

# OpenCV 설치

## □ OpenCV 설치란?

- OpenCV 헤더 파일, LIB 파일, DLL 파일을 컴퓨터에 생성하는 작업

## □ OpenCV 설치 방법

### 1. OpenCV 설치 실행 파일로 설치하기

- <http://opencv.org/releases.html> 에서 설치 파일 다운로드

### 2. OpenCV 소스 코드 다운받아 설치하기

#### 1) 최신 GIT 소스 코드 다운받아 설치하기

- <https://github.com/opencv/opencv>
- [https://github.com/opencv/opencv\\_contrib/](https://github.com/opencv/opencv_contrib/)

#### 2) 공식 Release (또는 Tag) 소스 코드 다운받아 설치하기

- <https://github.com/opencv/opencv/releases>
- [https://github.com/opencv/opencv\\_contrib/releases/](https://github.com/opencv/opencv_contrib/releases/)

# OpenCV 설치

## □ OpenCV 설치 방법에 따른 장단점

### 1. OpenCV 설치 실행 파일로 설치하기

- Windows 운영체제에서 Visual Studio 2015/2017 x64 환경에서 미리 빌드된 LIB & DLL 파일 제공
- CUDA, Extra module 지원하지 않음



### 2. OpenCV 소스 코드 다운받아 설치하기

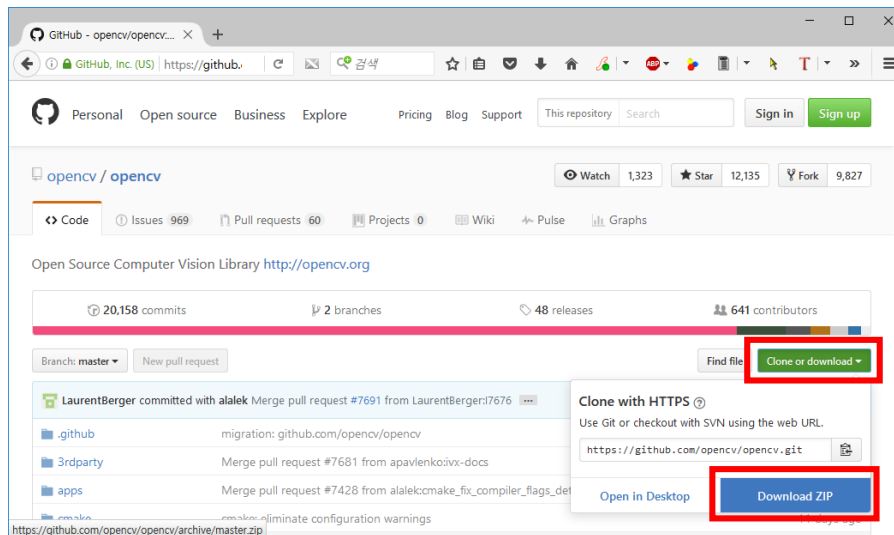
- 사용하고 있는 컴퓨터 시스템에 최적화된 OpenCV 빌드 (CPU 최적화 옵션 [링크](#) 참고, 기본 빌드는 SSSE3 사용)
- 원하거나 원치 않는 옵션 선택 가능. (CUDA, Extra module, parallel\_for backend, etc)
- 별도의 빌드 작업이 필요.

- CPU optimizations build options: <https://github.com/opencv/opencv/wiki/CPU-optimizations-build-options>

# OpenCV 설치 – 소스 코드 다운로드

## □ OpenCV 최신 소스 코드 다운받기

- <https://github.com/opencv/opencv>
- [https://github.com/opencv/opencv\\_contrib/](https://github.com/opencv/opencv_contrib/)



- **장점:** 최신 소스 (버그 패치, 신기능)
- **단점:** (잠재적) 버그?

- 또는 git 프로그램을 이용

```
c:\opencv> git clone https://github.com/opencv/opencv.git  
c:\opencv> git clone https://github.com/opencv/opencv_contrib.git
```

# OpenCV 설치 – 소스 코드 다운로드

## □ OpenCV 공식 릴리즈 소스 코드 다운로드

■ 장점: Stable!

■ <https://github.com/opencv/opencv/releases/>

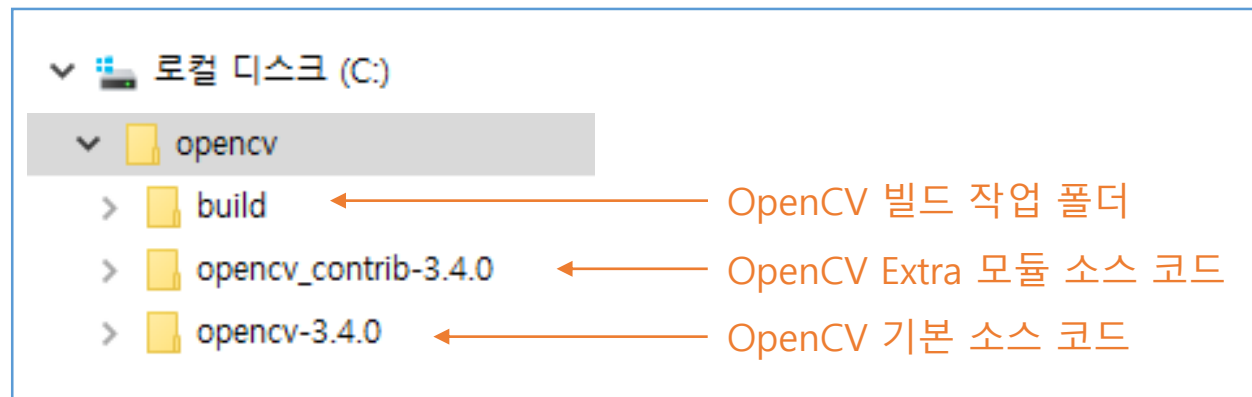
■ [https://github.com/opencv/opencv\\_contrib/releases/](https://github.com/opencv/opencv_contrib/releases/)

opencv-3.4.0.zip 파일과  
opencv\_contrib-3.4.0.zip 파일  
다운로드!!

# OpenCV 설치 – 소스 코드 다운로드

## □ OpenCV 소스 코드 빌드를 위한 폴더 구조 만들기

1. **C:\opencv\** 폴더 만들기
2. C:\opencv\ 폴더 아래에 opencv-3.4.0.zip 파일 압축 풀기  
→ **c:\opencv\opencv-3.4.0\** 폴더 생성됨
3. C:\opencv\ 폴더 아래에 opencv\_contrib-3.4.0.zip 파일 압축 풀기  
→ **c:\opencv\opencv\_contrib-3.4.0\** 폴더 생성됨
4. **C:\opencv\build** 폴더 만들기

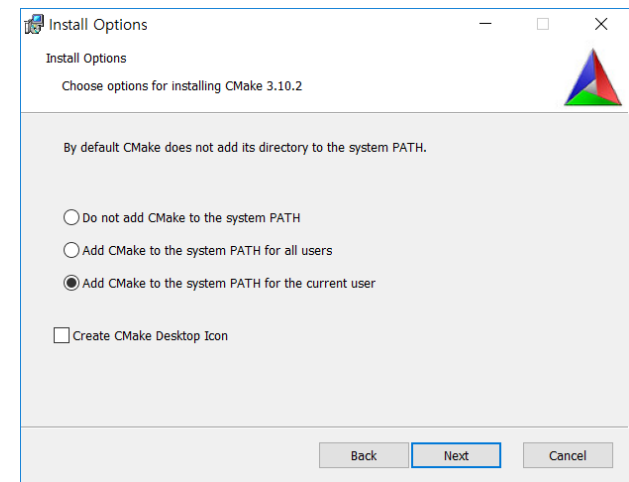
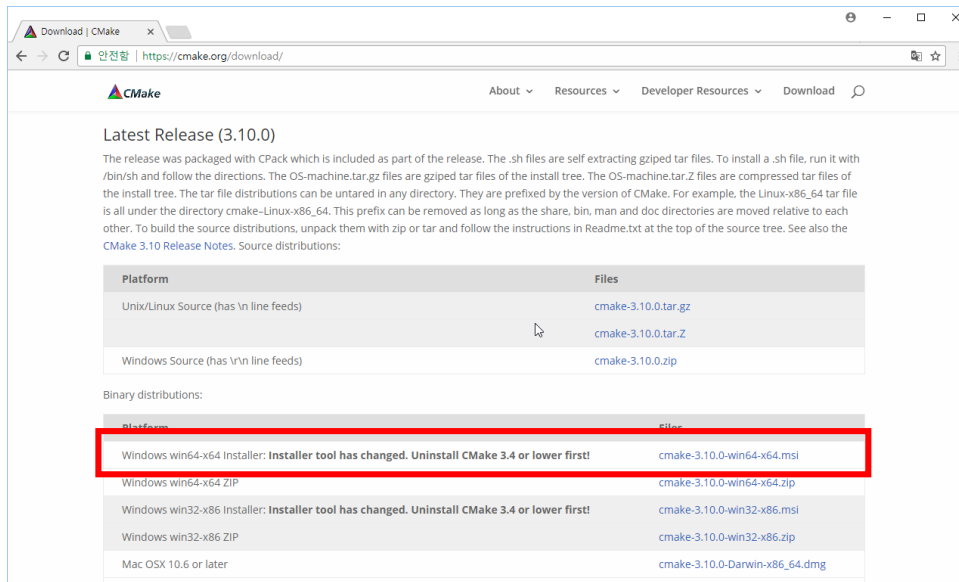


OpenCV 3.3.0 설치 동영상: <https://youtu.be/IhjUSJT60KI>

# OpenCV 설치 – CMake 설정

## □ CMake 프로그램

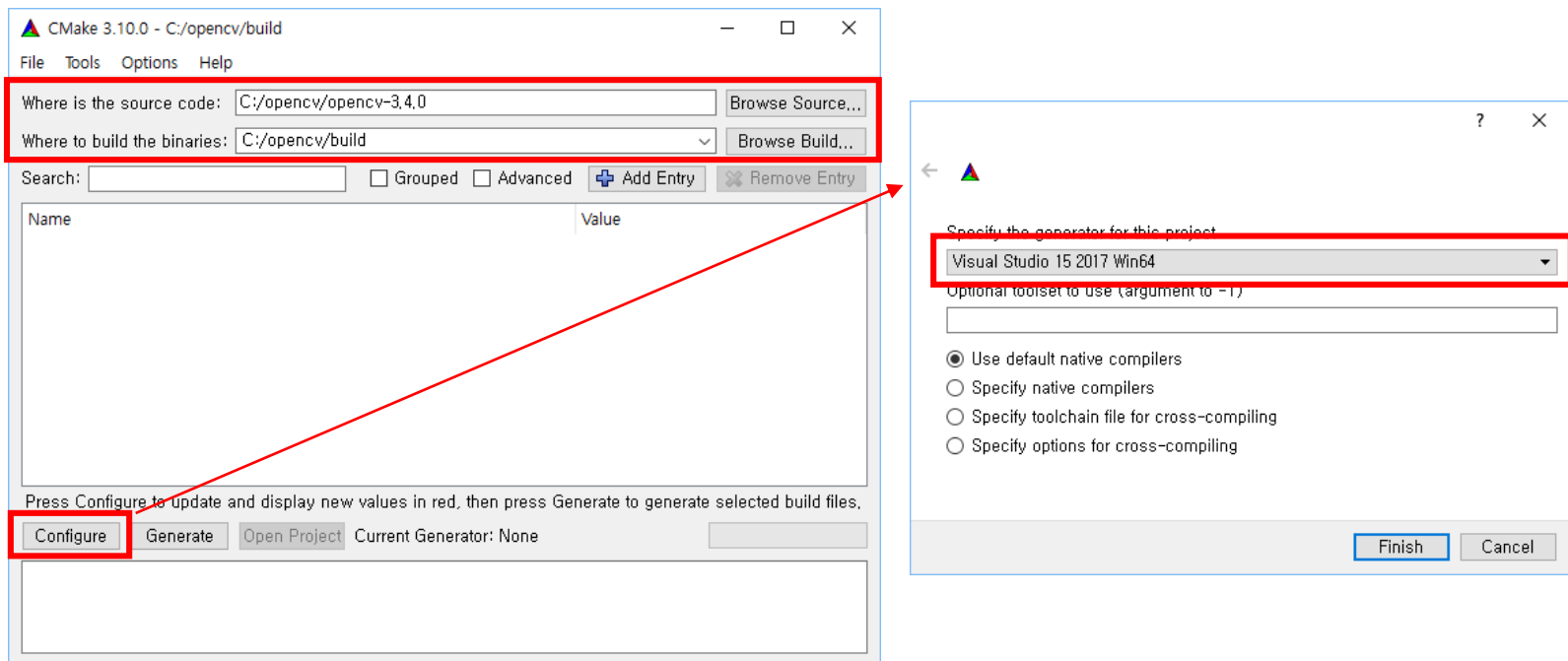
- 다양한 운영 체제(또는 IDE)에 맞는 Make 파일(또는 프로젝트, 솔루션 파일)을 생성하는 유틸리티
- <https://cmake.org/download/> 에서 Latest Release (3.10.2) 설치 파일 다운로드 & 실행
- CMake 폴더를 시스템 PATH에 추가 권장



# OpenCV 설치 – CMake 설정

□ CMake를 이용하여 Visual Studio 솔루션 파일 생성하기

- ① Where is the source code: <C:/opencv/opencv-3.4.0>
- ② Where to build the binaries: <C:/opencv/build>
- ③ [Configure] 버튼 선택: **“Visual Studio 15 2017 Win64”** 선택



# OpenCV 설치 – CMake 설정

□ CMake를 이용하여 Visual Studio 솔루션 파일 생성하기 (Con't)

④ 아래와 같이 빌드 설정 변경

- **BUILD\_EXAMPLES**      **선택** (Optional)
- **BUILD\_opencv\_world**      **선택**
- **ENABLE\_CXX11**      **선택** (Optional)
- **OPENCV\_EXTRA\_MODULES\_PATH** 지정  
(e.g.) **C:/opencv/opencv\_contrib-3.4.0/modules** **선택**
- **WITH\_CUDA**      **선택 해제**
- **WITH\_OPENGL**      **선택**

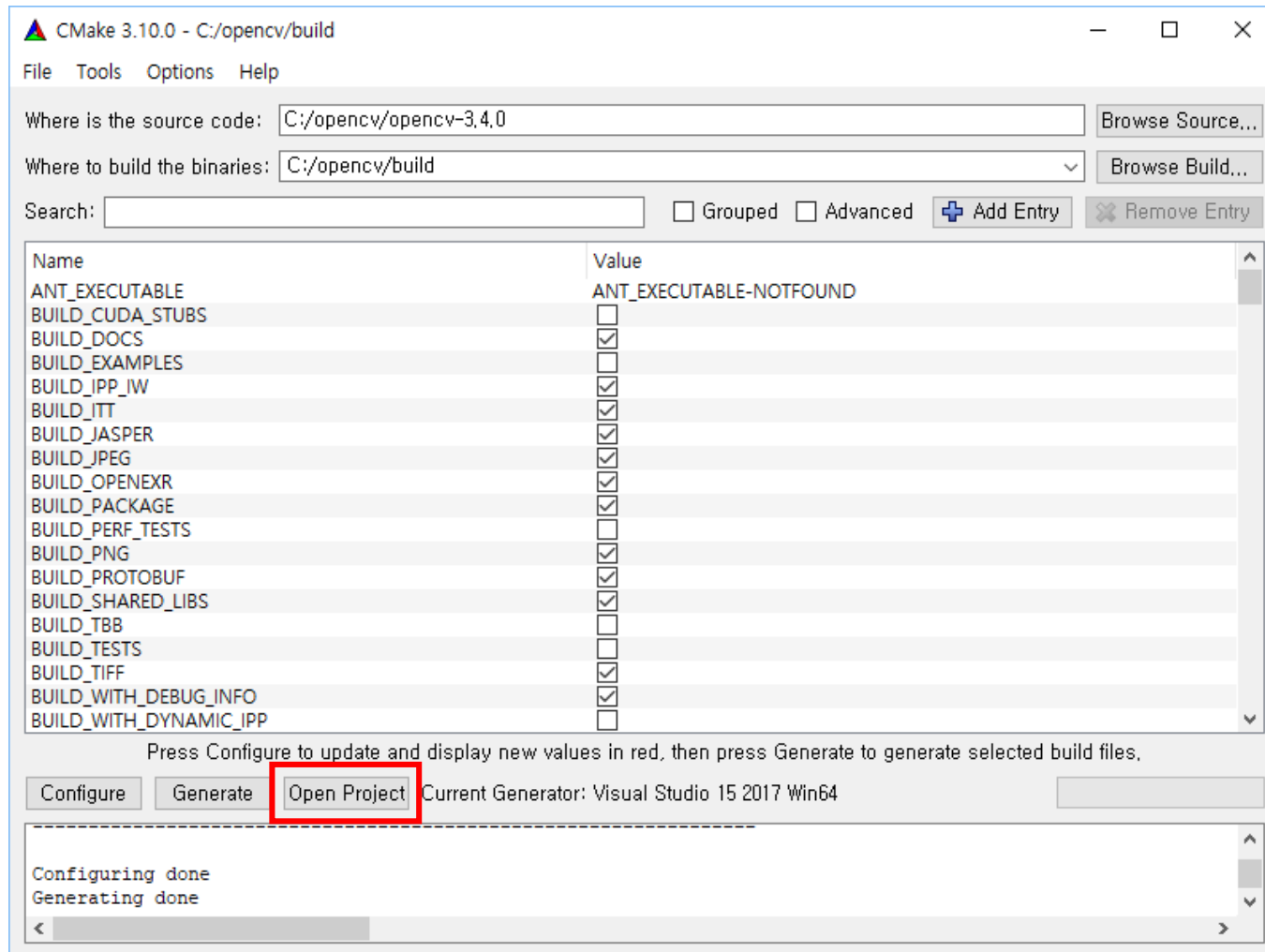
⑤ 다시 [Configure] 버튼 클릭 → Extra 모듈 설정은 기본 설정을 사용

⑥ 다시 [Configure] 버튼 클릭 → 모든 항목이 흰색 배경으로 변경됨

⑦ 이상이 없으면 [Generate] 버튼 클릭



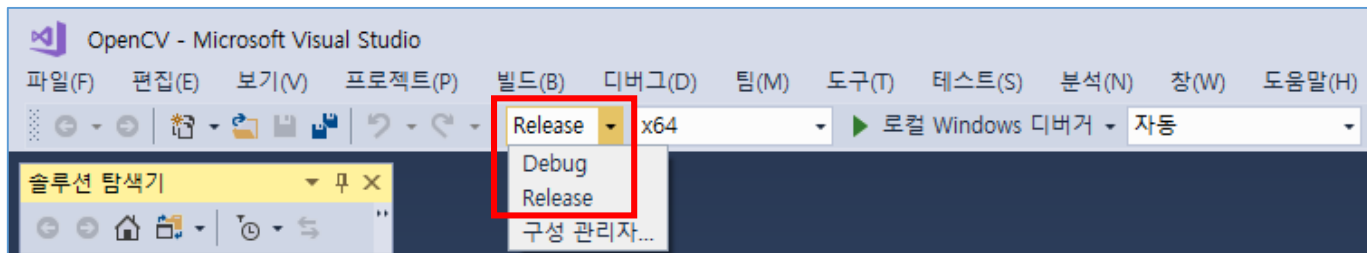
# OpenCV 설치 – CMake 설정



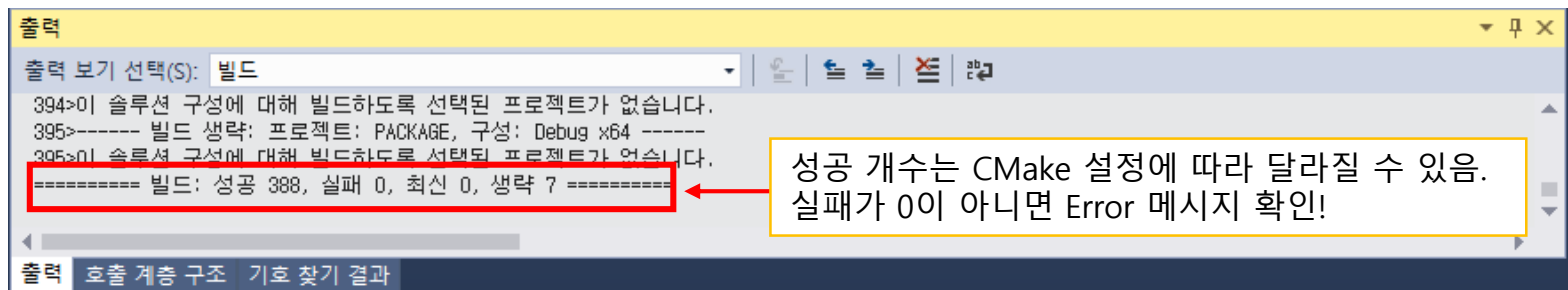
# OpenCV 설치 - 빌드

## □ OpenCV 솔루션 빌드하기

- Visual Studio 2017에서 `C:\opencv\build\OpenCV.sln` 솔루션 파일 열기
- **Debug & Release** 모드로 각각 빌드!



## ■ 빌드 결과 확인

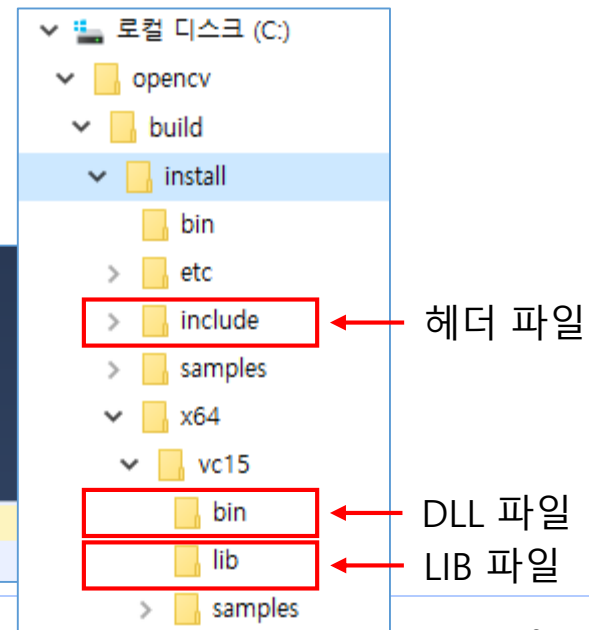
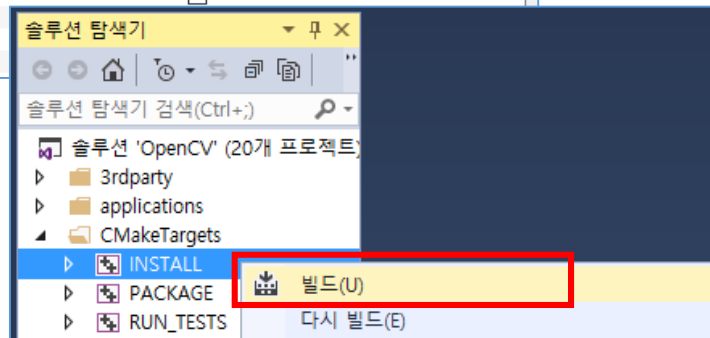
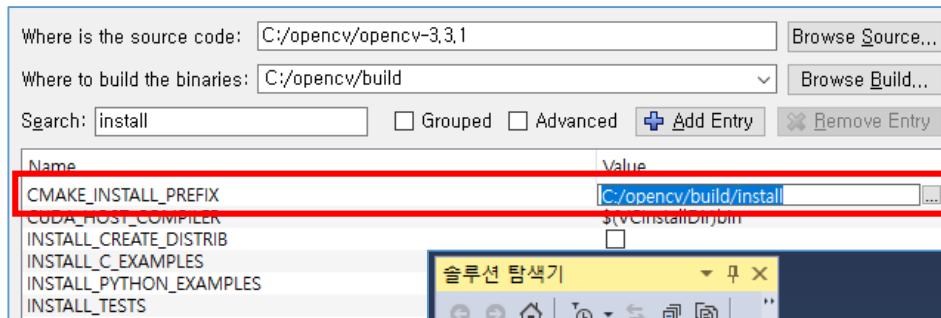


# OpenCV 설치 - 빌드

## □ OpenCV 솔루션 중 **INSTALL** 프로젝트 빌드

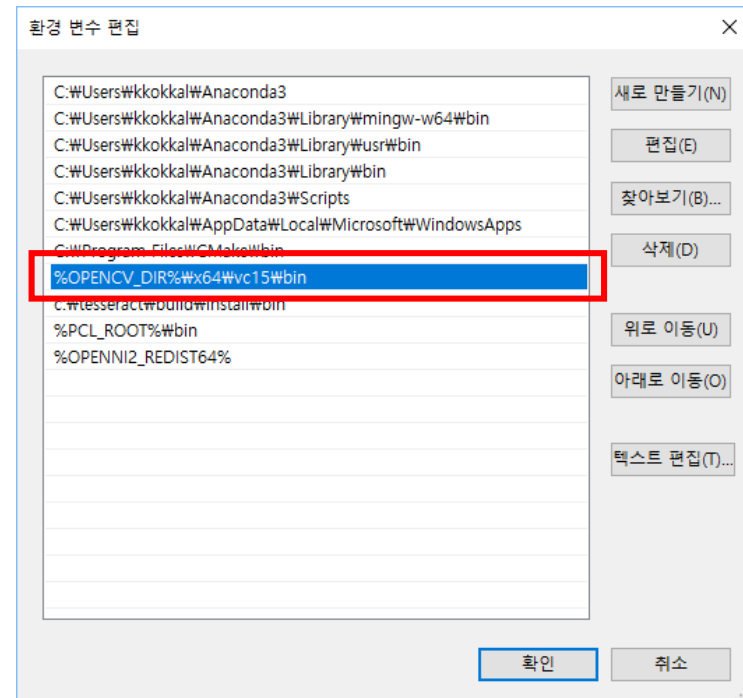
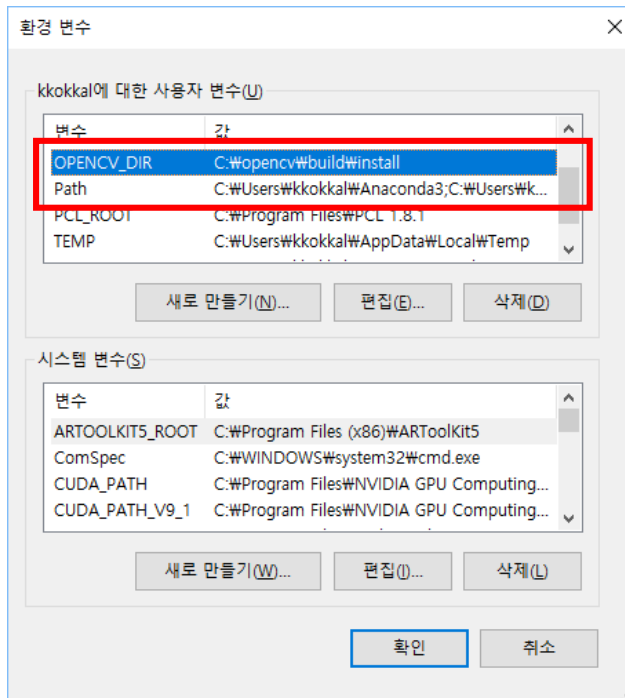
- 새로 빌드한 OpenCV 관련 헤더 파일, lib 파일, dll 파일을 정해진 폴더에 자동으로 복사하는 작업
- **Debug & Release** 모드에서 각각 수행
- CMake에서 설치 폴더 위치가 지정되어 있음

□ **CMAKE\_INSTALL\_PREFIX** : **C:\opencv\build\install**



# OpenCV 설치 – 환경 변수 등록

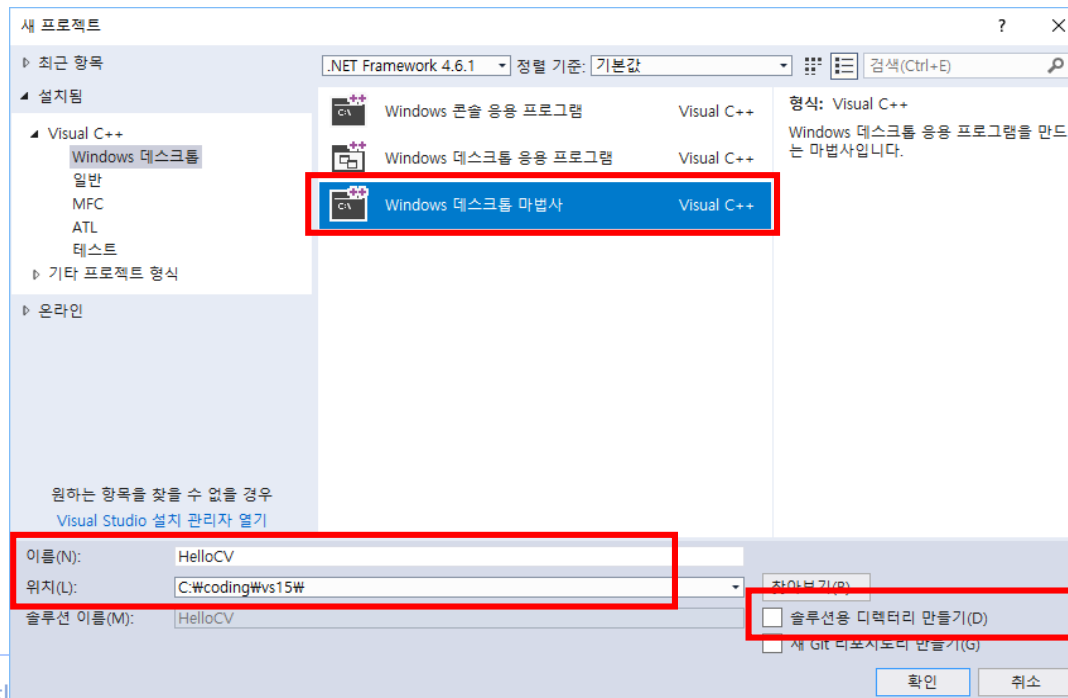
- OpenCV DLL 파일 위치를 PATH 환경 변수에 등록
  - [제어판] → [시스템] → [고급 시스템 설정] → [환경 변수]
  - **OPENCV\_DIR** : C:\wopencv\build\install
  - **PATH** : %OPENCV\_DIR%\x64\vc15\bin 추가



# HelloCV 프로젝트 만들기

## □ Visual C++ 새 프로젝트 생성하기

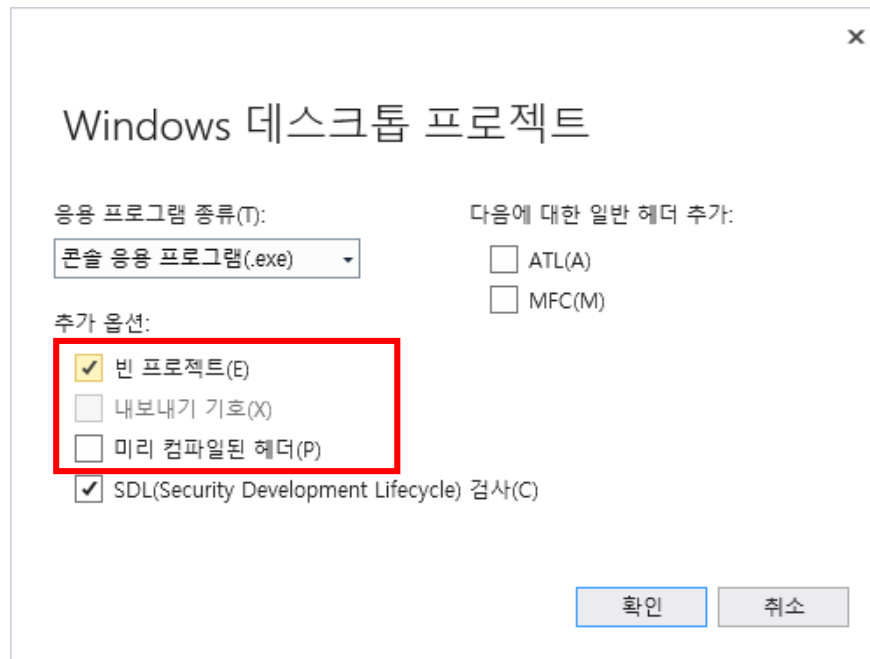
- Visual Studio에서 [파일] → [새로 만들기] → [프로젝트] 메뉴 선택
- 설치된 프로젝트 형식 중에서 “Windows 데스크톱 마법사” 선택
- 프로젝트 이름에 HelloCV 입력
- “솔루션용 디렉터리 만들기” 선택 해제



# HelloCV 프로젝트 만들기

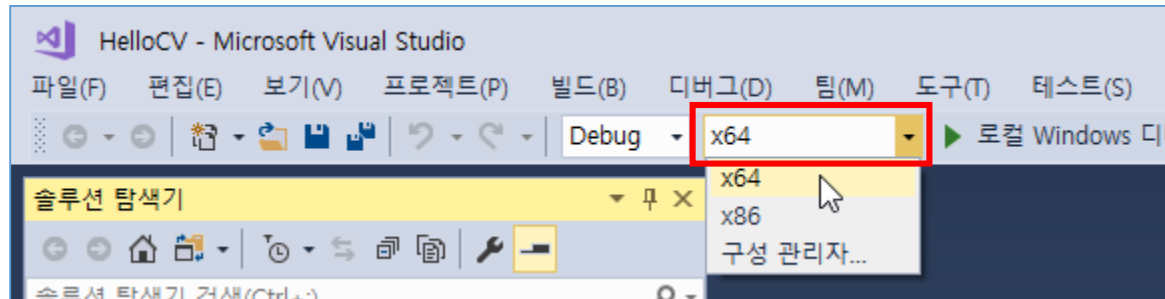
## □ Visual C++ 새 프로젝트 생성하기 (Con't)

- “빈 프로젝트” 선택
- “미리 컴파일된 헤더” 선택 해제



# HelloCV 프로젝트 만들기

- (활성) 솔루션 플랫폼을 x64로 변경



- C++ 소스 코드 입력
  - 메뉴에서 [프로젝트] → [새 항목 추가] 선택하여 `main.cpp` 파일 추가

```
#include "opencv2/opencv.hpp"
#include <iostream>

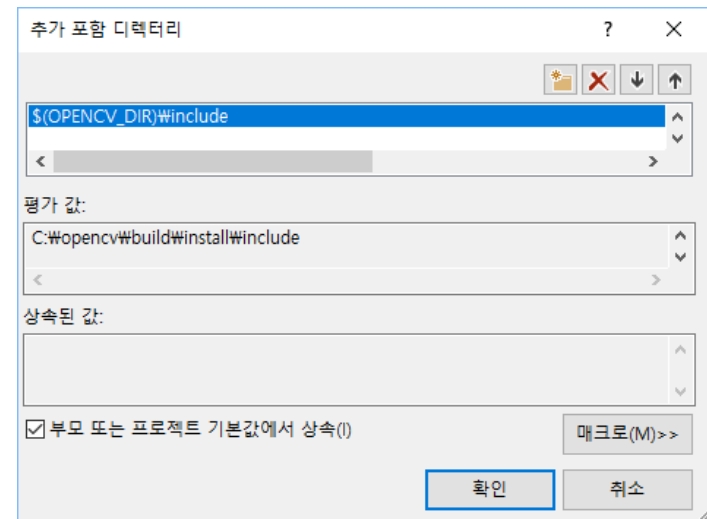
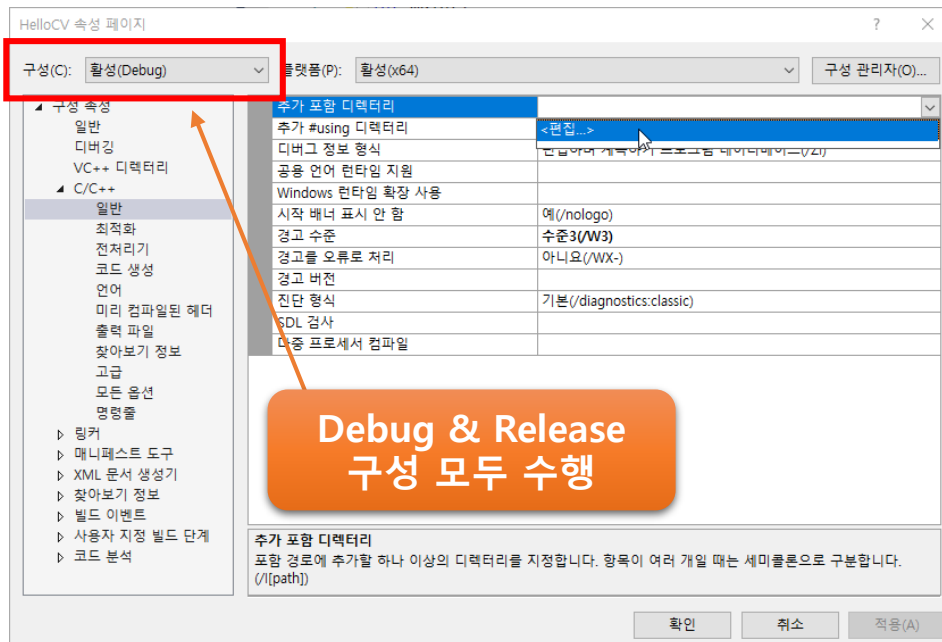
int main()
{
    std::cout << "Hello OpenCV " << CV_VERSION << std::endl;

    return 0;
}
```

# HelloCV 프로젝트에 OpenCV 속성 추가

## □ OpenCV 헤더 파일 위치 지정

- Visual Studio 메뉴에서 [프로젝트] → [HelloCV 속성...] 선택
- 속성 창에서 [구성 속성] → [C/C++] → [일반] 선택
- “추가 포함 디렉터리” 항목에 OpenCV 헤더 파일 디렉터리를 추가 (e.g. `$(OPENCV_DIR)\include`)

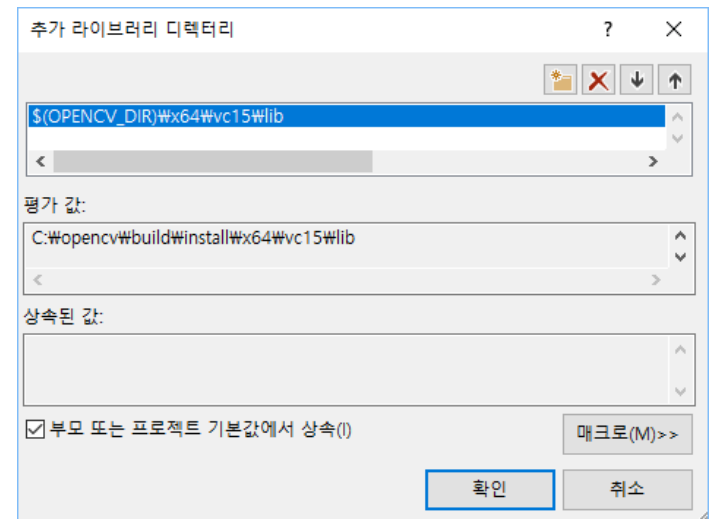
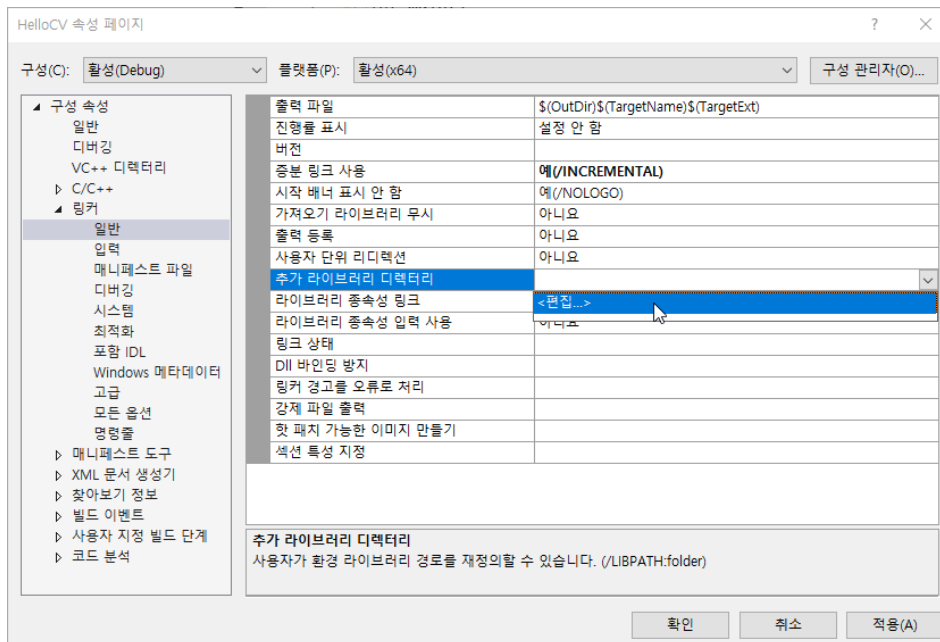




# HelloCV 프로젝트에 OpenCV 속성 추가

## □ OpenCV LIB 파일 위치 지정

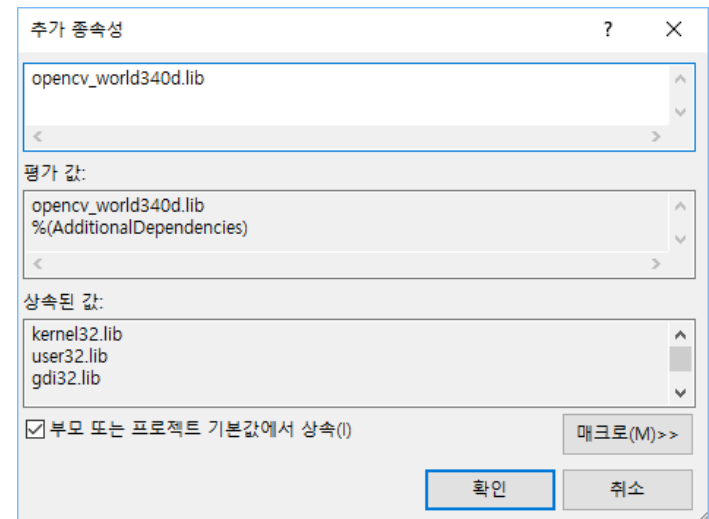
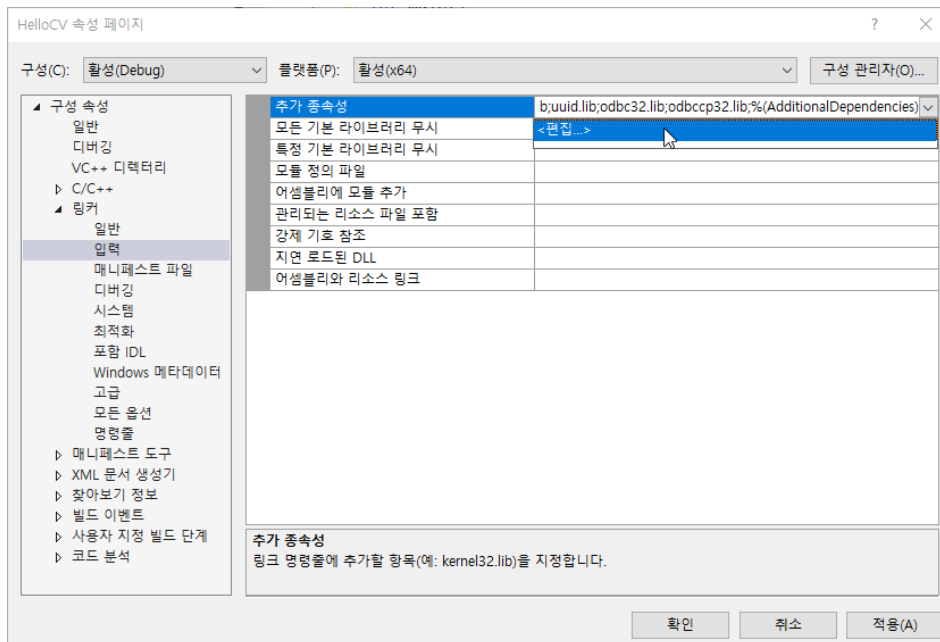
- [프로젝트] → [HelloCV 속성...] 선택
- 속성 창에서 [구성 속성] → [링커] → [일반] 선택
- “추가 라이브러리 디렉터리” 항목에 OpenCV 라이브러리 디렉터리 추가 (e.g. `$(OPENCV_DIR)\x64\vc15\lib`)



# HelloCV 프로젝트에 OpenCV 속성 추가

## □ OpenCV LIB 파일 이름 지정

- [프로젝트] → [HelloCV 속성...] 선택
- 속성 창에서 [구성 속성] → [링커] → [입력] 선택
- “추가 종속성” 항목에 **opencv\_world340d.lib** 추가  
(Release 모드에서는 **opencv\_world340.lib** 추가)



# HelloCV 프로젝트에 OpenCV 속성 추가

- 프로젝트 속성 페이지에서 OpenCV 설정하기 요약
  - (방법 #1) 시스템 속성에서 OPENCV\_DIR 환경 변수를 설정:  
(e.g.) C:\opencv\build\install

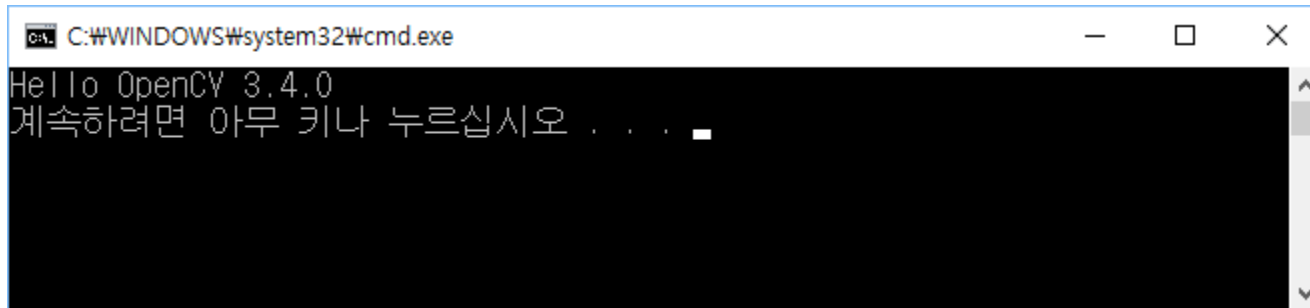
	Debug 모드	Release 모드
추가 포함 디렉터리	\$(OPENCV_DIR)\include	
추가 라이브러리 디렉터리	\$(OPENCV_DIR)\x64\vc15\lib	
추가 종속성	opencv_world340d.lib	opencv_world340.lib

- (방법 #2) 또는 디렉터리 이름을 직접 명시:

	Debug 모드	Release 모드
추가 포함 디렉터리	C:\opencv\build\install\include	
추가 라이브러리 디렉터리	C:\opencv\build\install\x64\vc15\lib	
추가 종속성	opencv_world340d.lib	opencv_world340.lib

# HelloCV 프로젝트 빌드 및 실행

- 프로젝트 빌드
  - [빌드] → [솔루션 빌드] 메뉴 선택 (단축키 [F7])
- 프로그램 디버깅 또는 실행
  - [디버그] → [디버깅 시작] 메뉴 선택 (단축키 [F5])
  - [디버그] → [디버깅하지 않고 시작] 메뉴 선택 (단축키 [Ctrl + F5])



```
C:\WINDOWS\system32\cmd.exe
Hello OpenCV 3.4.0
계속하려면 아무 키나 누르십시오 . . .
```