

Basic Image Processing

Microsoft Visual Studio C++ OpenCV

Tokyo Metropolitan College

Yamasho Lab

2019

画像とその処理について

画像について

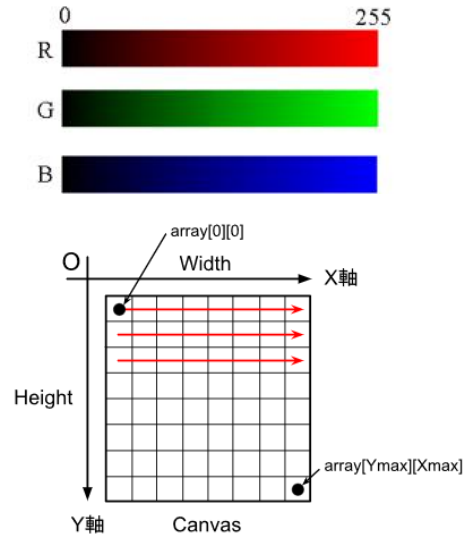
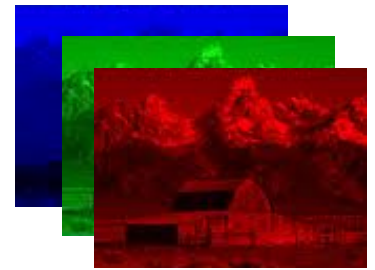
画像は細かく区切られた画素(Pixel)が2次元状に配置された集合である。各画素は通常3色(R, G, B)の画像平面で構成され、それぞれ0~255(8bit)の階調を持つ。

一般的な2次元グラフと異なり、画像原点は左上から数えられる。X軸方向を幅(width), Y軸方向を高さ(height)と呼び、幅×高さが総画素数となる。

コンピュータ上に保存された画像のサイズは通常、
ファイルサイズ = 総画素数 × 色数 × 8bit

で計算される。最も一般的な保存形式はBMP (ビットマップ) であるが、ほぼ上記計算でファイルサイズが一致する。一方、デジタルカメラなどで使用されているファイル形式はJpegであり、約1/10のファイルサイズまで圧縮されている。

保存には圧縮が有効であるが、画像処理などではなるべく劣化していないBMPやRAWを用いるべきである。



処理と利用ライブラリ

画像処理は非常に多くの手法が開発されてきている。これら処理を全て網羅するのは大変な作業であり、それらを有効活用するために、専用のライブラリが開発されている。中でも無償のOpenCVは世界中の技術者が理由して、現在も新しい処理アルゴリズムが追加されている。

そこで、本講義ではOpenCVを利用しながら画像処理の基本を学んでいく。OpenCVはC++言語で作成されているため、Microsoft社のVisual Studio上で統合するのが一般的である。最近ではPython用も提供されているため、Pythonベースで開発することも可能である。

なお、OpenCVは大変便利であるが、最低限、基本となるアルゴリズムは理解すべきと考え、標準C言語での記述方法も併記している部分がある。



Start Up (Setting)

本講義ではWindows10上にインストールされたVisual Studio 2015を統合ソフトウェア+OpenCV1.1preをライブラリとして学習していく。1.1preは古いライブラリであるが、なるべくクラス構造を使わない書き方が可能なため、基礎を学ぶ上では適している。また、Visual Studioにおいてもなるべく簡単に実行するため、win32コンソールアプリケーションで動作確認を行う。

(1)ダウンロード及び解凍

教員から配布されるDVDにてVisual Studio 2015 をインストールしておく。

このとき、C++をインストールするようにチェックを入れることを忘れずに。

(将来のことも考えるとPython Toolsも入れておく方が良い)

続いて、以下からOpenCV_1.1pre1a.exeをダウンロードしてくる

https://sourceforge.net/projects/opencvlibrary/files/opencv-win/1.1pre1/OpenCV_1.1pre1a.exeを実行


Add <...>%OpenCV%bin to the system PATH の設定はそのまま

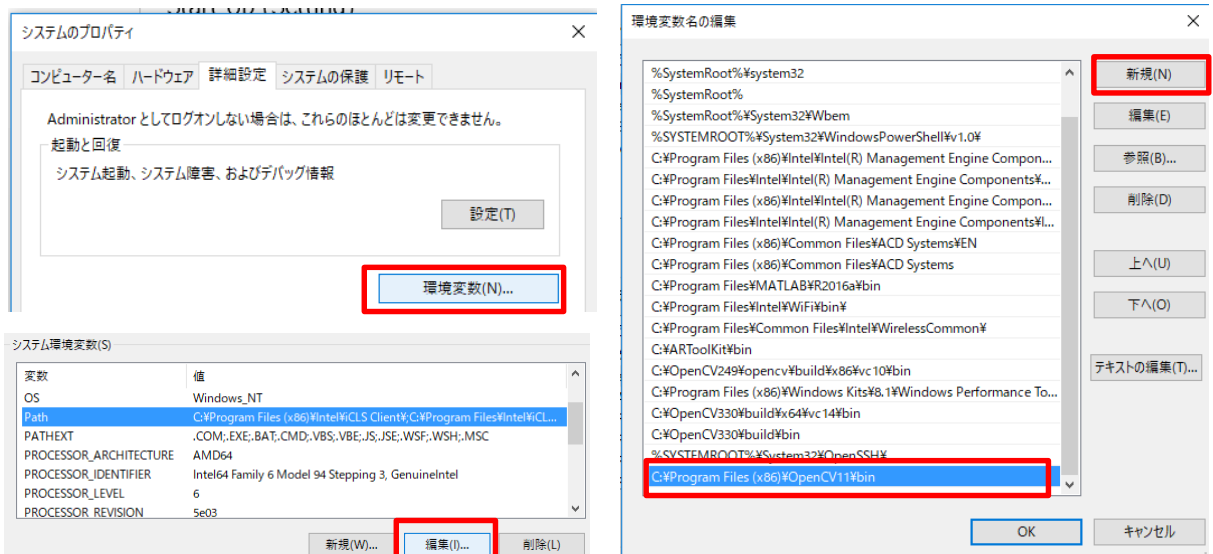
保存ディレクトリをC:%Program Files (x86)%OpenCV11にする



(2)PATHの設定

Pathは設定されているはず→一旦ログオフ+ログインで以下を確認

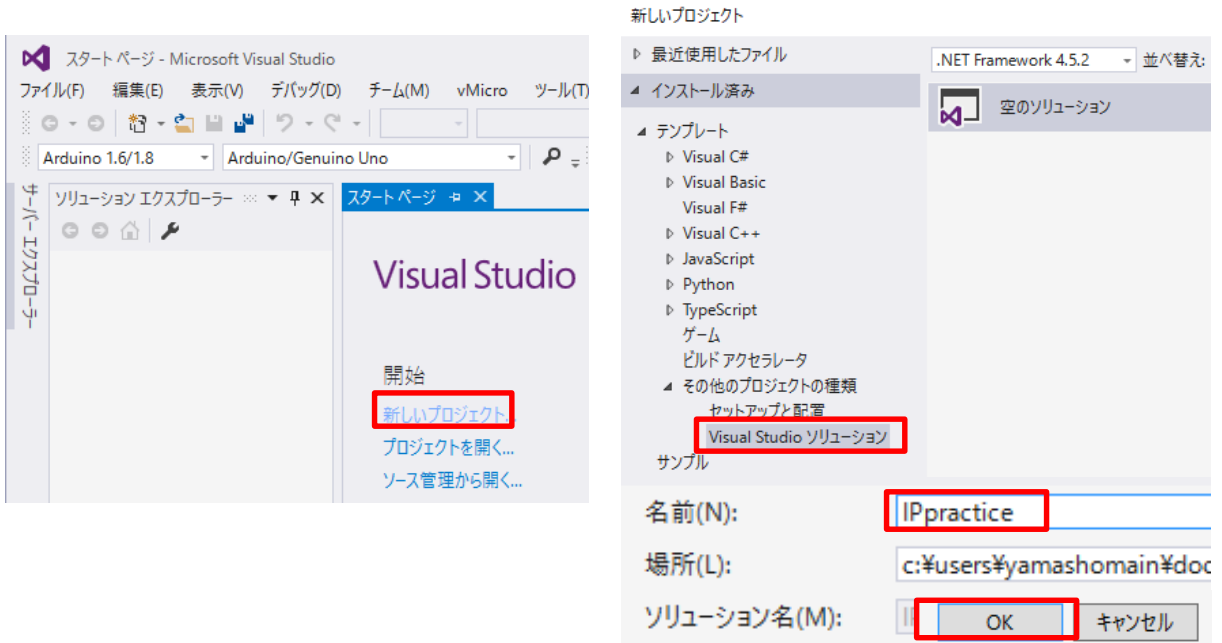
コントロールパネル¥システムとセキュリティ¥システム  システムの詳細設定



Start Up (Setting)

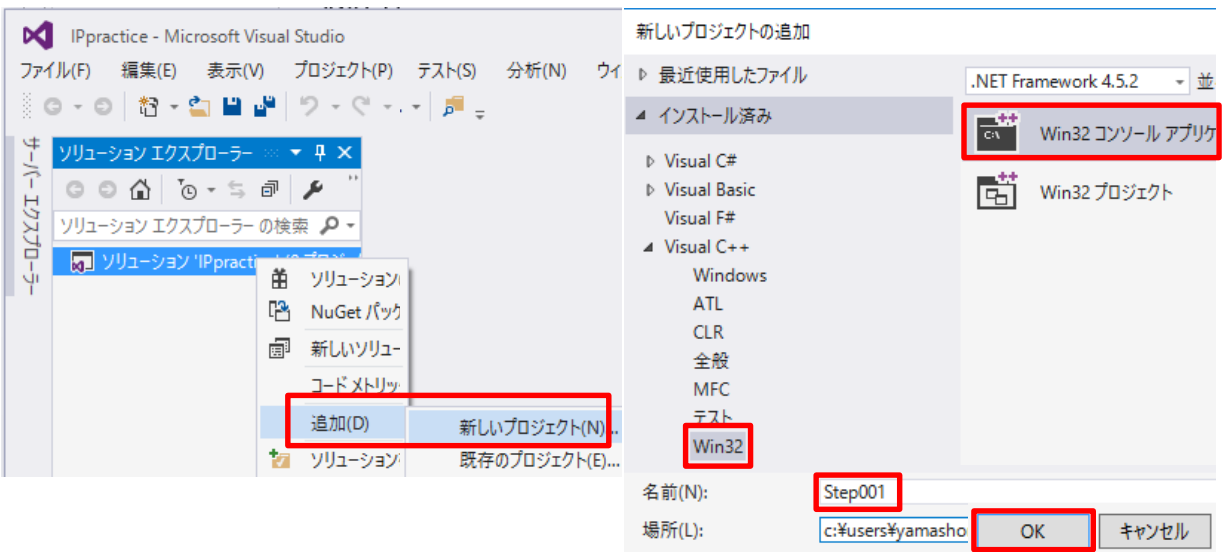
(3) VisualStudioのプロジェクト生成

VisualStudioを起動して、新しいプロジェクトを生成する。プログラムをまとめるために、今回はソリューションから作成する（通常はプロジェクトが良い）。



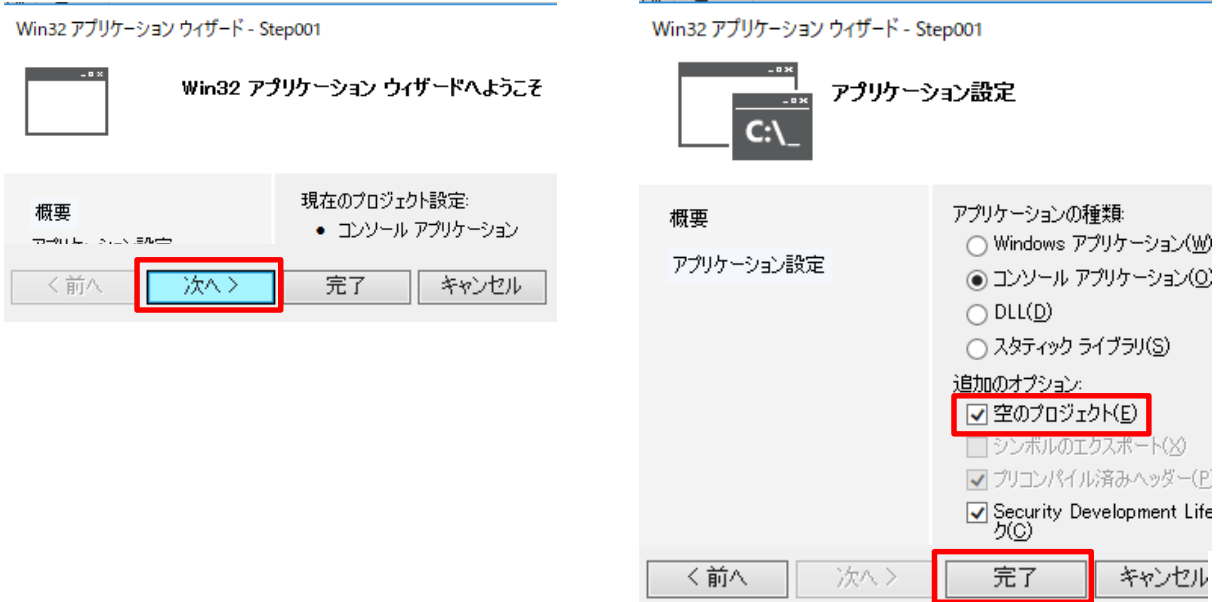
ソリューションを右クリックして追加、新しいプロジェクトを選択する。

作成するプロジェクトは Win32 のコンソールアプリケーションであり、ファイル名を記入してOKを押す。

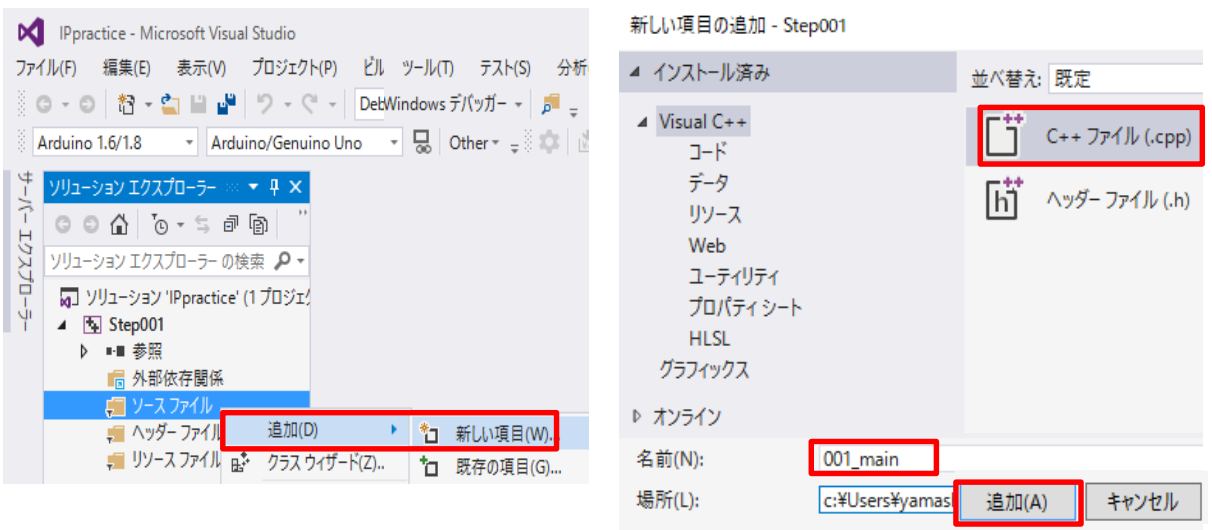


Start Up (Setting)

続いて、プロジェクトのウィザードが始まるので、まずは「次へ」で進み、設定で空のプロジェクトを選択して、完了させる。



最後に、作成したプロジェクト内のソースファイルを右クリックして、追加で新しい項目を選択する。

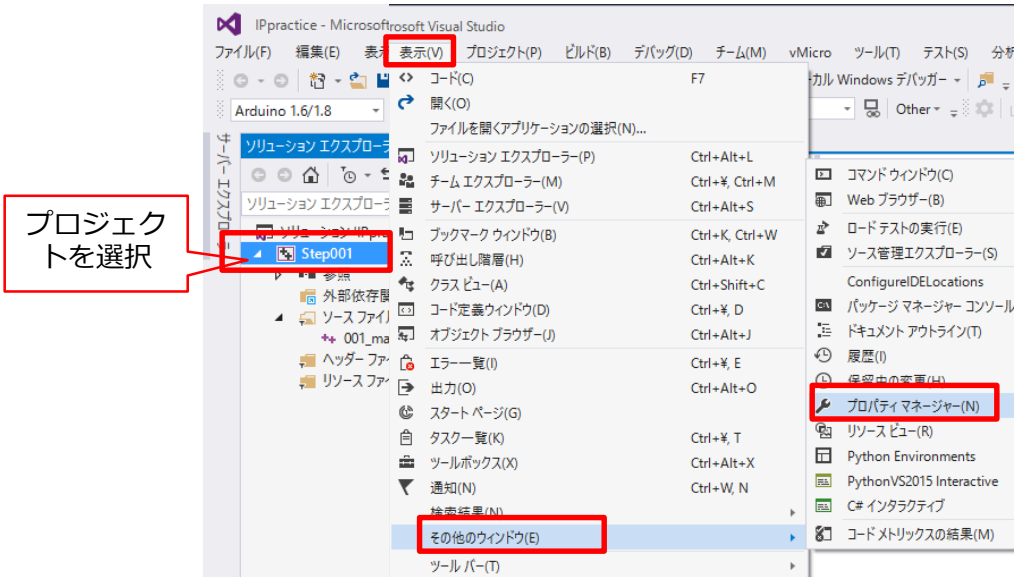


これでプログラムを作成する環境は整いました。次にはOpenCVを利用するためのプロパティを設定します。

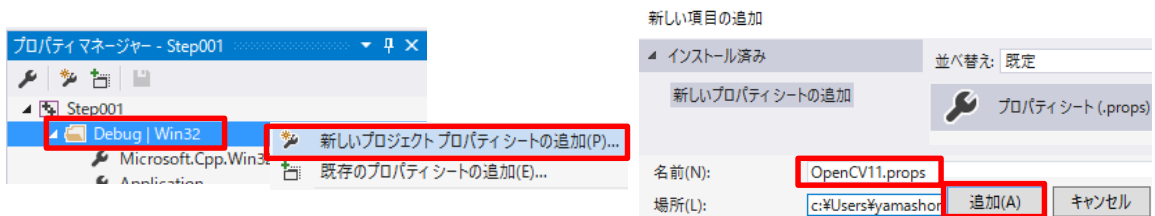
Start Up (Setting)

(4) プロパティの設定

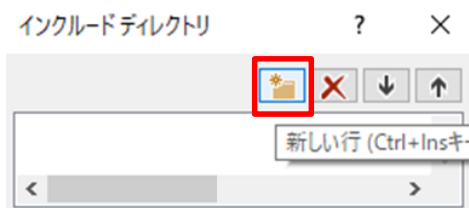
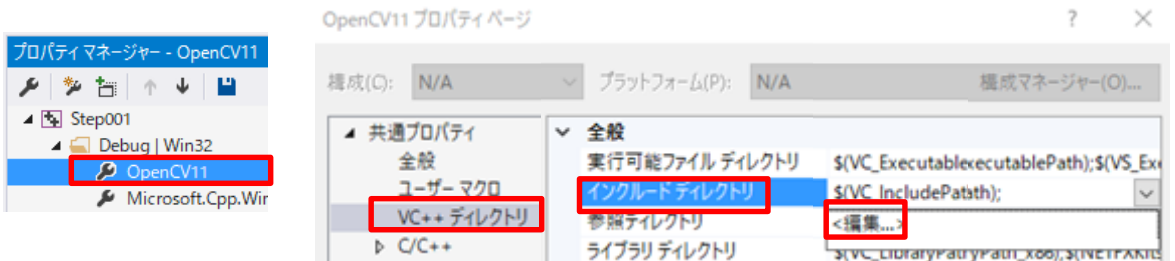
OpenCVは追加ライブラリであるため、使えるようにするための設定が必要。まずは表示→その他のウィンドウ→プロパティマネージャを起動する。



プロパティマネージャで選択したプロジェクトのDebug|Win32ホルダを右クリックして、新しいプロパティシートを作成する（名前を付けて追加）。



作成したプロパティシートをクリックして、VC++ ディレクトリを開く。その中のインクルードディレクトリ→編集を選択。

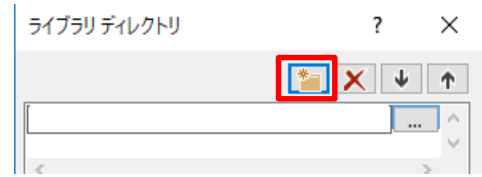
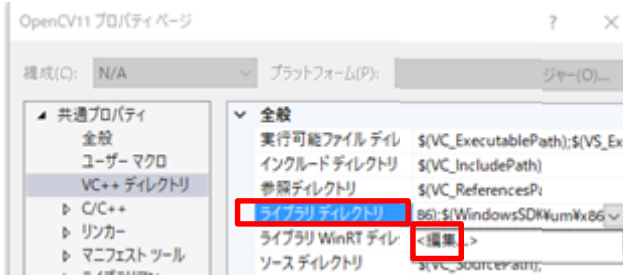


以下を登録

- C:\Program Files (x86)\OpenCV11\cv\include
- C:\Program Files (x86)\OpenCV11\cvaux\include
- C:\Program Files (x86)\OpenCV11\cxcore\include
- C:\Program Files (x86)\OpenCV11\otherlibs\highgui

Start Up (Setting)

同様に、ライブラリディレクトリ→編集を選択。



以下を登録
C:¥Program Files (x86)¥OpenCV11¥lib

プロパティシートはプロジェクトごとに設定しなければならない。
(面倒くさい)
→プロジェクトホルダに保存されているので、それを流用する。

(5)動作確認

```
#include <cv.h>
#include <highgui.h>
#include <stdio.h>
#include <stdlib.h>
```

← OpenCVのインクルード

```
#pragma comment(lib,"cv.lib") // for OpenCV
#pragma comment(lib,"cxcore.lib") // for OpenCV
#pragma comment(lib,"highgui.lib") // for OpenCV
```

← OpenCVのLib宣言

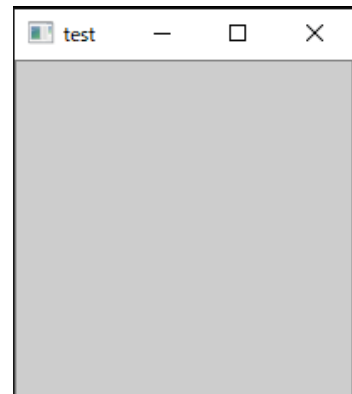
```
int main( int argc, char** argv ){

    //入力画像
    IplImage *test = cvCreateImage(cvSize(100, 100), IPL_DEPTH_8U, 3);
    // ウィンドウを生成する
    cvNamedWindow( "test", CV_WINDOW_AUTOSIZE );

    // 画像表示
    cvShowImage( "test", test);
    cvWaitKey( 0 ); //Key入力待ち

    // メモリを解放する
    cvReleaseImage( &test );

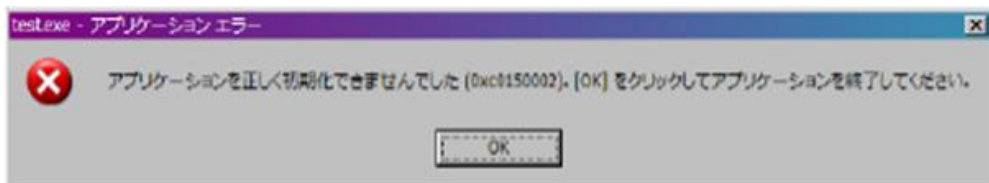
    // ウィンドウを破棄する
    cvDestroyWindow( "test" );
    return 0;
}
```



Start Up (Setting)

<エラー事例集>

- (1) プロジェクトをビルドしようとする時、"cxcore110.dll"といったエラーが表示される
PATHの設定がなされていない可能性が高い。再チェックを。
- (2) OpenCVをコンパイルしようとする時、以下のエラーが表示される。



C++ 2005 SP1再頒布可能パッケージが入っていない。以下からダウンロードしてインストールする。

<https://www.microsoft.com/ja-jp/download/details.aspx?id=26347>

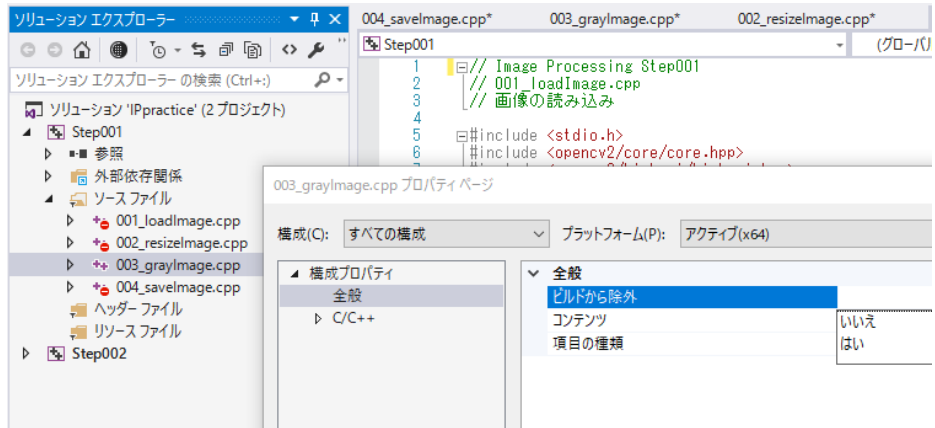
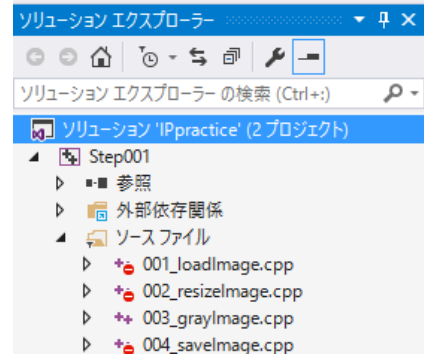
(Yamasho研のサーバーにも存在している, vcredist_x86.exe)

Start Up (Setting)

【備考】複数ファイルの管理及びビルド制御

複雑なプログラムの場合、複数のソースファイルが生成され、それぞれ、ビルドに含めるかどうかを選択しなければならない場合がある。

その場合、各ファイルのプロパティページからビルドの除外制御をすることができる。



また、複数のプロジェクトを選択する場合には、実行対象のプロジェクトを右クリックして、「スタートアッププロジェクトに設定」を選択すれば、そのプロジェクトだけコンパイル/実行できる。

