

3D マップ生成システム インストールマニュアル

発行日 2017 年 12 月 27 日

公立大学法人会津大学

株式会社東日本計算センター

目次

1. 本書について.....	1
1.1. インストール環境.....	1
2. 開発環境セットアップ手順.....	2
2.1. 統合開発環境インストール手順.....	2
2.1.1. Visual Studio Professional 2013 インストール.....	2
2.2. OpenRTM 開発環境インストール手順.....	6
2.2.1. OpenRTM_Cplus インストール.....	6
2.2.2. Python2.7 インストール.....	9
2.2.3. PyYAML インストール.....	12
2.2.4. Cmake インストール.....	14
2.2.5. Doxygen インストール.....	16
2.2.6. OpenRTM_Python インストール.....	19
2.3. 各種ライブラリのインストール手順.....	22
2.3.1. OpenCV インストール.....	22
2.3.2. NumPy インストール.....	29
2.3.3. python-dateutil インストール.....	30
2.3.4. FreeGLUT インストール.....	31
2.3.5. PyQt インストール.....	36
2.3.6. PCL インストール.....	38
3. 開発に際しての注意点.....	43
3.1. OpenCV を使用する際の設定.....	43
3.1.1. OpenCV 使用時のコンパイルオプション.....	43
3.2. PCL を使用する際の設定.....	46
3.2.1. PCL 使用時の Cmake ファイル.....	46
3.3. Windows 標準の PathFileExists を使用する際の設定.....	47
3.3.1. PathFileExists 使用時のコンパイルオプション.....	47
4. 参考.....	48

1. 本書について

本書は 3D マップ生成システムの開発環境を構築するための手順を記載します。OS やネットワークの基本設定等は実施済みであること、また OS やネットワークの知識を有することを前提とし記載していますので、詳しい説明等は別途マニュアルなどを参照してください。

1.1. インストール環境

本書では以下の環境を想定しています。

表 1-1 動作環境

環境		バージョン	補足
CPU	Intel Xeon E5-2620v3 (2.40GHz, 6 コア, 15MB, 1866MHz)	-	-
メモリ	64.0GB	-	8GB 以上であれば動作可能
HDD	920GB	-	2GB 以上の空き容量があればインストール可能
ディスプレイ解像度	1920×1080	-	1366×768 以上が推奨
OS	Windows	8.1	-
RT ミドルウェア	OpenRTM-aist C++ OpenRTM-aist Python	1.1.1 1.1.0	-
依存ライブラリ	OpenCV	3.0	画像制御に使用
	NumPy	1.13.3	画像行列計算に使用
	python-dateutil	2.6.1	datetime モジュールの拡張機能として使用
	Freeglut	2.8.1	画像描画に使用
	PyQt	4.11.4	GUI 制御に使用
	PCL(Point Cloud Library)	1.8.0	Affine 変換に使用

2. 開発環境セットアップ手順

3D マップ生成システム開発に必要な IDE（統合開発環境）及び各種ライブラリのインストール手順を記載します。

なお、インストールのフェーズとして下記の通り大きく 3 段階あり、フェーズ毎に章を分けて記載します。

- 統合開発環境インストール
- OpenRTM 開発環境インストール
- 各種ライブラリインストール

また、インストールの順番は依存性があり、手順書の記載と異なる順番でインストールした場合、正常にインストールできない可能性があるため注意が必要です。

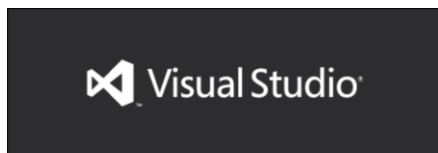
2.1. 統合開発環境インストール手順

2.1.1. Visual Studio Professional 2013 インストール

簡易的なインストール手順を記載しますが、詳しいインストール手順はお手元のマニュアルを参照してください

① インストールファイル実行

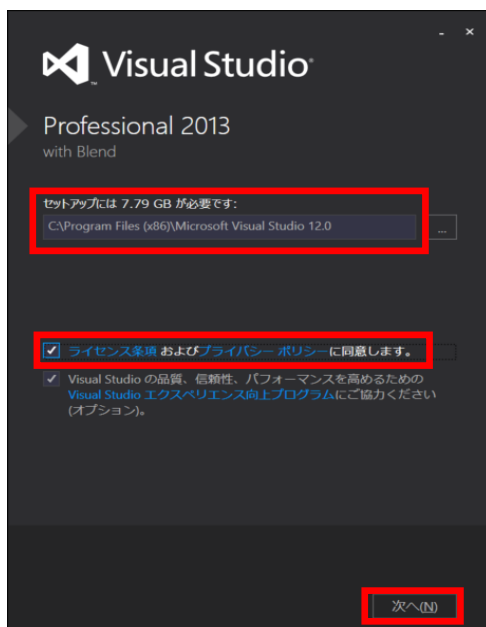
スプラッシュスクリーンが閉じるまで待機します。



② インストール先及びライセンス条項の同意

インストール先はデフォルトパスを指定し、「ライセンス条項およびプライバシーポリシーに同意します。」にチェックを入れます。

次へボタン押下します。



③ インストールするオプション機能の選択

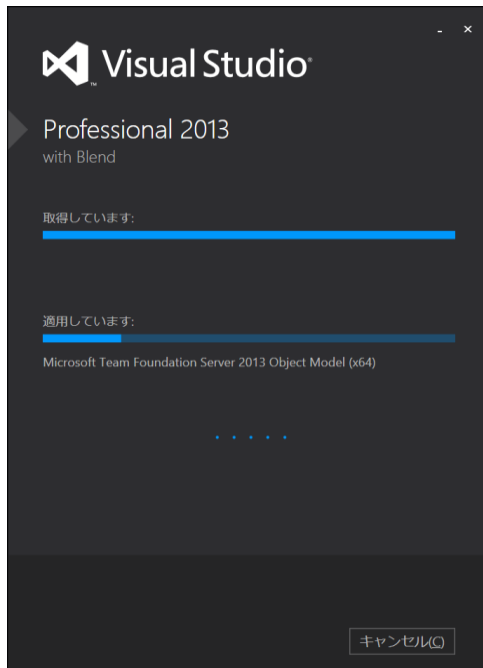
デフォルトで良いですが、必要に応じてオプションを選択します。

インストールボタン押下します。



④ インストール実行中

インストール完了まで待機します。

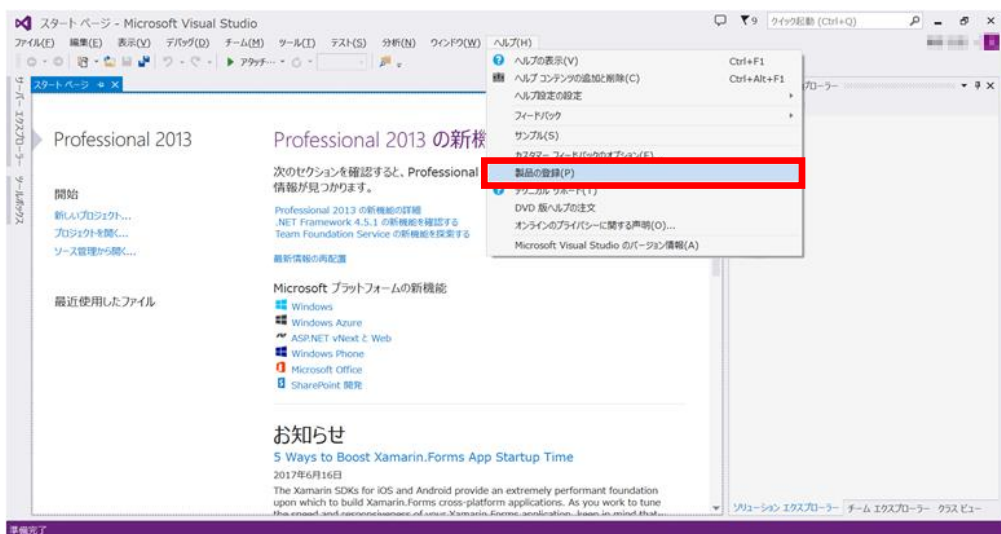


⑤ マイクロソフトアカウントへログイン

インストール完了後、起動し任意のマイクロソフトアカウントにログインを行います。

⑥ プロダクトキーの設定

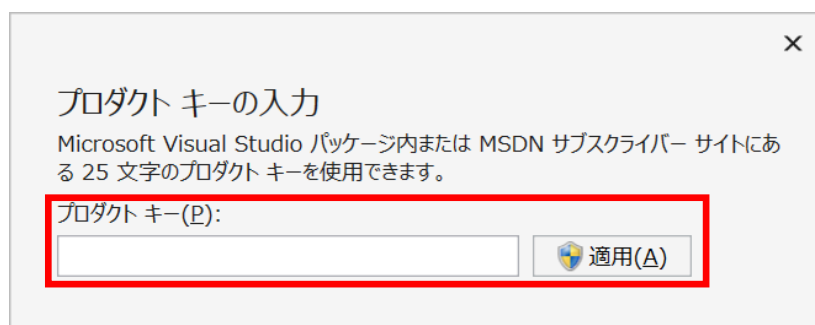
[ヘルプ]-[製品の登録]メニューをクリックします。



「製品ライセンスを変更します」をクリックします。



下のテキストボックスにプロダクトキーを入力し、適用ボタンを押下します。



⑦ インストール完了

2.2. OpenRTM 開発環境インストール手順

2.2.1. OpenRTM_Cplus インストール

簡易的なインストール手順を記載しますが、詳しいインストール手順は公式サイトを参照してください。(http://openrtm.org/openrtm/ja/content/lets_start)

① インストールファイルの入手

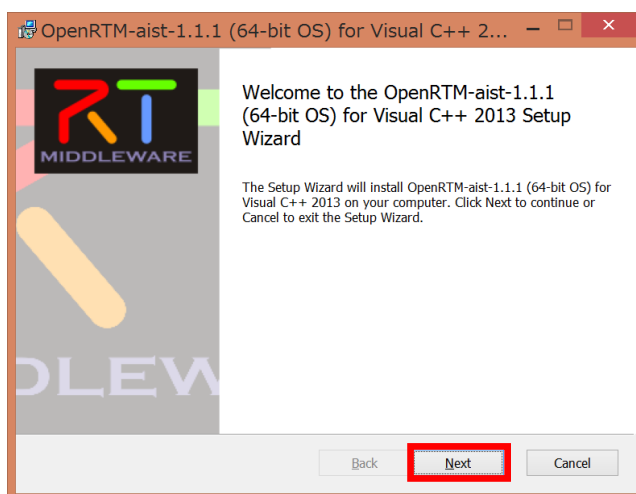
インストールファイルを公式サイトより入手します。

http://openrtm.org/pub/Windows/OpenRTM-aist/cxx/1.1/OpenRTM-aist-1.1.1-RELEASE_x86_64_vc12.msi

② インストールファイルの実行

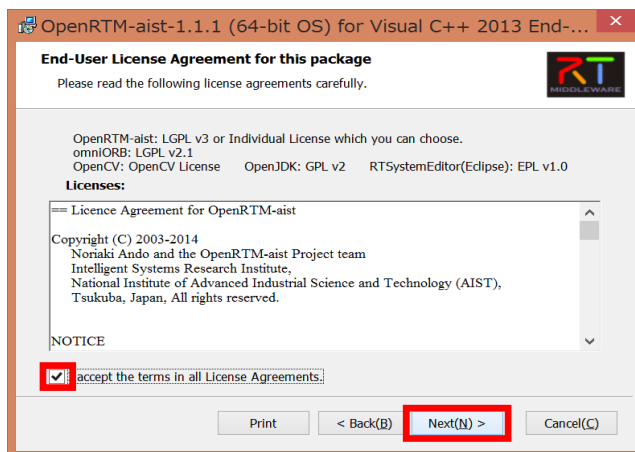
「OpenRTM-aist-1.1.1-RELEASE_x86_64_vc12.msi」を実行します。

Next ボタン押下します。



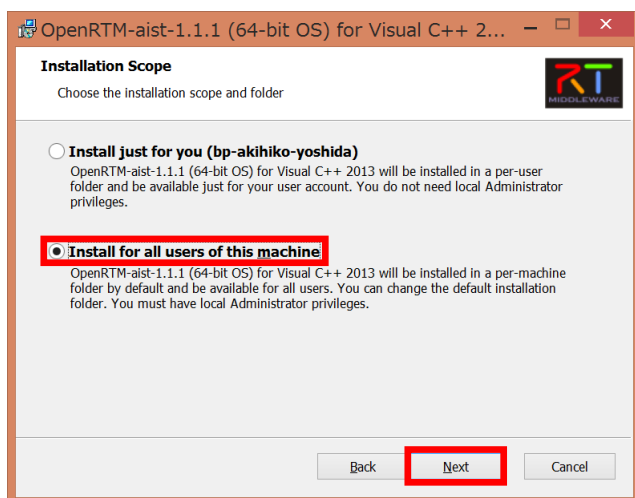
③ ライセンス条項の同意

「I accept the terms in all License Agreements.」にチェックを入れます。
Next ボタン押下します。



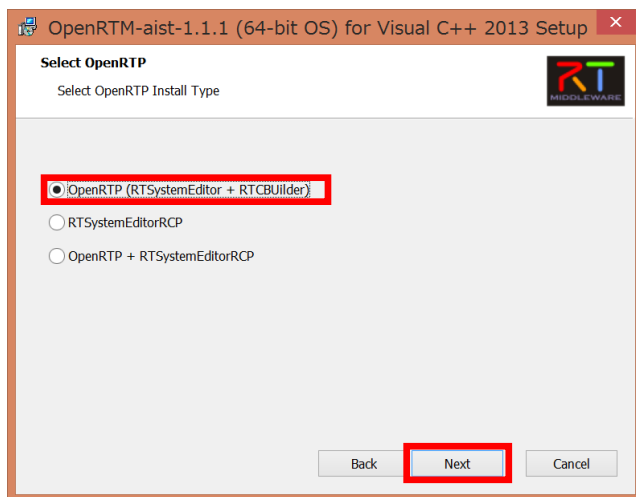
④ インストールの種類

「Install for all users of this machine」のラジオボタンを選択します。
Next ボタン押下します。



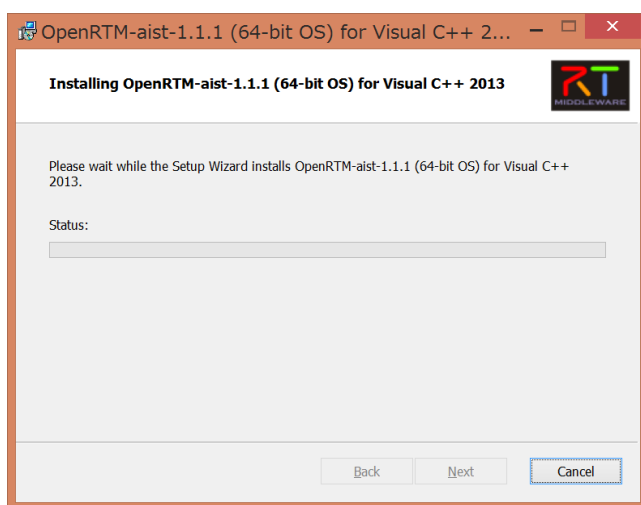
⑤ インストールオプション選択

「OpenRTP(RTSystemEditor + RTCBuilder)」のラジオボタンを選択します。
Next ボタンを押下します。



⑥ インストール実行

インストール完了まで待機します。



※環境によっては OpenJDK のインストールを求められる場合があります。
ウィザードに沿ってデフォルトのままインストールしてください。

⑦ インストール完了

2.2.2. Python2.7 インストール

① インストールファイルの入手

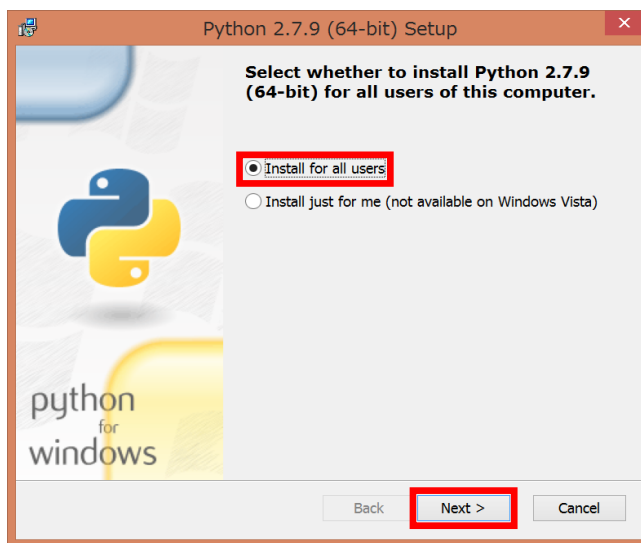
インストールファイルを公式サイトより入手します。

<https://www.python.org/ftp/python/2.7.9/python-2.7.9.amd64.msi>

② インストールファイルの実行

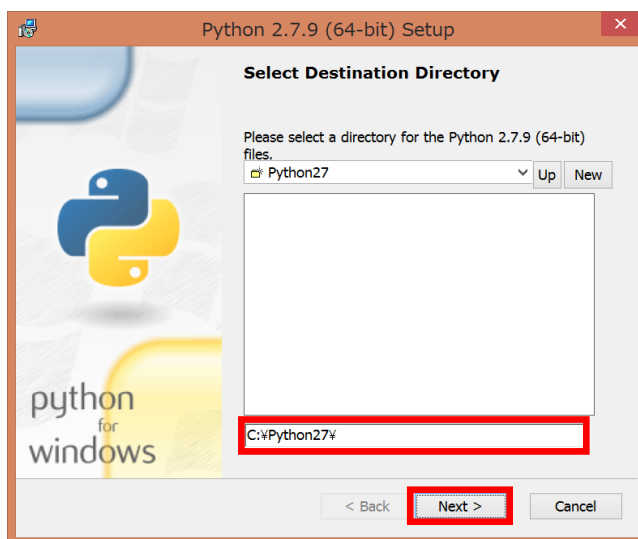
「python-2.7.9.amd64.msi」を実行します。

「Install for all users」ラジオボタンを選択し、Next ボタンを押下します。



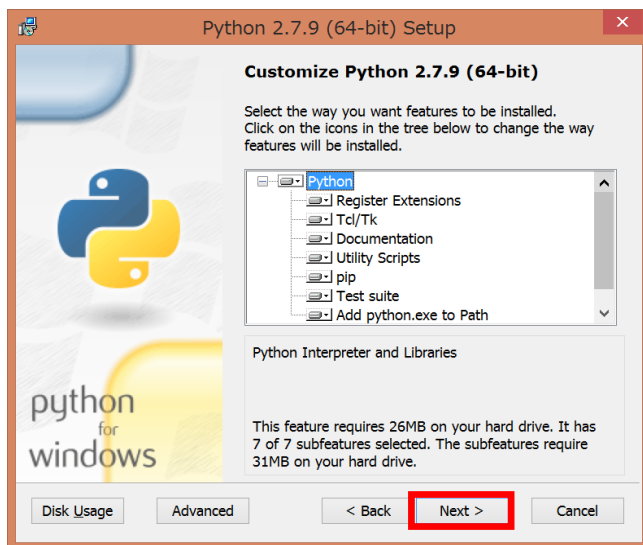
③ インストール先の指定

「C:\Python27\」と指定し、Next ボタンを押下します。



④ インストールするオプション選択。

デフォルトのまま Next ボタンを押下します。



⑤ インストール実行

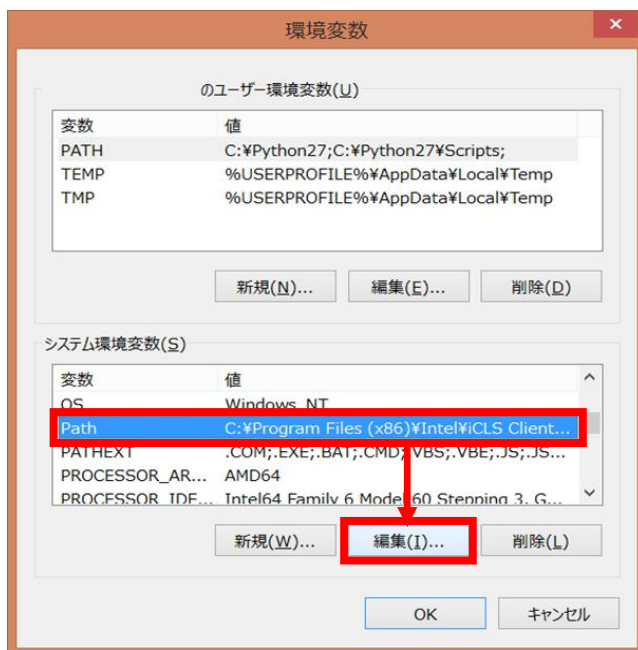
インストールが完了するまで待機します。

⑥ 環境変数の設定

システム環境変数内にある”Path”に次の値を追加します。

C:¥Python27

C:¥Python27¥Scripts



⑦ コンピュータの再起動

環境変数を適用するため、OS をリブートします。

⑧ pip のバージョンアップ

pip のバージョンが古いため、バージョンアップを行います。

コマンドプロンプトより下記のコマンドを実行してください。

```
> pip install -U pip
```

⑨ インストール完了

2.2.3. PyYAML インストール

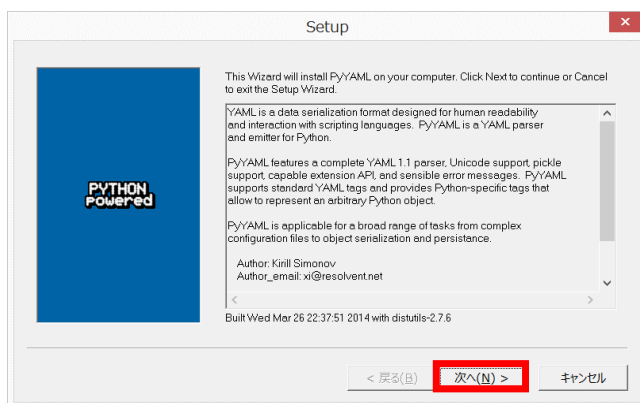
① インストールファイルの入手

インストールファイルを公式サイトより入手します。

<http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py2.7.exe>

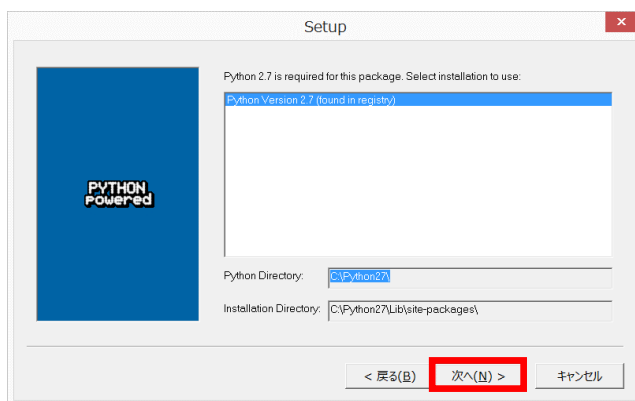
② インストールファイルの実行

「PyYAML-3.11.win-amd64-py2.7.exe」を実行し、次へボタンを押下します。



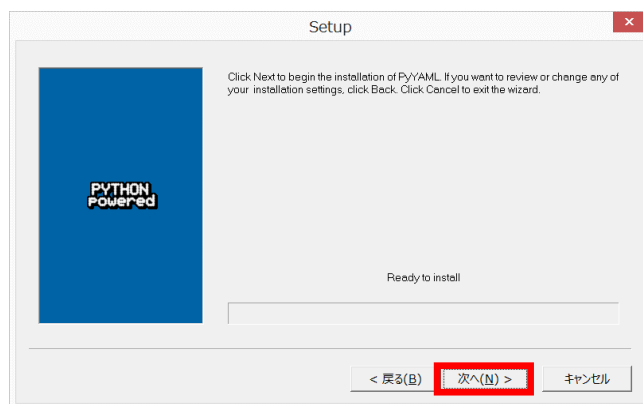
③ インストール先の指定

デフォルトのまま次へボタンを押下します。



④ インストール実行

次へボタンを押下し、インストール完了まで待機します。



⑤ インストール完了

2.2.4. Cmake インストール

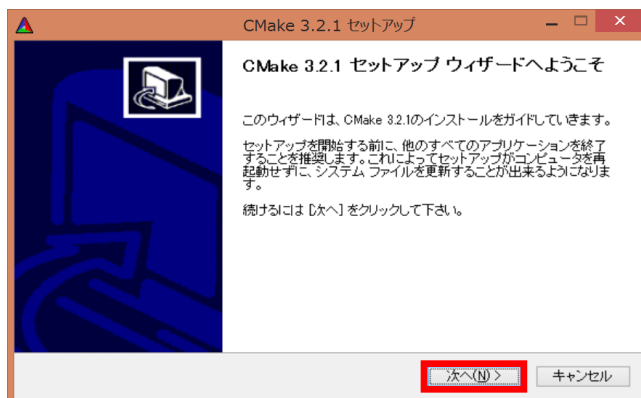
① インストールファイルの入手

インストールファイルを公式サイトより入手します。

<http://www.cmake.org/files/v3.2/cmake-3.2.1-win32-x86.exe>

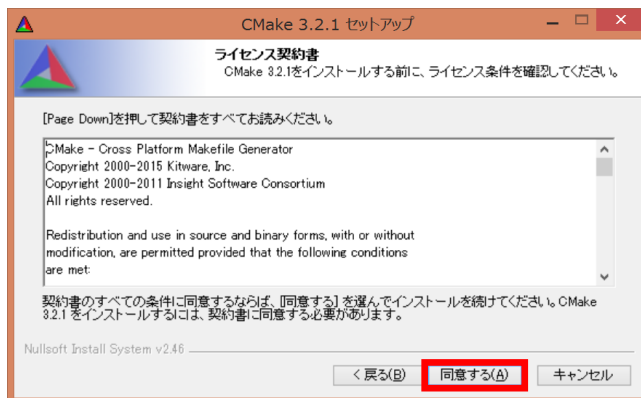
② インストールファイルの実行

「cmake-3.2.1-win32-x86.exe」を実行し、次へボタンを押下します。



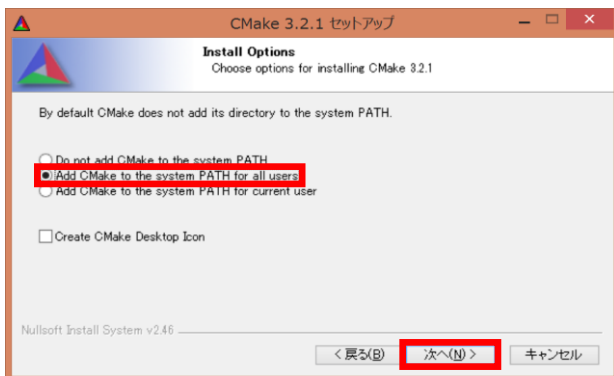
③ ライセンス条項の同意

「同意する」ボタンを押下します。



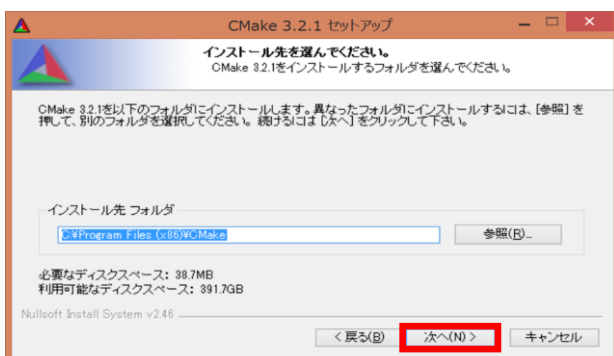
④ インストールオプション選択

「Add Cmake to the system PATH for all users」ラジオボタンを選択し、次へボタンを押下します。



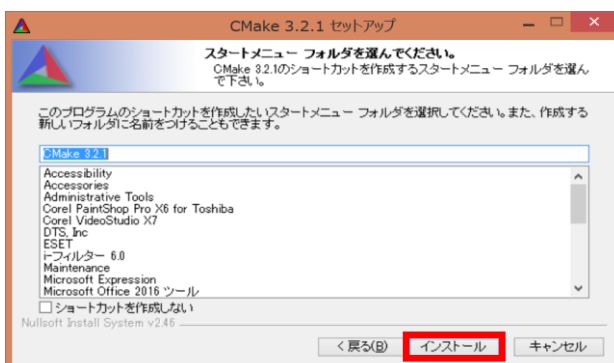
⑤ インストール先の選択

デフォルトのまま次へボタンを押下します。



⑥ スタートメニューフォルダ選択

デフォルトのまま「インストール」ボタンを押下します。



⑦ インストール完了

2.2.5. Doxygen インストール

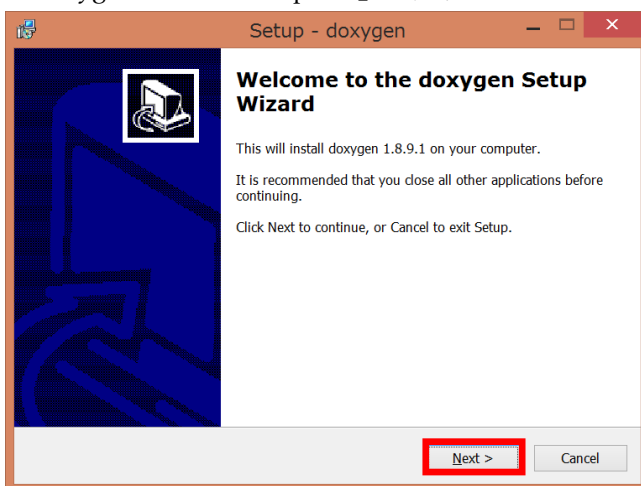
① インストールファイルの入手

インストールファイルを公式サイトより入手します。

<http://ftp.stack.nl/pub/users/dimitri/doxygen-1.8.9.1-setup.exe>

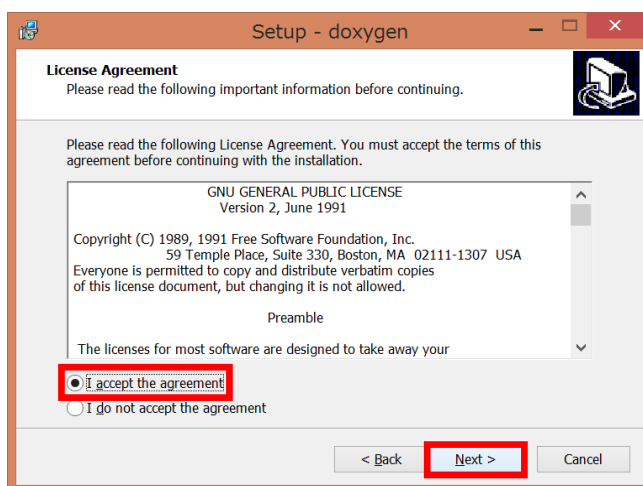
② インストールファイルの実行

「doxygen-1.8.9.1-setup.exe」を実行し、Next ボタンを押下します。



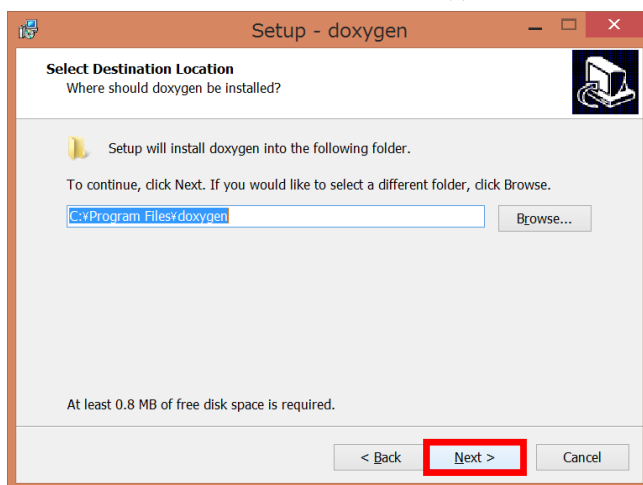
③ ライセンス条項の同意

「I accept the agreement」ラジオボタンを選択し、Next ボタンを押下します。



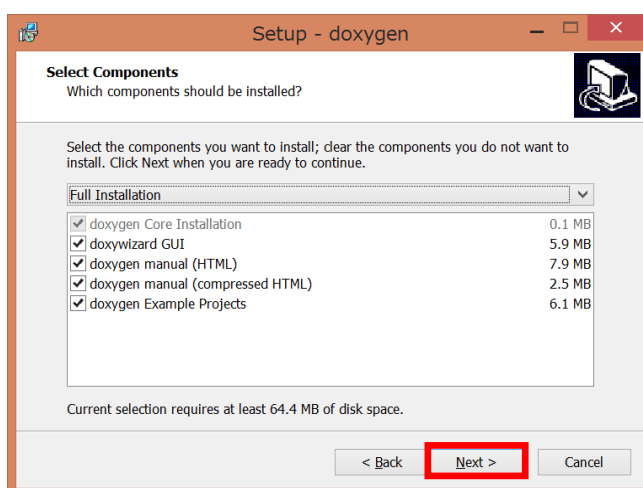
④ インストール先の選択

デフォルトのまま Next ボタンを押下します。



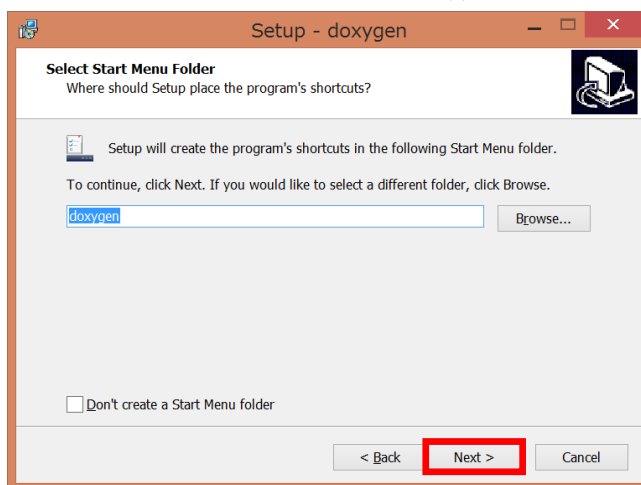
⑤ インストールオプション選択

デフォルトのまま Next ボタンを押下します。



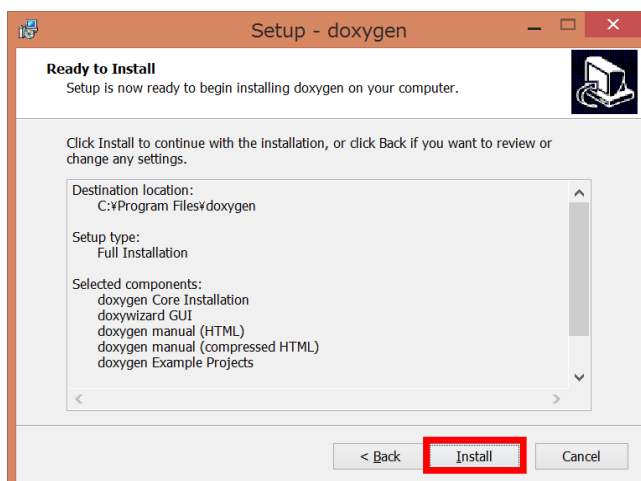
⑥ スタートメニューフォルダ選択

デフォルトのまま Next ボタンを押下します。



⑦ インストール実行

「Install」 ボタンを押下し、インストール完了まで待機します。



⑧ インストール完了

2.2.6. OpenRTM_Python インストール

簡易的なインストール手順を記載しますが、詳しいインストール手順は公式サイトを参照してください。(<http://openrtm.org/openrtm/ja/content/openrtm-aist-python-110-release>)

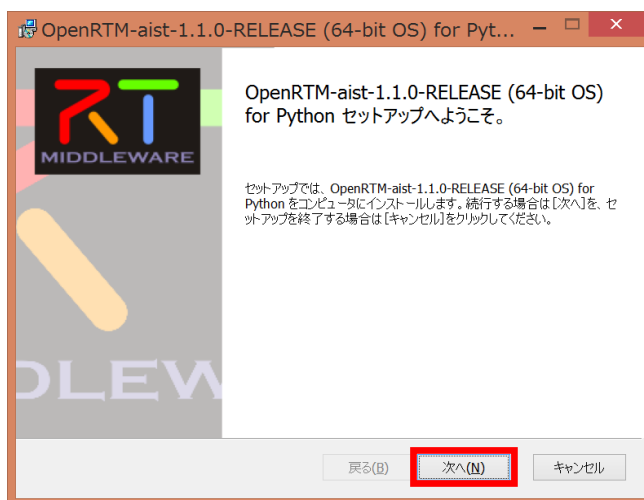
① インストールファイルの入手

インストールファイルを公式サイトより入手します。

http://openrtm.org/pub/Windows/OpenRTM-aist/python/OpenRTM-aist-Python_1.1.0-RELEASE_x86_64.msi

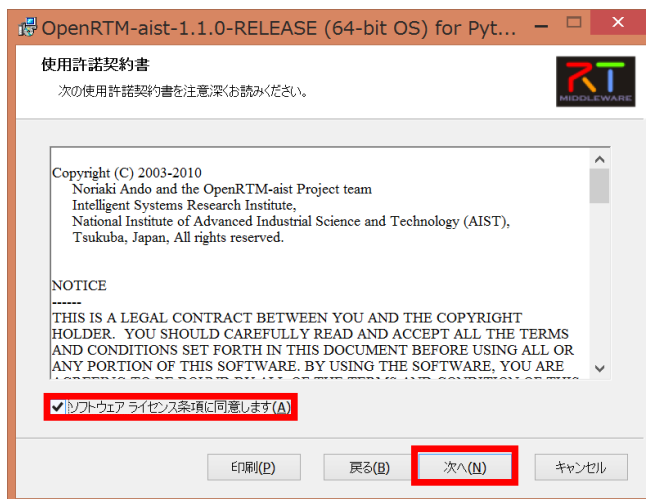
② インストールファイルの実行

「OpenRTM-aist-Python_1.1.0-RELEASE_x86_64.msi」を実行し、次へボタンを押下します。



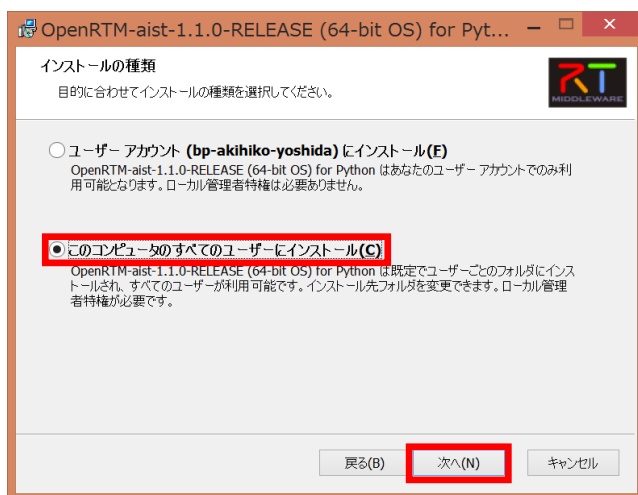
③ ライセンス条項の同意

「ソフトウェアライセンス条項に同意します」ラジオボタンを選択し、次へボタンを押下します。



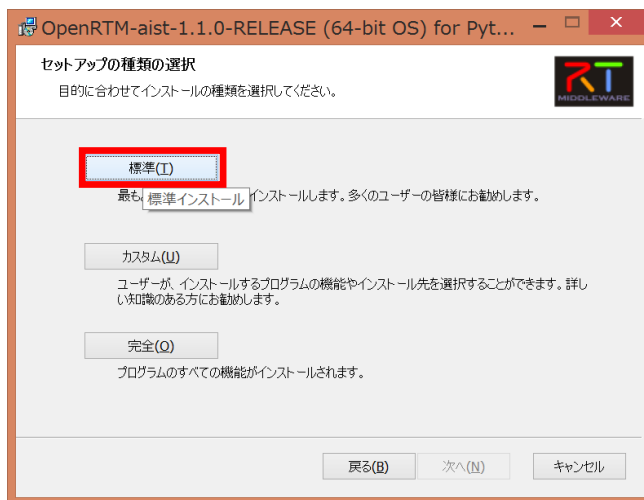
④ インストールの種類

「このコンピュータのすべてのユーザーにインストール」ラジオボタンを選択し、次へボタンを押下します。



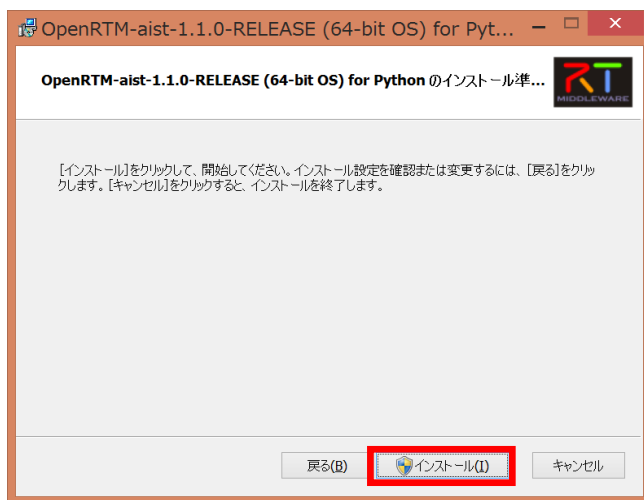
⑤ セットアップの種類を選択

「標準」ボタンを押下します。



⑥ インストール実行

「インストール」ボタンを押下し、インストール完了まで待機します。



⑦ インストール完了

2.3. 各種ライブラリのインストール手順

2.3.1. OpenCV インストール

OpenCV のインストールはほかのインストーラによる導入とは異なり、ソースファイルよりコンパイルする必要があります。大きな流れは以下の通り。

- ✓ ソースファイルのダウンロード
- ✓ ファイルの解凍
- ✓ Cmake にてソリューション作成
- ✓ Visual Studio 2013 にて OpenCV のコンパイル
- ✓ 作成された Lib 及び Bin フォルダのコピー
- ✓ 環境変数の設定

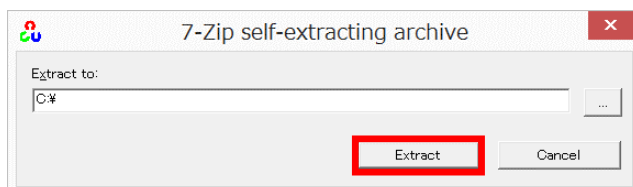
① ソースファイルの入手

ソースファイルを公式サイトより入手します。

<https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.0.0/opencv-3.0.0.exe/download>

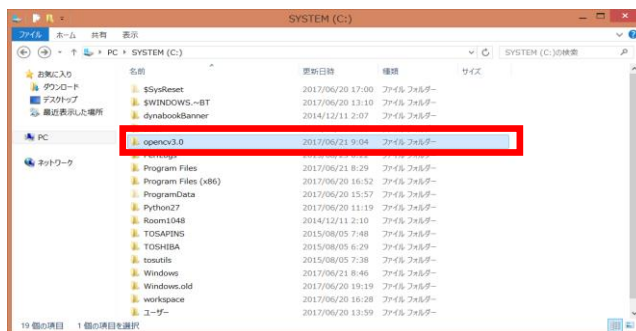
② ソースファイルの解凍

自己解凍型実行ファイルである「opencv-3.0.0.exe」を実行し、任意の解凍先ディレクトリを指定し、Extract ボタンを押下する。



③ 解凍したフォルダの移動

C ドライブ直下に「opencv3.0」フォルダを作成し、当該フォルダへ②で解凍した「opencv」フォルダを移動します。



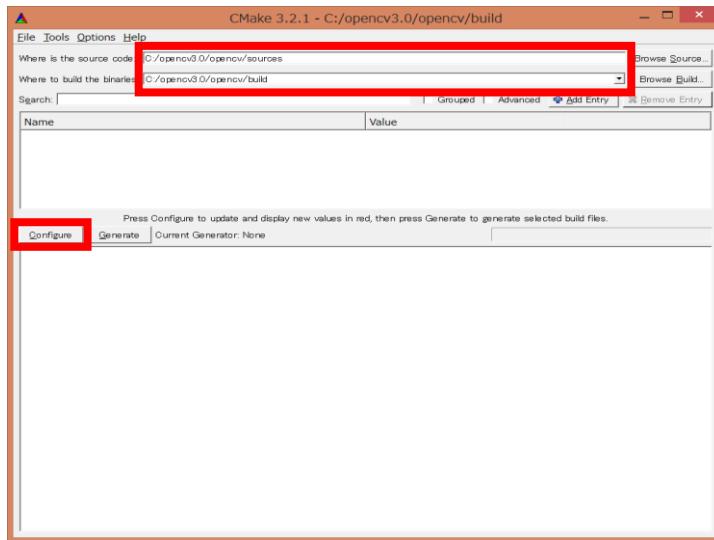
④ Cmakeにてコンパイルの準備（ソリューション作成）

Cmake を実行しソリューションを作成します。

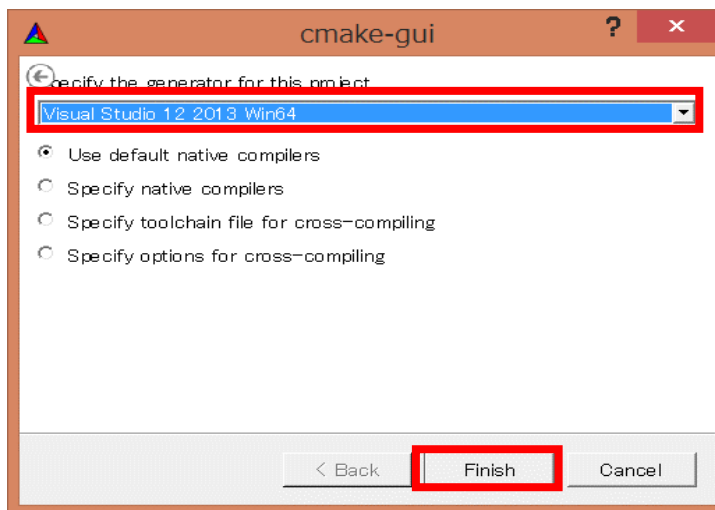
ソース格納場所に"C:\opencv3.0\opencv\sources"を指定し、

ビルド先に"C:\opencv3.0\opencv\build"を指定します。

各パスを指定後、Configure ボタンを押下します。

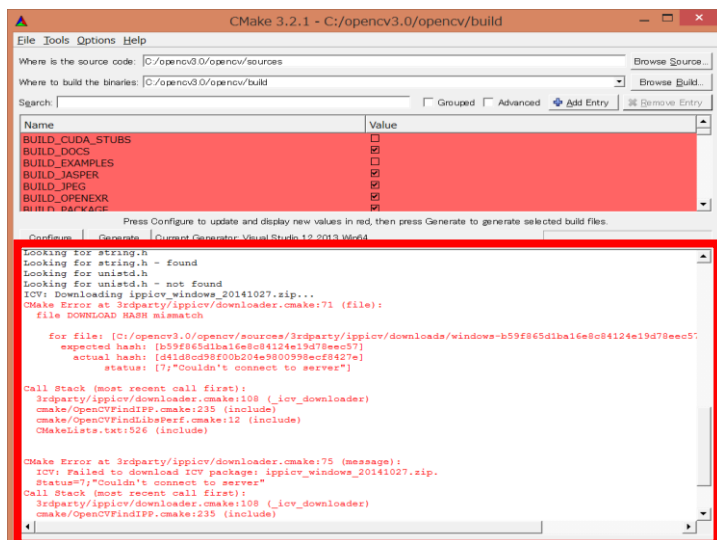


「Visual Studio 12 2013 Win64」を選択し、Finish ボタン押下します。



⚠ 以下のエラーが発生する場合があります

IPPICV ファイルをファイルサーバより取得する際に、バージョン（ハッシュ）エラーが発生しファイルが取得できていないことがあります。



この場合は、手動にてファイルを取得しファイル置き換える必要があります。

下記アドレスへアクセスし、"ippicv_windows_20141027.zip"ファイルをローカルへ保存します。

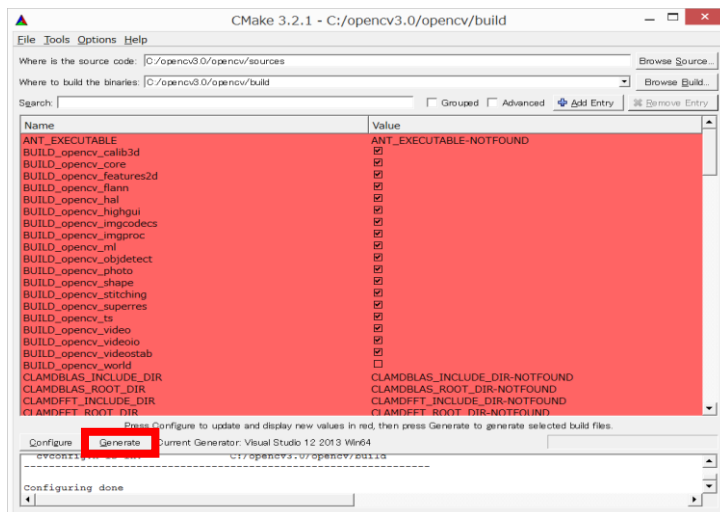
<https://sourceforge.net/projects/opencvlibrary/files/3rdparty/ippicv/>

ローカルに保存した"ippicv_windows_20141027.zip"ファイルを下記パスに格納されているファイルへ上書きします（解凍する必要はありません）。

C:\¥opencv3.0¥opencv¥sources¥3rdparty¥ippicv¥downloads¥windows-b59f865d1ba16e8c84124e19d78eec57

Cmakeにて再度「Configure」ボタンを押下します。

Configure 完了後、Generate ボタンを押下する

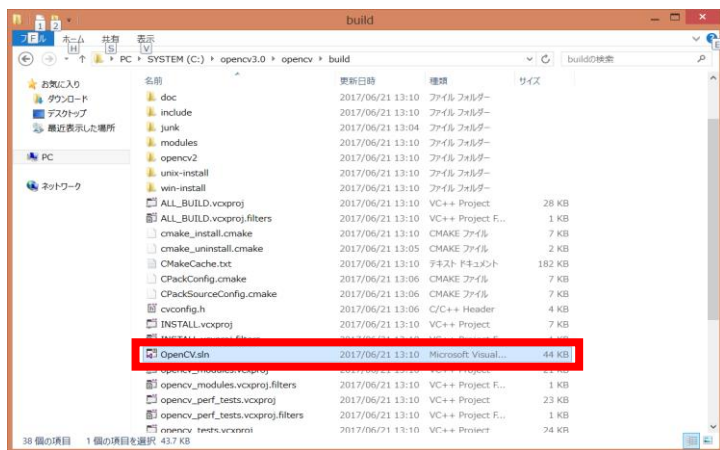


⑤ コンパイル実行

Visual Studio 2013 にてコンパイルを行います。

④で作成された下記パスのソリューションファイルを開きます。

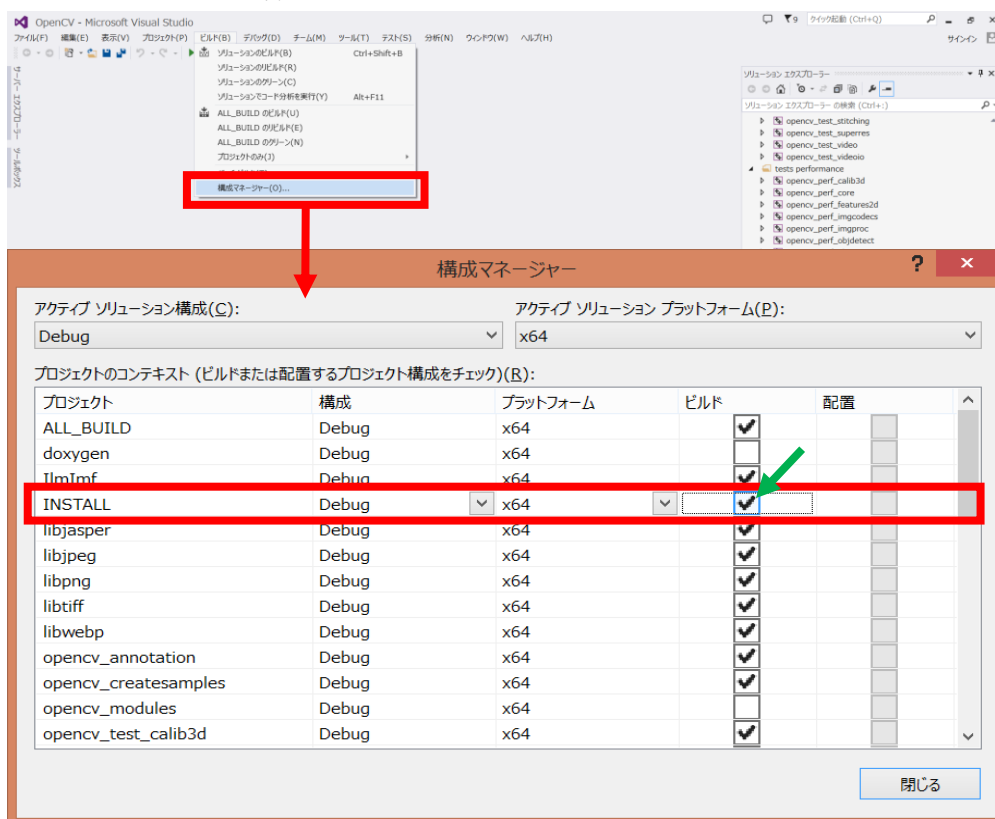
C:\¥opencv3.0¥opencv¥build¥OpenCV.sln



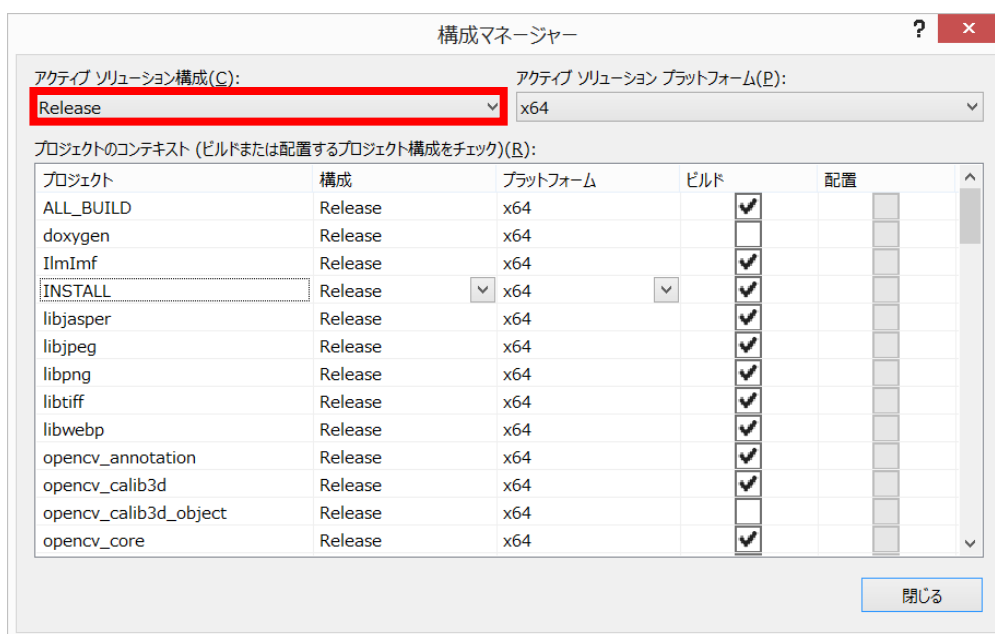
※インクルード解析に時間がかかります。

解析が終了するまでしばらくお待ちください。

ビルドメニューの構成マネージャーを押下し、"INSTALL"のビルドのチェックボックスにチェックを付けます。

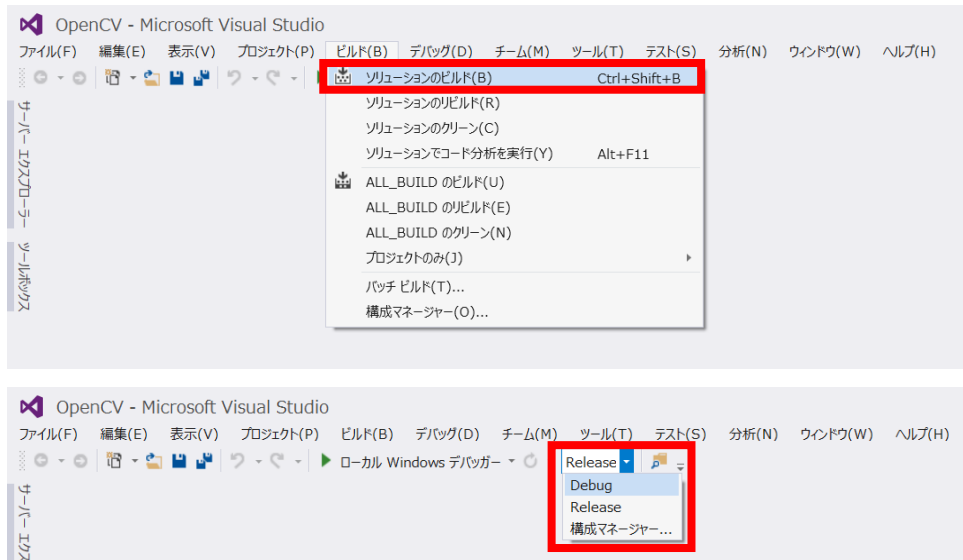


この設定をアクティブソリューション構成の Debug と Release の両方に行います。



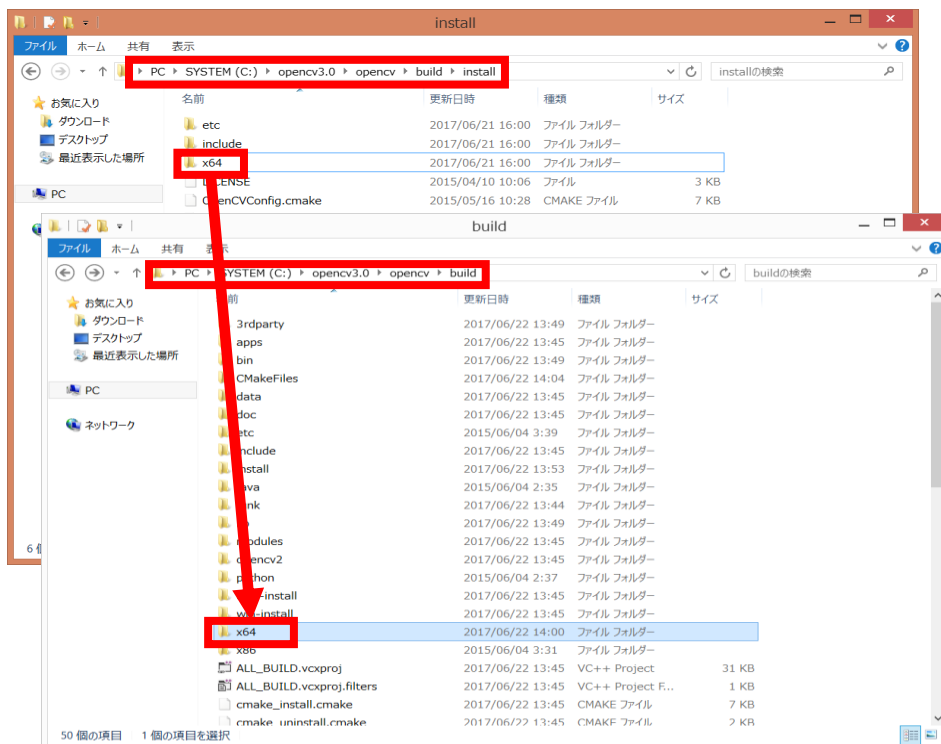
閉じるボタン押下し、ダイアログボックスを閉じます。

[ビルド]-[ソリューションのビルド]よりビルドを行います。
 なお、ビルドは Release と Debug の両方行います。



⑥ 作成された Lib 及び Bin のコピー

"C:\opencv3.0\opencv\build\install" フォルダに作成された"x64"フォルダをコピーし、一つ上の階層の"C:\opencv3.0\opencv\build"へ貼り付けます。なお、既存の"x64"フォルダが存在しますが、そのまま書きし統合します。



⑦ Python モジュールのコピー

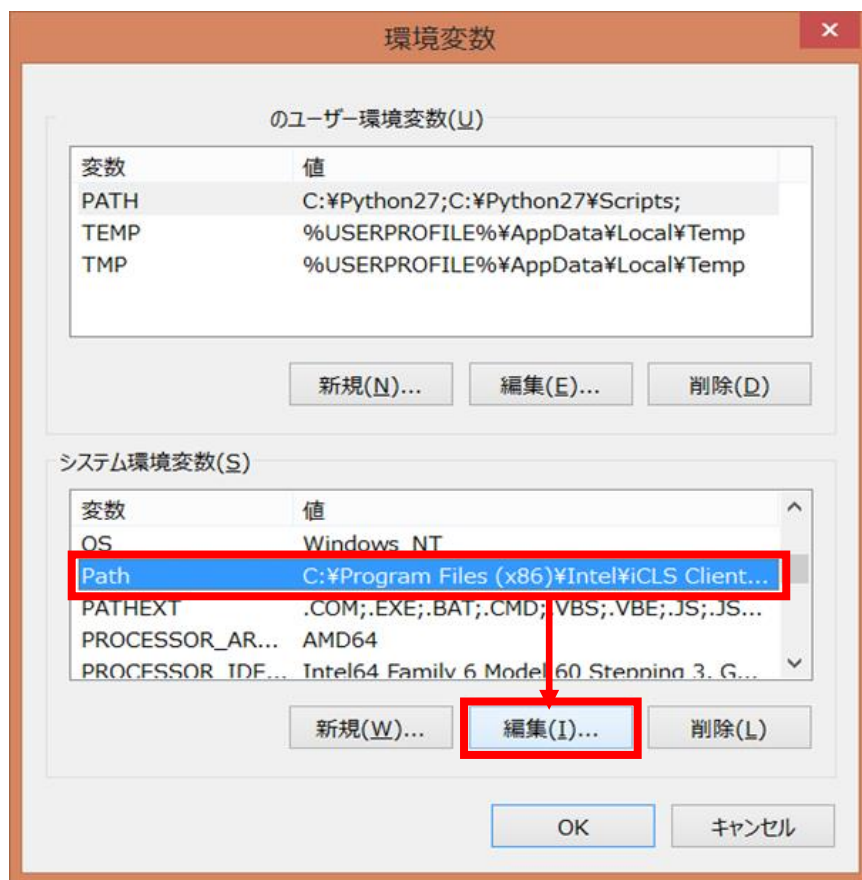
" C:\opencv3.0\opencv\build\python\2.7\x64" フォルダにある "cv2.pyd" ファイルを "C:\Python27\Lib\site-packages" へ貼り付けます。

⑧ 環境変数の設定

システム環境変数内にある "path" に次の変更を行います。

【削除】 C:\Program Files\OpenRTM-aist\1.1\OpenCV2.4.9\x64\vc12\bin

【追加】 C:\opencv3.0\opencv\build\x64\vc12\bin



⑨ コンピュータの再起動

環境変数を適用するため、OS をリブートします。

⑩ インストール完了

2.3.2. NumPy インストール

① インストールファイルの入手

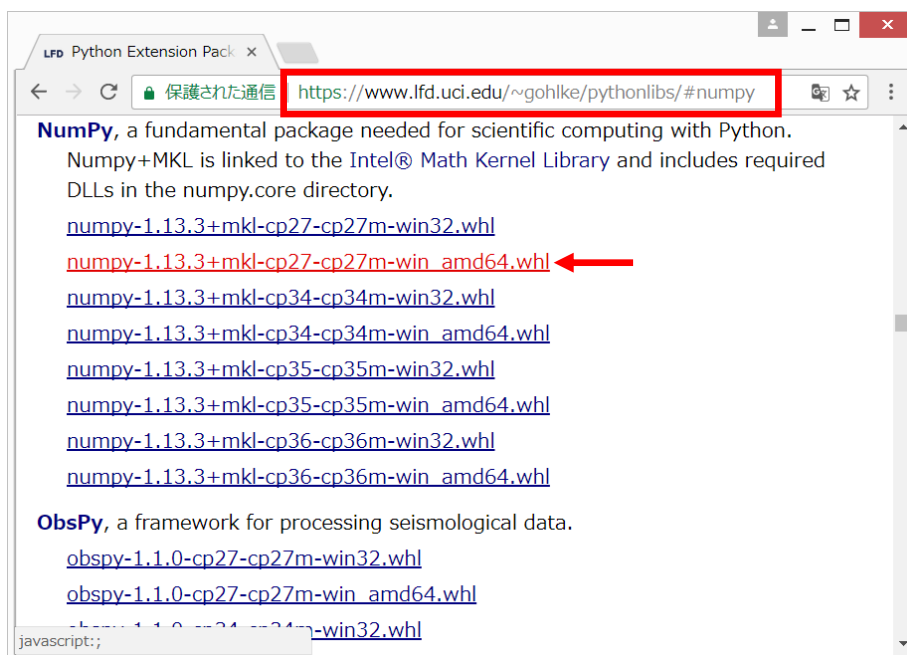
インストールファイルを LFD の Pythonlibs より最新ファイル入手します。
インストールファイルは任意の場所に保存してください。

<https://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>

対象ファイル名：numpy-[X.XX.X]+mkl-cp27-cp27m-win_amd64.whl

※[X.XX.X]はバージョンに対応しており、例とは異なる場合があります。

なお、本書では現時点の最新バージョン 1.13.3 を用いて説明します。



② インストール実行

pip を使ってインストールを行います。

コマンドプロンプトより下記のコマンドを実行してください。

```
> cd [①で取得した whl ファイルのディレクトリ]
```

```
> pip install mumpy-1.13.3+mkl-cp27-cp27m-win_amd64.whl
```

※実行できない場合、pip のバージョンが古い可能性があります。

「Python2.7 インストール」の「⑧pip のバージョンアップ」(11p)を参照。

③ インストール完了

```
> Successfully installed numpy-1.13.3+mkl
```

と画面に出力されればインストール完了です。

2.3.3. python-dateutil インストール

① インストールファイルの入手

pip を使って最新のファイル入手します。

コマンドプロンプトより下記のコマンドを実行してください。

```
> pip download python-dateutil
```

※以下のファイルがカレントディレクトリにダウンロードされます。

```
python_dateutil-[X.X.X]-py2.py3-none-any.whl
```

[X.X.X]はバージョンに対応しています。

本書では最新バージョンの python_dateutil-2.6.1-py2.py3-none-any.whl

を用いて説明します。

② インストール実行

pip を使ってインストールを行います。

コマンドプロンプトより下記のコマンドを実行してください。

```
> pip install python_dateutil-2.6.1-py2.py3-none-any.whl
```

③ インストール完了

```
> Successfully installed python-dateutil-2.6.1
```

と画面に出力されればインストール完了です。

2.3.4. FreeGLUT インストール

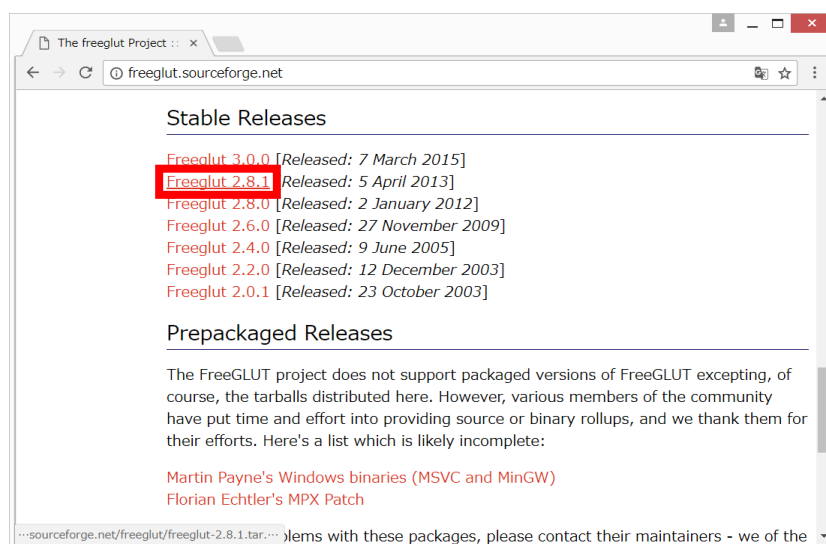
FreeGLUT のインストールは OpenCV のようにソースファイルよりコンパイルを行う必要があります。大きな流れは OpenCV とほぼ同じですが、ソリューションファイルが既に用意されていますので Cmake は行う必要はありません。また、環境変数の設定も必要ありません。

① インストールファイルの入手

インストールファイルを公式サイトより入手します。

<http://freeglut.sourceforge.net/>

対象ファイル名：freeglut-2.8.1.tar.gz



② インストールファイルの解凍

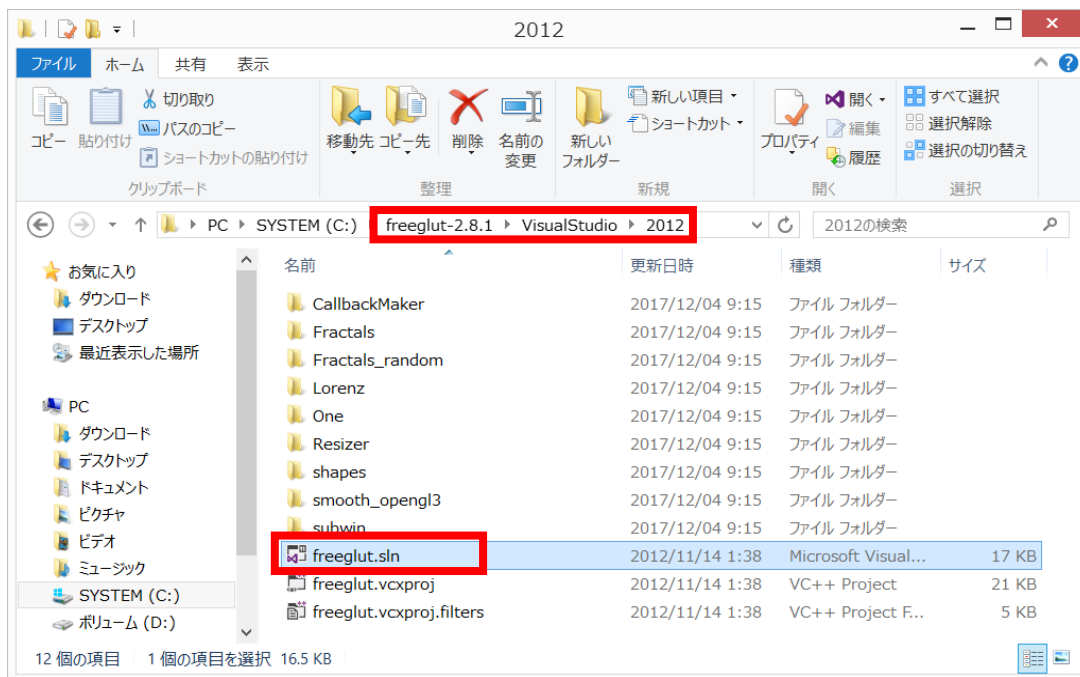
任意の解凍ソフトを用いて解凍してください。

解凍先は任意ですが本書では C ドライブ直下に解凍した例を用いて説明します。

③ コンパイル実行

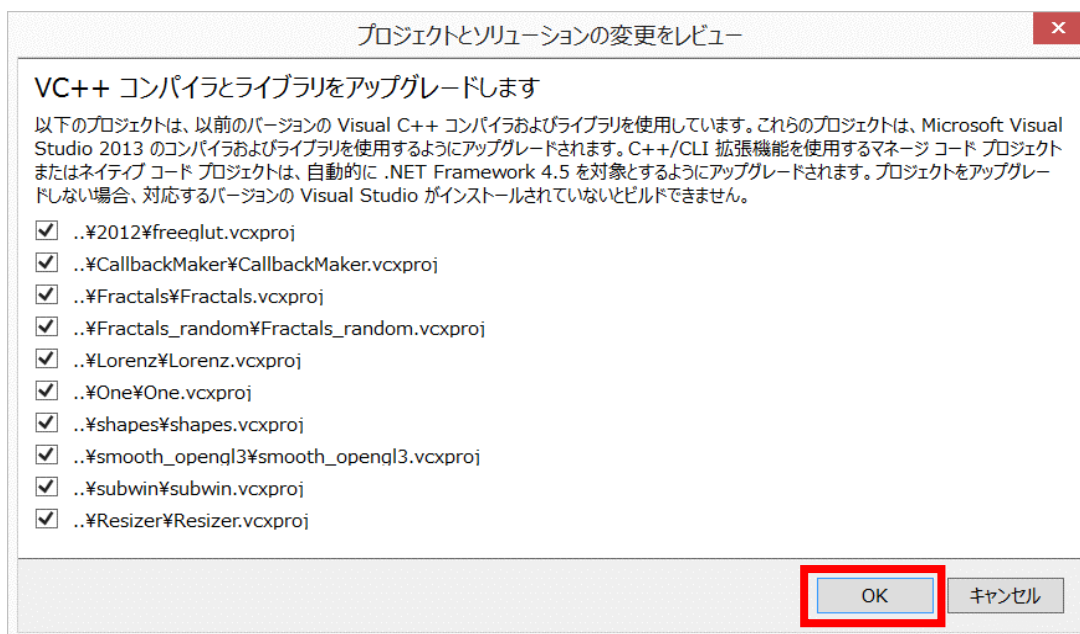
¥freelut-2.8.1¥VisualStudio¥2012¥freelut.sln を開きます。

※VisualStudio フォルダ配下に複数のバージョンがありますが、必ず”2012”フォルダのソリューションを使用してください。

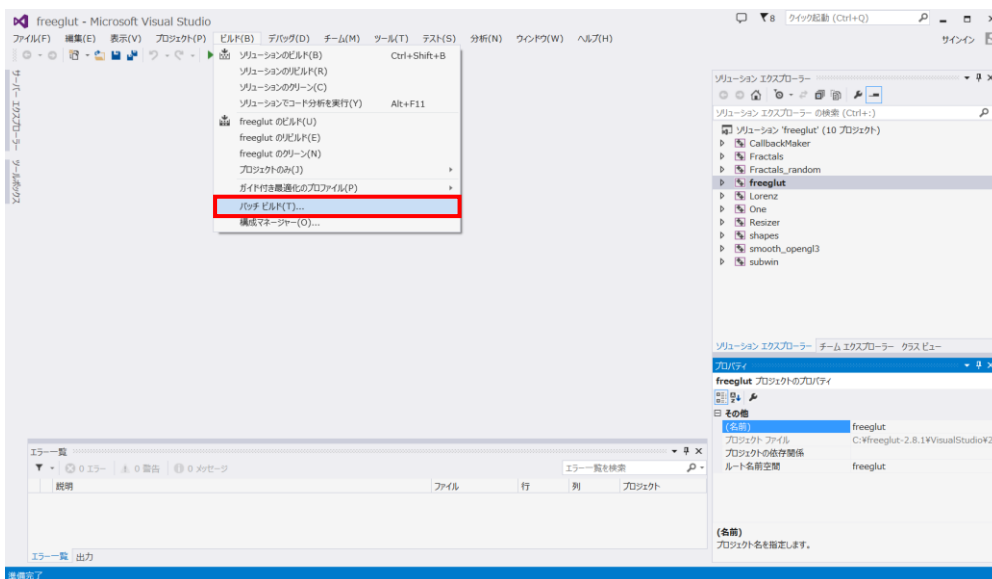


初回展開時に下のダイアログが表示されます。

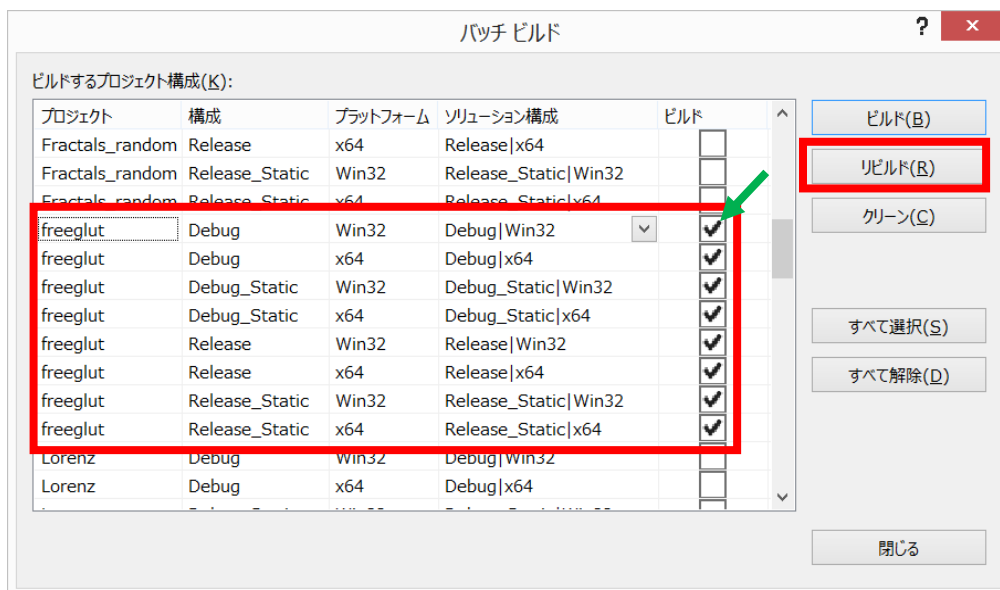
全てチェックが ON である事を確認し、OK ボタンを押下します。



ビルドメニューからバッチビルドを押下します。



プロジェクトの項目が”freelut”である行のビルドのチェックボックスにチェックしてリビルドボタンを押下します。

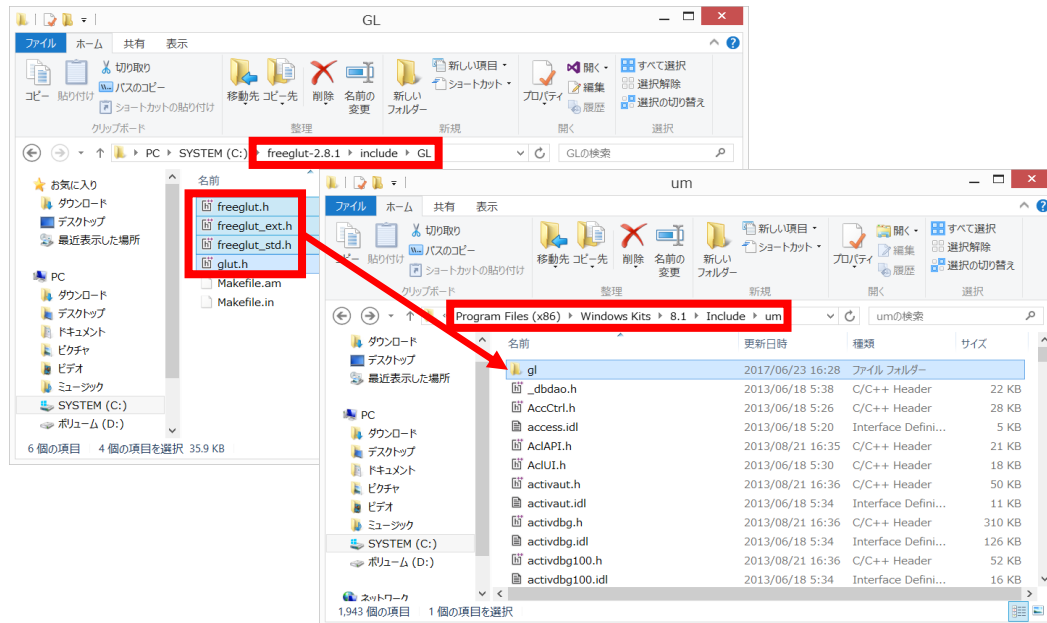


④ ヘッダファイルのコピー

Visual Studio から認識できる場所にヘッダファイルをコピーします。

¥¥freelut-2.8.1¥¥include¥¥GL フォルダ内のヘッダファイル(*.h)を下記パスにコピーします。

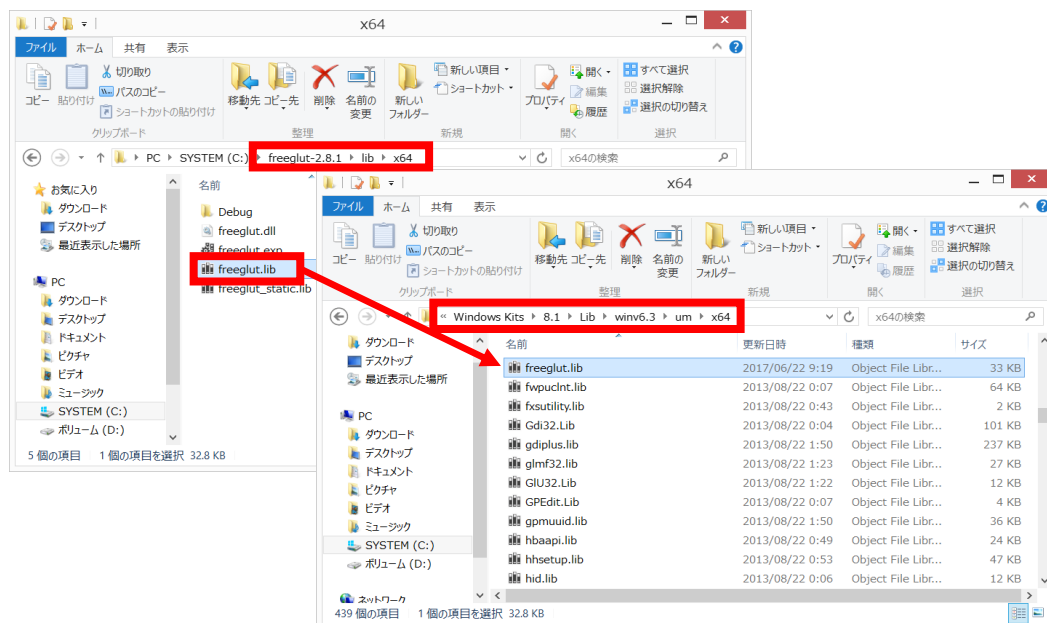
C:¥¥Program Files (x86)¥¥Windows Kits¥¥8.1¥¥Include¥¥um¥¥gl



⑤ 作成された Lib のコピー

¥¥freelut-2.8.1¥¥lib¥¥x64 フォルダ内の freeglut.lib を下記パスにコピーします。

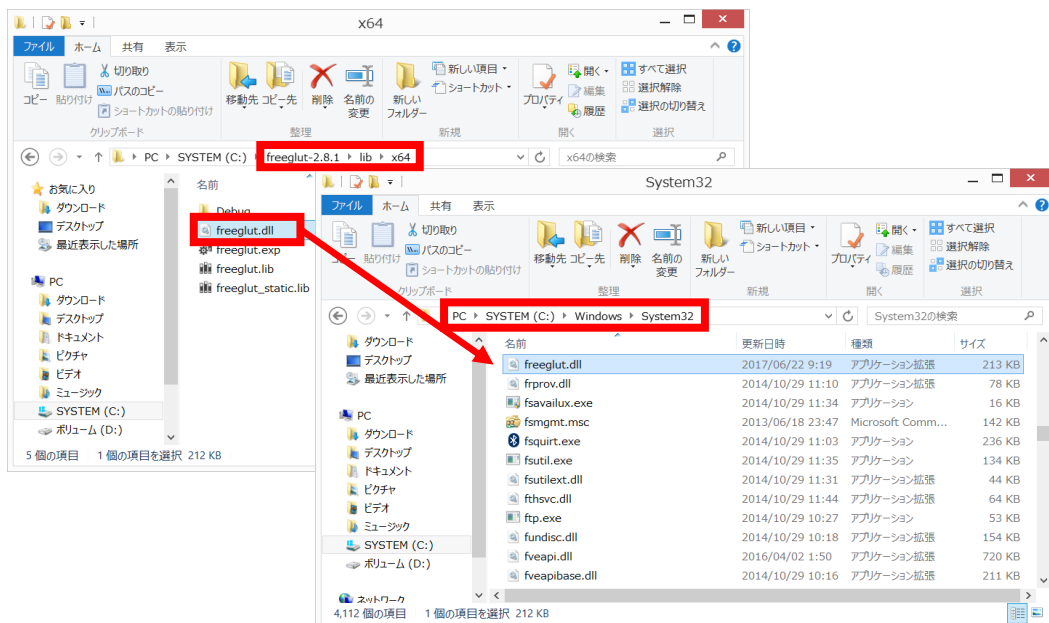
C:¥¥Program Files (x86)¥¥Windows Kits¥¥8.1¥¥Lib¥¥winv6.3¥¥um¥¥x64



⑥ 作成された Bin のコピー

¥¥freelut-2.8.1¥¥lib¥¥x64 フォルダ内の freelut.dll を下記パスにコピーします。

C:¥¥Windows¥¥System32



⑦ インストール完了

2.3.5. PyQt インストール

① インストールファイルの入手

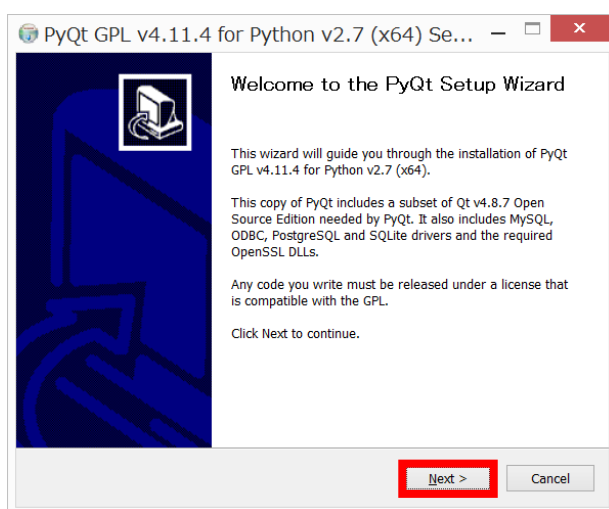
SourceForge.net からインストールファイル入手します。

<https://sourceforge.net/projects/pyqt/files/PyQt4/PyQt-4.11.4/PyQt4-4.11.4-gpl-Py2.7-Qt4.8.7-x64.exe/download>

② インストールファイルの実行

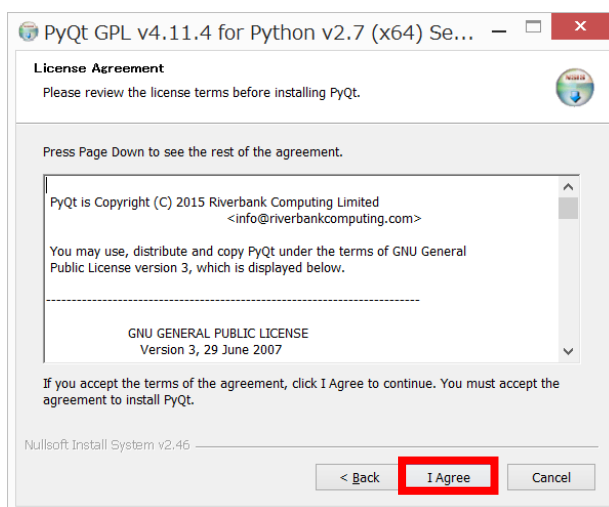
「PyQt4-4.11.4-gpl-Py2.7-Qt4.8.7-x64.exe」を実行します。

Next ボタンを押下します。



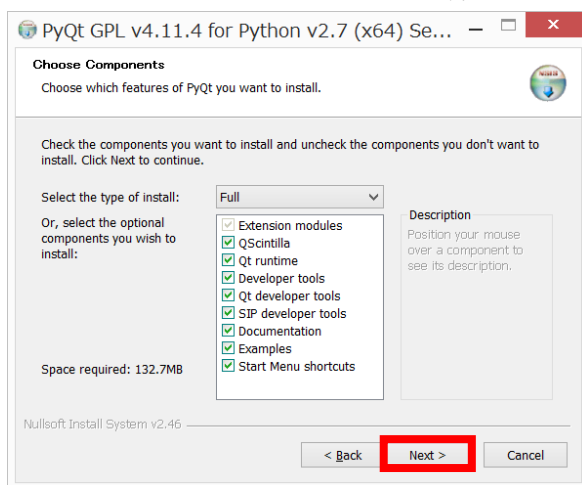
③ ライセンス条項の同意

I Agree ボタンを押下します。



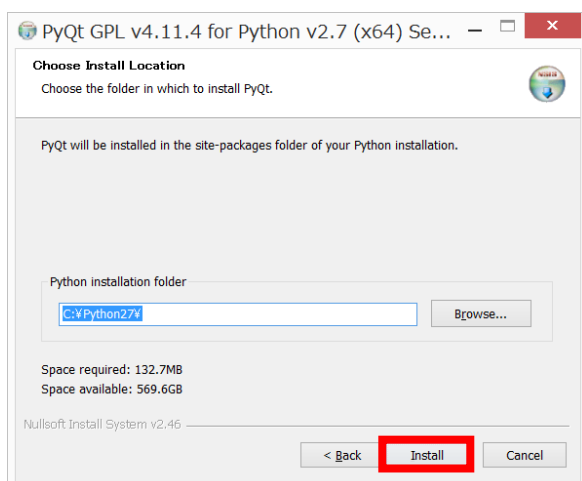
④ インストールオプション選択

デフォルトのまま Next ボタンを押下します。



⑤ インストール先の選択

デフォルトのまま Install ボタンを押下します。



⑥ インストール完了

2.3.6. PCL インストール

① インストールファイルの入手

以下のサイトからインストールファイルを入手します。

<http://unanancyowen.com/pcl18/>



② インストールファイルの実行

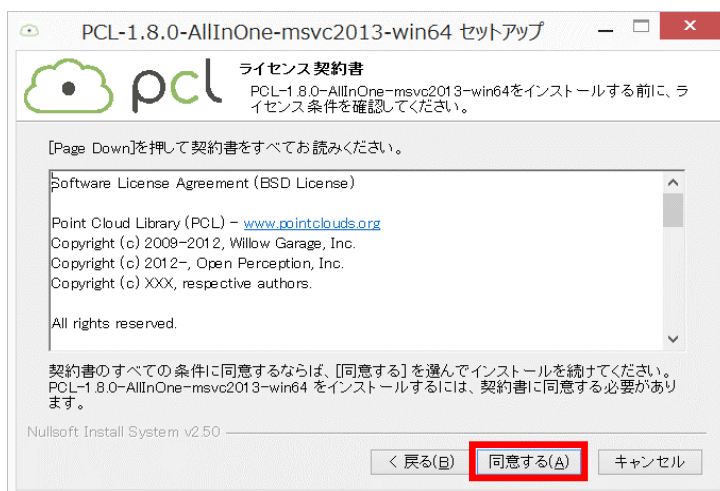
「PCL-1.8.0-AllInOne-msvc2013-win64.exe」を実行します。

次へボタンを押下します。



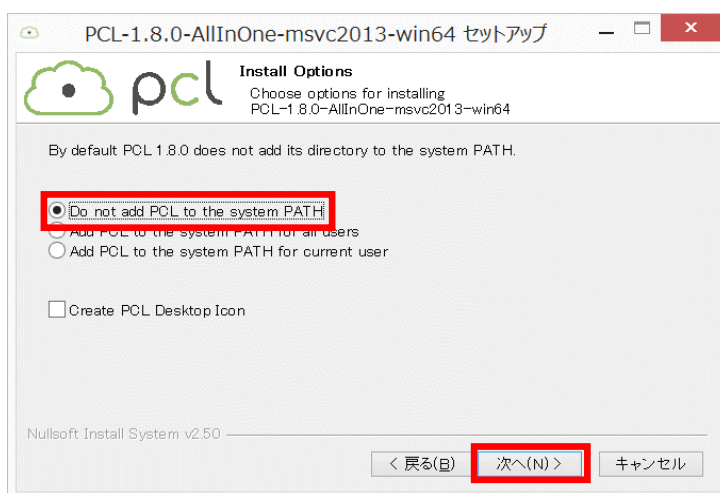
③ ライセンス条項の同意

同意するボタンを押下します。



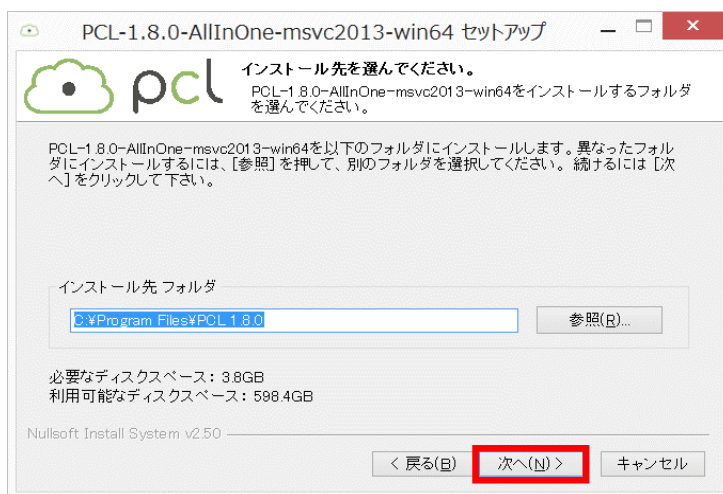
④ インストールオプション選択

「Do not add PCL to the system PATH」ラジオボタンを選択し、次へボタンを押下します。



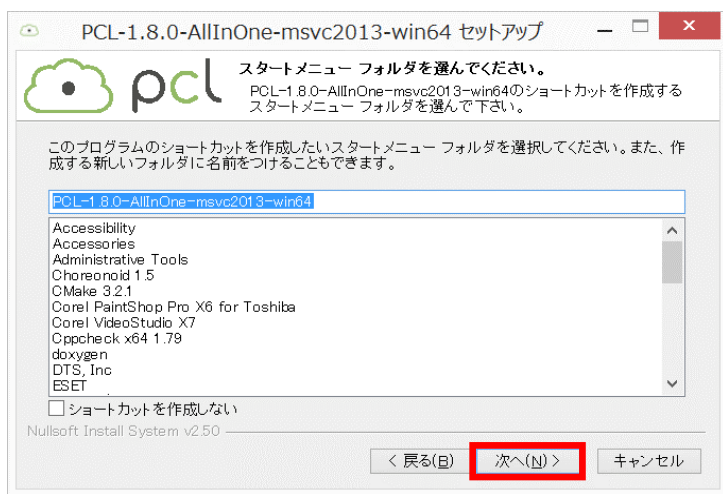
⑤ インストール先の選択

デフォルトのまま次へボタンを押下します。



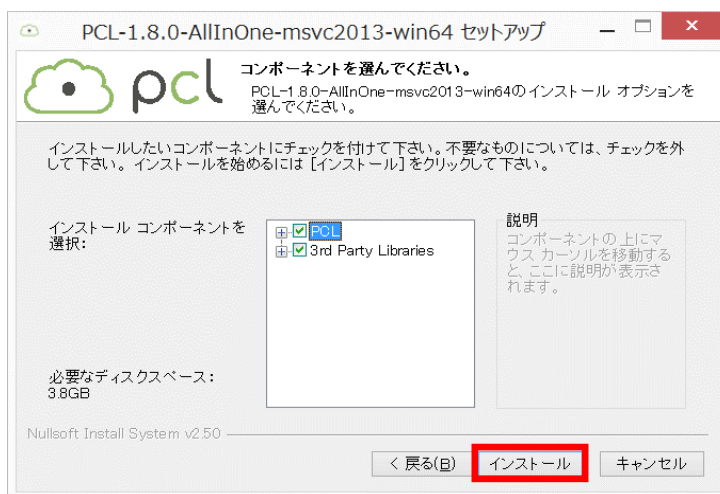
⑥ スタートメニューフォルダ選択

デフォルトのまま次へボタンを押下します。



⑦ コンポーネント選択

デフォルトのままインストールボタンを押下します。



※途中で OpenNI のインストーラが実行される場合があります。

ウィザードに沿ってデフォルトのままインストールしてください。

⑧ インストール完了

⑨ 環境変数設定

環境変数を下記表に合わせて設定します。

変数名	変数値
PCL_ROOT	C:%Program Files%PCL 1.8.0
Path	%PCL_ROOT%bin
	%OPENNI2_REDIST64%
	C:%Program Files%PCL 1.8.0%3rdParty%Boost%include%boost-1_61

⑩ コンピュータの再起動

環境変数を適用するため、OS をリブートします。

△ プロジェクトのコンパイルエラー時に C1001 が出る場合

PCL ライブラリを含んだプロジェクトをコンパイルする際に、C1001 エラーが発生する場合があります。これは Visual Studio 2013 のバグの可能性が高くパッチを当てることにより、エラーが解消される場合があります。

以下、マイクロソフトサポートサイト及び、パッチファイルダウンロード先
<https://support.microsoft.com/en-hk/help/3185410/c1001-internal-compiler-error-occurs-when-you-build-a-visual-c-project>

サポートサイトで配布されているパッチ「vs12-kb3185410.exe」を適用する。

3. 開発に際しての注意点

OpenCV 及び、PCL ライブラリを含むコンパイルを行う際は設定が必要です。また、Windows 標準の PathFileExists を参照する際の設定も合わせて記載します。

3.1. OpenCV を使用する際の設定

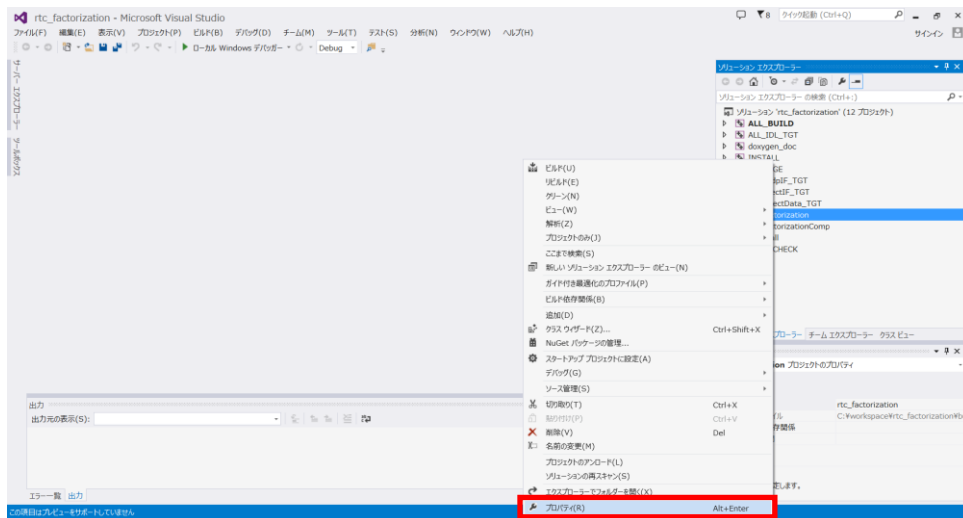
OpenCV のコンパイルにはコンパイルオプションの設定が必要です。

3.1.1. OpenCV 使用時のコンパイルオプション

OpenRTM にてコンポーネント開発を行う際のコンパイルオプション設定手順を記載します。下記に示す手順を「対象プロジェクト」の「Debug 及び Release」並びに「Comp プロジェクト」の「Debug 及び Release」の計 4 回の設定が必要です。

① プロパティ画面を開く

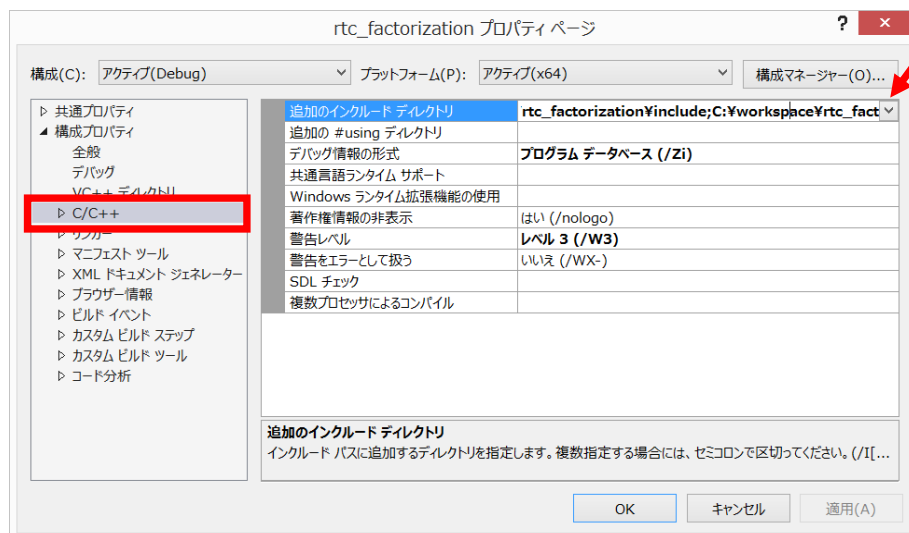
対象のプロジェクトから[右クリック]-[プロパティ]を開きます。



② インクルードディレクトリの設定

C/C++の[追加のインクルードディレクトリ]欄にある下向き三角を押下・編集画面を展開し、下記パスを追加する。

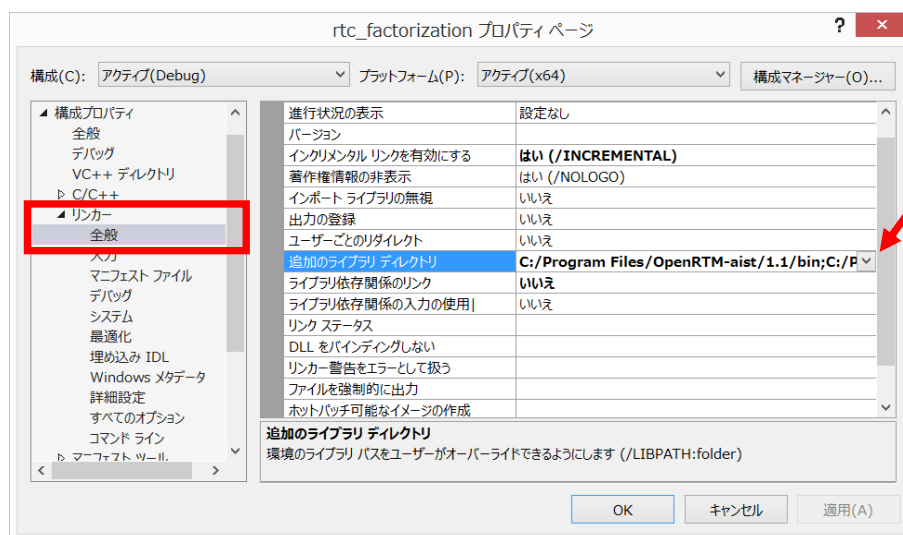
“C:\¥opencv3.0¥opencv¥build¥include”



③ ライブラリディレクトリの設定

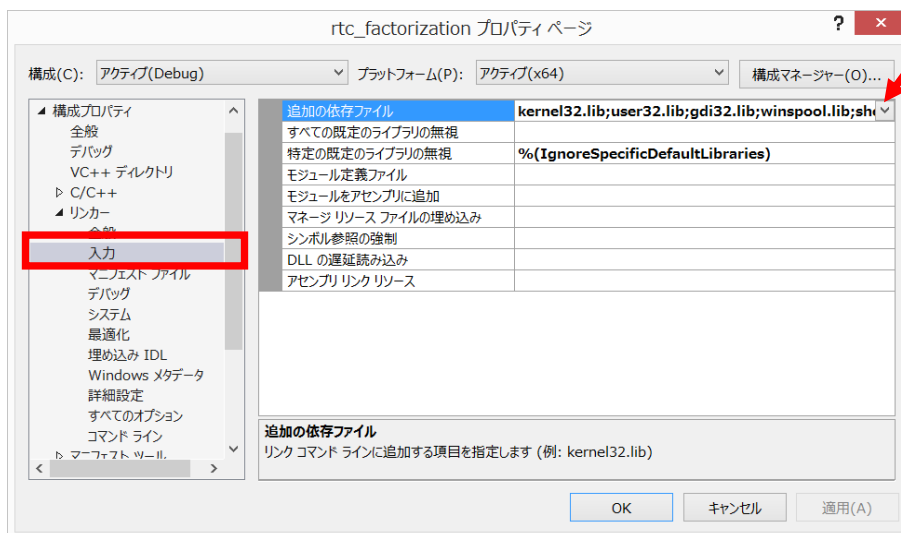
[リンカー]-[全般]の[追加のライブラリディレクトリ]欄にある下向き三角を押下・編集画面を展開し、下記パスを追加する。

“C:\¥opencv3.0¥opencv¥build¥x64¥vc12¥lib”



④ 依存ライブラリの設定

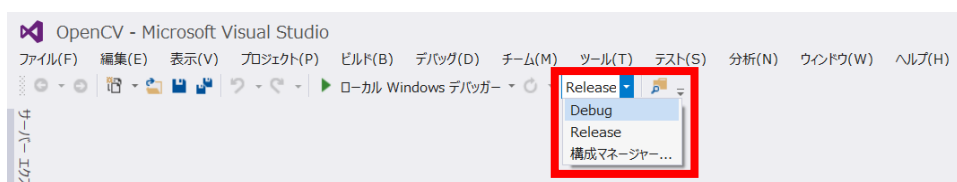
[リンカー]-[入力]の[追加の依存ファイル]欄にある下向き三角を押下・編集画面を展開し、下記表のファイル名を追加する。



なお、コンパイルモード(※)により設定するファイル名が異なりますので注意が必要です。また、Debug・Release 両方の設定が必要です。

コンパイルモード	ファイル名
Debug	opencv_ts300d.lib
	opencv_world300d.lib
	opencv_imgproc300d.lib
	opencv_core300d.lib
	opencv_highgui300d.lib
	opencv_calib3d300d.lib
Release	opencv_ts300.lib
	opencv_world300.lib
	opencv_imgproc300.lib
	opencv_core300.lib
	opencv_highgui300.lib
	opencv_calib3d300.lib

※コンパイルモードの変更



⑤ Comp プロジェクトの設定

①～④の設定を対象プロジェクトの Comp プロジェクトに対しても行います。

3.2. PCL を使用する際の設定

PCL を使用するには Cmake ファイルの修正が必要です。修正後の Cmake ファイルより作成したソリューションファイルを用いてコンパイルしてください。

3.2.1. PCL 使用時の Cmake ファイル

① Cmake ファイルの展開

RTCBuilder により作成された Cmake ファイルの修正を行います。下記パスに保存されている Cmake ファイルをエディタで展開します。

¥¥ [プロジェクト名]¥src¥ CMakeLists.txt

② Cmake ファイルの修正

下記表に合わせて、追加修正を行ってください。

修正箇所	値	修正方法
include_directories の前の行	find_package(PCL 1.8.0 REQUIRED)	追加挿入
include_directories の次の行	include_directories(\${PCL_INCLUDE_DIRS})	追加挿入
add_definitions の次の行	add_definitions(\${PCL_DEFINITIONS})	追加挿入
link_directories の次の行	link_directories(\${PCL_LIBRARY_DIRS})	追加挿入
target_link_libraries(\${PROJECT_NAME} \${OPENRTM_LIBRARIES}) で検索	target_link_libraries(\${PROJECT_NAME} \${OPENRTM_LIBRARIES} \${PCL_LIBRARIES})	置換編集
target_link_libraries(\${PROJECT_NAME}Comp \${OPENRTM_LIBRARIES})で検索	target_link_libraries(\${PROJECT_NAME}Comp \${OPENRTM_LIBRARIES} \${PCL_LIBRARIES})	置換編集

③ Cmake 実行

修正した Cmake ファイルを用いて、Cmake を実行しソリューションファイルを作成してください。

3.3. Windows 標準の PathFileExists を使用する際の設定

PathFileExists を使用する場合、コンパイルオプションの設定が必要です。

3.3.1. PathFileExists 使用時のコンパイルオプション

OpenCV と同様にプロジェクトのプロパティからコンパイルオプションの設定を行います。但しインクルードディレクトリ及び、ライブラリディレクトリの指定は不要で、依存ライブラリのみを設定となります。

① 依存ライブラリの設定

[リンカー]-[入力]の[追加の依存ファイル]欄にある下向き三角を押下・編集画面を展開し、“shlwapi.lib”を追加する。

② 全てのプロパティに適応

①の設定を「対象プロジェクト」の「Debug 及び Release」並びに「Comp プロジェクト」の「Debug 及び Release」の計4回行います。

4. 参考

3D マップ生成システムの開発環境構築にあたり、参考にしたサイトを記載します。

- OpenRTM インストール
OpenRTM-aist 公式サイト : <http://openrtm.org/openrtm/>

- OpenCV 及び PyQT インストール
Geeknet 社 : <https://sourceforge.net/>

- NumPy インストール
カリフォルニア大学アーバイン校 : <https://www.lfd.uci.edu/>

- FreeGULT インストール
FreeGULT 公式サイト : <http://freeglut.sourceforge.net/>

- PCL インストール
PCL Official Maintainer 杉浦 司 氏 : <http://unanancyowen.com/>

著作権

本文書の著作権は公立大学法人 会津大学に帰属します。

この文書のライセンスは以下のとおりです。

[クリエイティブ・コモンズ 表示 2.1 日本](http://creativecommons.org/licenses/by/2.1/jp/)

<http://creativecommons.org/licenses/by/2.1/jp/>

