RGB to HSI 값 변환

```
for(int i = 0; i<imgHeight; i++){
    for(int j = 0; j<imgWidth; j++) {
        float r = rgbBuffer[i][j].rgbRed;
        float g = rgbBuffer[i][j].rgbGreen;
        float b = rgbBuffer[i][j].rgbBlue;
        intenBuffer[i][j] = (r+g+b)/(float)(3*255); //intensity
        float total = r+g+b;
        r = r/total; g = g/total; b = b/total;
        satuBuffer[i][j] = 1- 3*(r>g?(g>b?b:g):(r>b?b:r));
        if(r == g && g == b) {
            hueBuffer[i][j] = 0; satuBuffer[i][j] = 0; }
        else {
            total = (0.5*( r-g + r-b )/sqrt( (r-g)*(r-g) + (r-b)*(g-b) ));
            hueBuffer[i][j] = acos((double)total);
            if(b > g){
                hueBuffer[i][j] = 6.28 - hueBuffer[i][j]; }
        }
    }
}
```



# RGB to HSI 함수 작성 (3/3)

4. 출력 값 범위 정규화: 출력 시, 값의 범위를 [0, 255]로 맞춰줌

```
for(int i = 0; i<imgHeight; i++){
    for(int j = 0; j<imgWidth; j++)
    {
        hueBuffer[i][j] = hueBuffer[i][j]*255/(3.14*2);
        satuBuffer[i][j] = satuBuffer[i][j]*255;
        intenBuffer[i][j] = intenBuffer[i][j]*255;
    }
}
```

5. 출력

```
viewType = 2;
Invalidate(FALSE);
```



# 이미지 화면에 출력

## □ View 클래스 OnDraw함수를 다음을 추가

```
if(rgbBuffer != NULL){
    for(int i = 0; i<imgHeight; i++){
        for(int j = 0; j<imgWidth; j++){ //rgbBuffer를 출력했던 2중 For문

            if(viewType == 2) {
                p.x = j+ imgWidth + 10;
                p.y = i;
                pDC->SetPixel(p, RGB(hueBuffer[i][j], hueBuffer[i][j], hueBuffer[i][j]));

                p.x = j+ imgWidth + 10;
                p.y = i+ imgHeight + 10;
                pDC->SetPixel(p, RGB(satuBuffer[i][j], satuBuffer[i][j], satuBuffer[i][j]));

                p.x = j;
                p.y = i+ imgHeight + 10;
                pDC->SetPixel(p, RGB(intenBuffer[i][j], intenBuffer[i][j], intenBuffer[i][j]));
            }
        }
    }
}
```

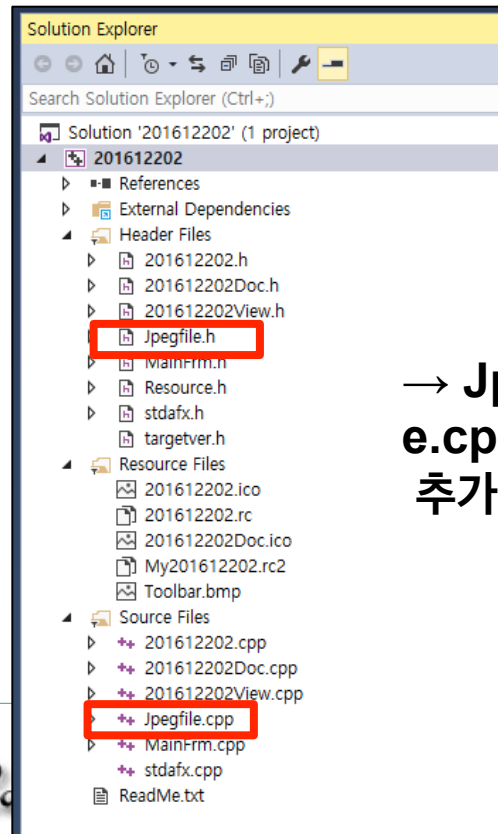


# JPG 로드를 위한 프로젝트 설정

□ 라이브러리 및 헤더, 소스 파일들을 프로젝트 폴더 내에 복사

내 PC > 문서 > Visual Studio 2015 > Projects > 201612202 > 201612202

이름	수정된 날짜	유형	크기
Debug	2017-02-06 오후...	파일 폴더	
res	2017-02-06 오후...	파일 폴더	
201612202.apr	2017-02-06 오후...	APS 파일	122KB
201612202	2017-02-06 오후...	CPP 파일	5KB
201612202	2017-02-06 오후...	C/C++ Header	1KB
201612202	2017-02-06 오후...	Resource Script	13KB
201612202	2017-02-06 오후...	VC++ Project	11KB
201612202.vcxproj	2017-02-06 오후...	VC++ Project Fil...	3KB
201612202Doc	2017-02-06 오후...	CPP 파일	3KB
201612202Doc	2017-02-06 오후...	C/C++ Header	1KB
201612202View	2017-02-06 오후...	CPP 파일	5KB
201612202View	2017-02-06 오후...	C/C++ Header	2KB
JCONFIG	2008-12-11 오후...	C/C++ Header	2KB
Jmorecfg	2008-12-11 오후...	C/C++ Header	13KB
Jpegfile	2008-12-11 오후...	CPP 파일	31KB
Jpegfile	2008-12-11 오후...	C/C++ Header	7KB
JPEGLIB	2008-12-11 오후...	C/C++ Header	45KB
Jpeglib2	2008-12-11 오후...	Object File Library	770KB
MainFrm	2017-02-06 오후...	CPP 파일	5KB
MainFrm	2017-02-06 오후...	C/C++ Header	1KB
ReadMe	2017-02-06 오후...	텍스트 문서	7KB
resource	2017-02-06 오후...	C/C++ Header	1KB
stdafx	2017-02-06 오후...	CPP 파일	1KB
stdafx	2017-02-06 오후...	C/C++ Header	2KB
targetver	2017-02-06 오후...	C/C++ Header	1KB



→ Jpegfile.h / Jpegfile.cpp 파일을 프로젝트에 추가



# JPG 로드를 위한 프로젝트 설정

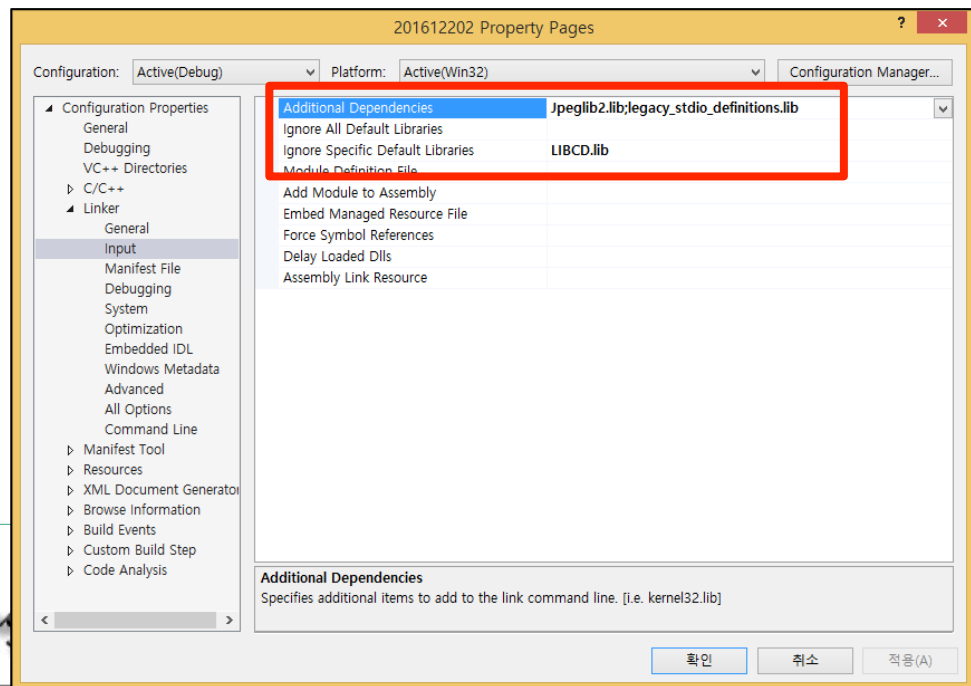
## □ 프로젝트 ➔ 속성 ➔ 링커 ➔ 입력

### ■ 추가 종속성

↓ VS2015 이상의 경우

□ Jpeglib2.lib; legacy\_stdio\_definitions.lib 입력

### ■ 특정 라이브러리 무시: LIBCD.lib 입력



# JPG 로드를 위한 프로젝트 설정

- JpegFile.h를 뷰클래스 헤더파일에 include

```

Microsoft.CppCommon.targets  jpegfile.h  201612202.rc - ID...ed States]] - Menu  201612202View.cpp  201612202View.h
201612202  CMy201612202View
1
2  // 201612202View.h : interface of the CMy201612202View class
3  //
4
5  #pragma once
6  #include "Jpegfile.h"
7

```



# JPG 로드 메뉴 추가 및 이벤트 처리 생성

---

- BMP와 동일한 방식으로 리소스뷰에 메뉴 추가
- 메뉴 ID 설정 및 이벤트 핸들러를 이용하여 호출될 함수 추가





# JPG 로드를 위한 코드 작성(1/2)

///JPG 메뉴 호출 함수 전에 아래의 함수 추가

```
BYTE * LoadJpegFromOpenFile(FILE *fp, BITMAPINFOHEADER *pbh, UINT *pWidth, UINT *pHeight) {  
//파일로부터 JPG 정보 획득
```

```
    if( pWidth==NULL || pHeight==NULL) return NULL;
```

```
    BYTE *tmp = JpegFile::OpenFileToRGB( fp, pWidth, pHeight);
```

```
    if (!tmp) return NULL;
```

```
        JpegFile::BGRFromRGB(tmp, *pWidth, *pHeight);
```

```
    UINT dw;
```

```
    BYTE *pbuf = JpegFile::MakeDwordAlignedBuf(tmp, *pWidth, *pHeight, &dw);
```

```
    delete [] tmp;
```

```
    if (!pbuf) return NULL;
```

```
    JpegFile::VertFlipBuf( pbuf, dw, *pHeight);
```

```
    pbh->biSize      = sizeof(BITMAPINFOHEADER);
```

```
    pbh->biWidth      = *pWidth;
```

```
    pbh->biHeight     = *pHeight;
```

```
    pbh->biPlanes     = 1;
```

```
    pbh->biBitCount    = 24;
```

```
    pbh->biCompression = BI_RGB;
```

```
    pbh->biSizeImage   = 0;
```

```
    pbh->biXPelsPerMeter = 0;
```

```
    pbh->biYPelsPerMeter = 0;
```

```
    pbh->biClrUsed      = 0;
```

```
    pbh->biClrImportant = 0;
```

```
    return pbuf;
```

```
}
```

# JPG 로드를 위한 코드 작성(2/2)

```
void View클래스 이름::OnJpgLoad()
{
    CFileDialog dlg(TRUE, ".jpg", NULL, NULL, "Jpeg File (*.jpg)|*.jpg|");
    if(IDOK != dlg.DoModal()) return;
    CString filename = dlg.GetPathName();
    //BMP 로드때와 마찬가지로, rgbBuffer가 NULL이 아닌 경우 메모리 해제 코드 추가하기
    FILE *fp = fopen(filename, "rb");
    BITMAPINFOHEADER pbh;
    UINT w,h;
    BYTE *pbuf = LoadJpegFromOpenFile(fp, &pbh, &w, &h);
    imgWidth = (int)w;
    imgHeight = (int)h;
    //BMP로드때와 마찬가지로, rgbBuffer를 이미지 사이즈 만큼 메모리 할당 코드 추가하기
    int dw = WIDTHBYTES(imgWidth*24);
    for(int i = 0 ; i < imgHeight; i++) {
        for(int j = 0 ; j < imgWidth; j++) {
            rgbBuffer[imgHeight-i-1][j].rgbBlue = pbuf[i*dw + j*3 +0];
            rgbBuffer[imgHeight-i-1][j].rgbGreen = pbuf[i*dw + j*3 +1];
            rgbBuffer[imgHeight-i-1][j].rgbRed = pbuf[i*dw + j*3 +2]; } }
    delete [] pbuf;
    fclose(fp);
    Invalidate(TRUE);
}
```

# JPG 로드 결과

