

<부록 1> VTK 설치법

1. VTK 다운로드

VTK 홈페이지의 다운로드 페이지(<http://www.vtk.org/download/>)에서 8.0.0 버전의 "Source", "Data"를 다운로드 한다. "Source"는 VTK의 소스코드이고, "Data"는 샘플 데이터이다. "Documentation"은 HTML로 된 API 설명문서이며 온라인으로 매뉴얼 페이지(<http://www.vtk.org/doc/nightly/html/>)에 접속하여 볼 수 있다. Documentation 파일을 오프라인으로 참조할 때는 압축을 풀어서 index.html 파일을 실행하여 사용한다.

Latest Release (8.0.0)

Platform	Files
Source	VTK-8.0.0.zip
	VTK-8.0.0.tar.gz
Data	VTKData-8.0.0.zip
	VTKData-8.0.0.tar.gz
	VTKLargeData-8.0.0.zip
	VTKLargeData-8.0.0.tar.gz
Documentation	vtkDocHtml-8.0.0.tar.gz

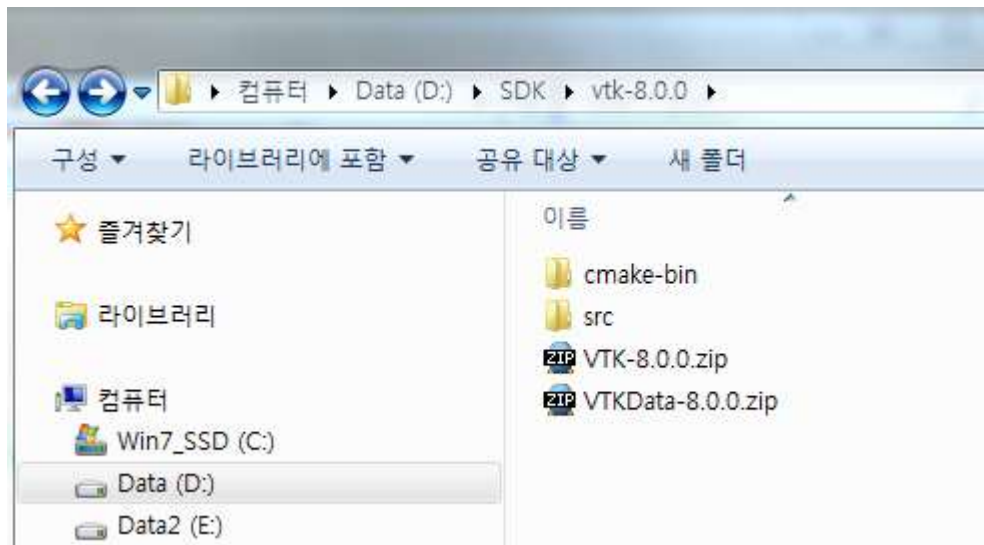
<그림 부록1.1 VTK 홈페이지 다운로드 화면>

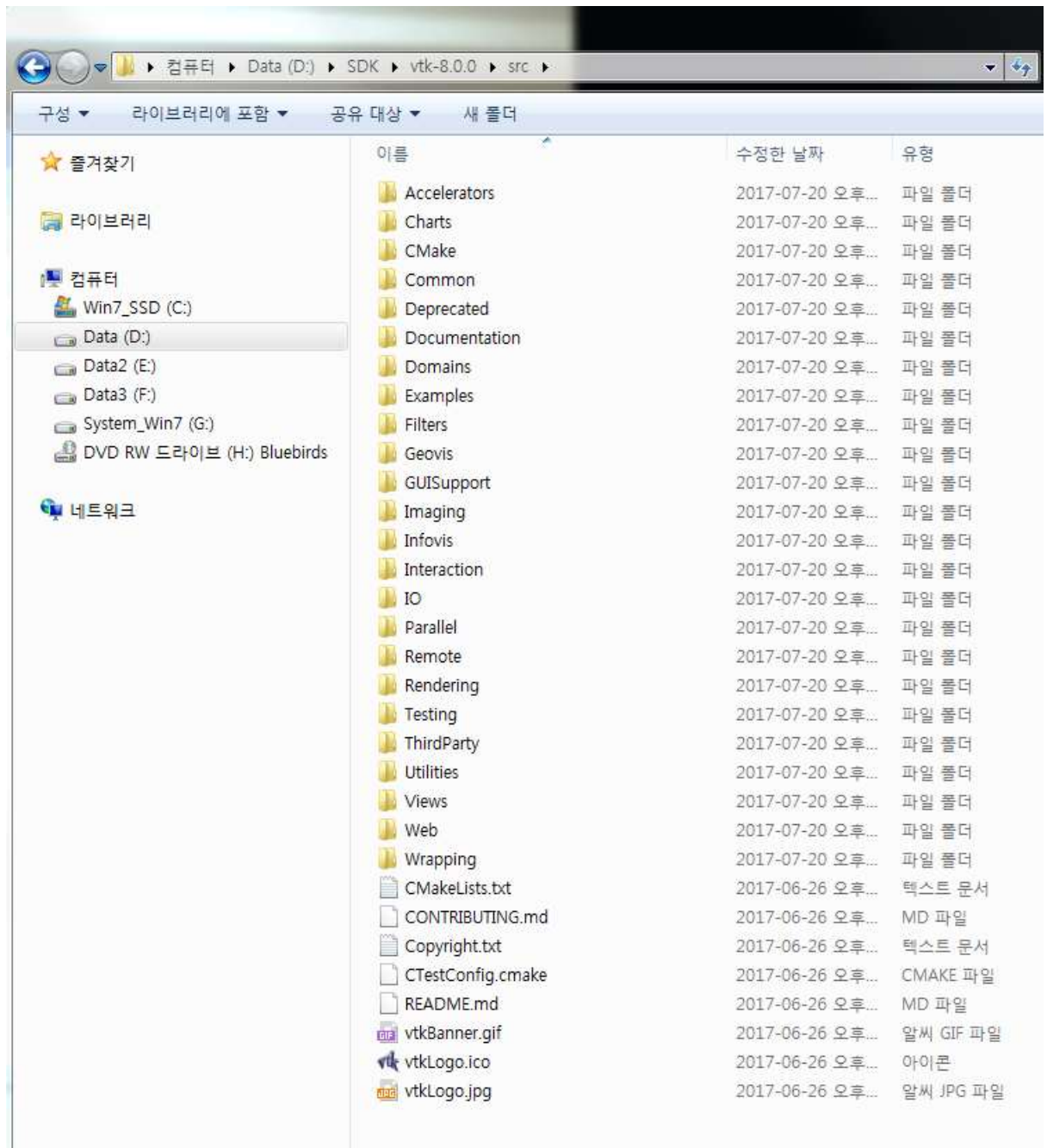
2. VTK용 폴더 생성 및 복사

(1) VTK 파일들을 모아 놓을 폴더 (예: D:\SDK\vtk-8.0.0)를 생성하고 VTK-8.0.0.zip 파일과 VTKData-8.0.0.zip 파일을 다운로드 하여 이 폴더로 복사한다. 두 개의 압축파일을 풀면 VTK-8.0.0 이라는 폴더가 생성되는데, 부모 폴더 이름과 중복되는 이름이므로 혼란을 피하기 위해 "src"라는 이름으로 수정한다. 그리고 CMake 실행 후,

해당 PC의 개발환경 (Operating System, VisualStudio 버전 등) 및 옵션에 맞추어 VTK 프로젝트가 생성될 폴더를 "cmake-bin"이라는 이름으로 생성한다.

※ 폴더 경로는 꼭 저자의 방법과 같을 필요는 없으며, 본서에서는 이후로 각각의 폴더를 ~\src, ~\cmake-bin 로 표기하기로 한다.





<그림 부록1.2 VTK 설치 폴더 경로>

3. CMake 설치 및 VTK 옵션 설정

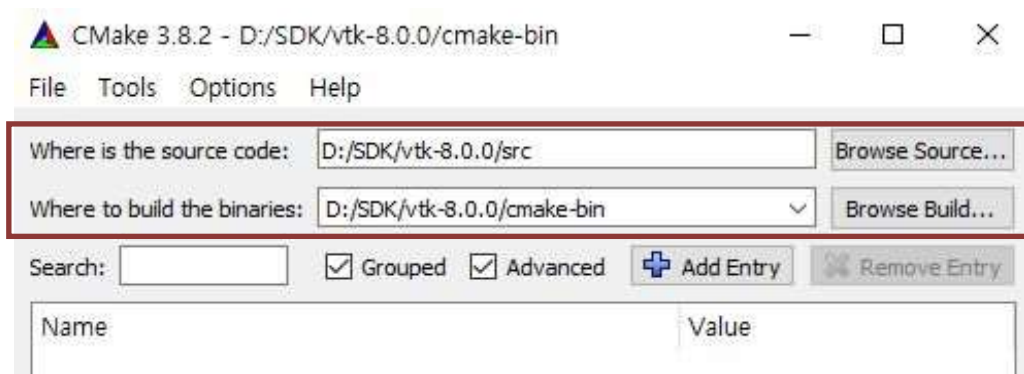


CMake (Cross Platform Make)은 여러 개발환경에 대해 각각의 플랫폼에 맞도록 open-source를 build 하여주는 프로그램이다. <http://www.cmake.org/> 에서 최신 버전을 다운로드 받아 설치한다. 본 저자는 Windows용 Installer(cmake-3.8.2-win64-x64.msi)를 사용하였다.

(1) CMake를 실행하여 상단의 source와 cmake-bin 경로를 설정한다.

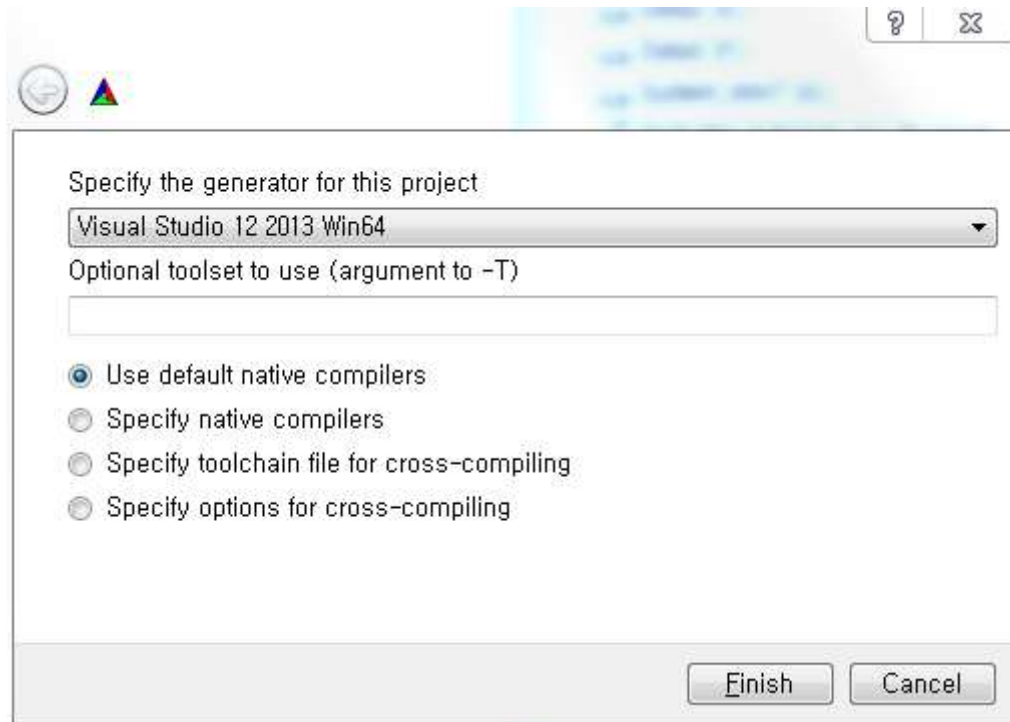
→ "Where is the source code": ~\src

→ "Where to build the binaries": ~\cmake-bin



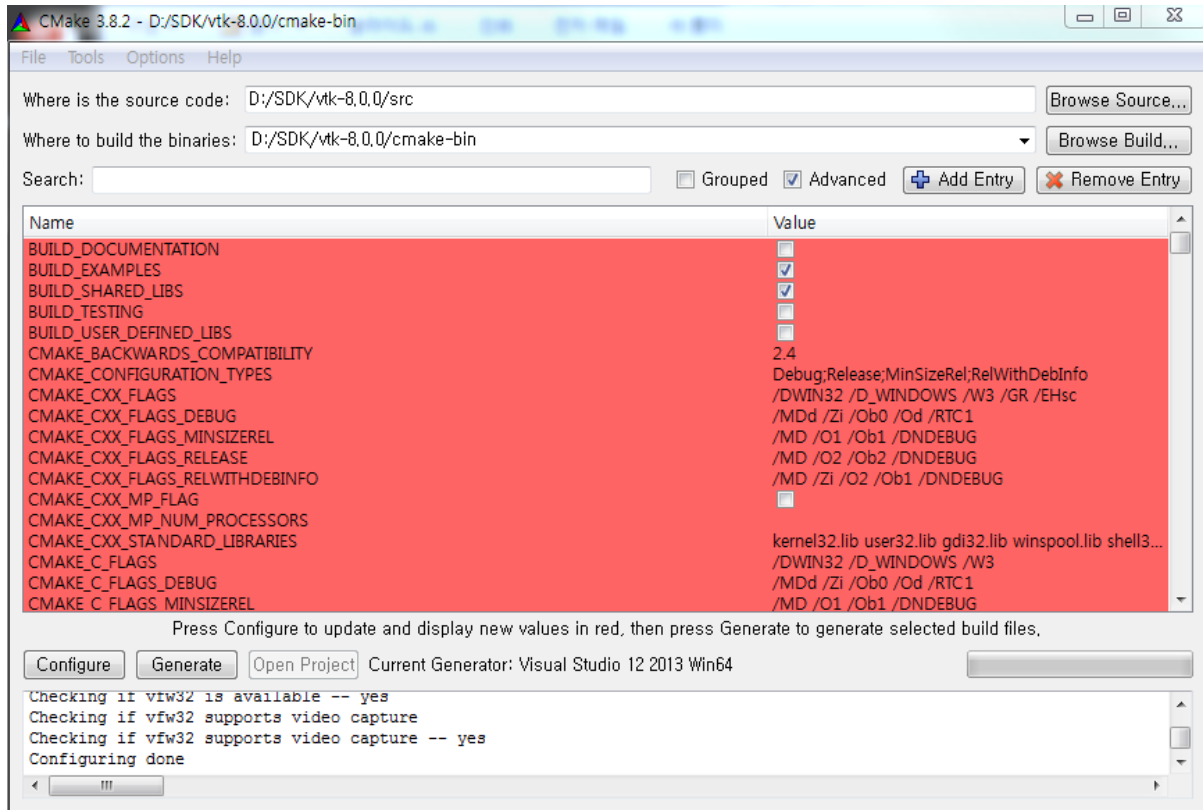
<그림 부록1.3 CMake 경로 설정>

(2) 하단의 "Configure" 버튼을 누른 후, 해당 플랫폼을 선택한다. 저자는 Windows7 64bit OS 및 VisualStudio2013을 사용하고 있으므로, 아래 그림과 같이 선택하였다. Windows 10이나 VisualStudio2015, VisualStudio2017을 사용하여도 무방하며 사용자의 개발 환경에 맞게 CMake 플랫폼 설정을 하도록 한다.



<그림 부록1.4 CMake 플랫폼 설정>

(3) 1차로 해당 코드가 생성되기를 기다린 후, 중간 부분의 "Advanced"를 선택하여 고급 옵션까지 표시한다.



<그림 부록1.5 CMake 옵션 설정>

(4) 중앙에 붉은색으로 표시된 값들 중 필요 항목을 다음과 같이 바꾸어 준 후, “Configure” 버튼을 다시 클릭한다.

“BUILD_EXAMPLES” → VTK 예제 파일들을 포함하여 빌드할지 결정하는 옵션이다. VTK

예제까지 포함하면 빌드 시간이 더 오래걸리며, 반드시 이 옵션을 체크하여 빌드할 필요는 없다. 단, 본서의 1.2절에 나와있는 “예제 코드 실행하기”를 실습해 보고자 하면 이 옵션을 체크한 후 빌드하여야 한다.

"BUILD_SHARED_LIBS" → ON (DLL 생성)

"BUILD_TESTING" → OFF

"CMAKE_INSTALL_PREFIX" → D:/SDK/vtk-8.0.0/\${BUILD_TYPE} (VTK 설치 폴더에
/\${BUILD_TYPE}를 덧붙인다. 이 명령어를 사용하면, 추후에 인스톨 시
Release / Debug와 같은 Visual Studio 구성 폴더로 자동 변경된다.)

"VTK_RENDERING_BACKEND" → OpenGL (모바일 디바이스가 아닌 PC에서 구동 시에는
OpenGL2보다 OpenGL이 호환성이 좋다.)

"Module_vtkRenderingParallel" → On (Volume Rendering 등에서 멀티 코어를 렌더링
가속화를 가능하게 한다.)

"Module_vtk~" → 위에 언급된 옵션 이외에도 추후에 필요한 모듈이 있으면
CMake에서 포함하여 다시 빌드 할 수 있다.

(5) 붉은색으로 표시된 새로운 하위 옵션이 나오면 그대로 "Configure" 버튼을 누른다.
한 옵션을 바꾸게 되면 해당 옵션 하위의 세부 옵션들이 붉은색으로 표시되어
사용자에게 세부 옵션에 대한 사용 여부를 묻게 되며, 모든 옵션들의 사용 여부가
정해지면 붉은색으로 표시된 항목 없이 Configuring이 완료가 된다. 특별히 바꿀 세부
옵션이 없으면 "Configure" 버튼을 다시 누른다.

(6) 더 이상 붉은색으로 표시된 항목 없이 Configuring이 완료되면 "Generate" 버튼을
눌러 코드를 생성한다. 즉, Configuring 작업은 사용자에게 필요한 VTK 옵션 세팅을 하는

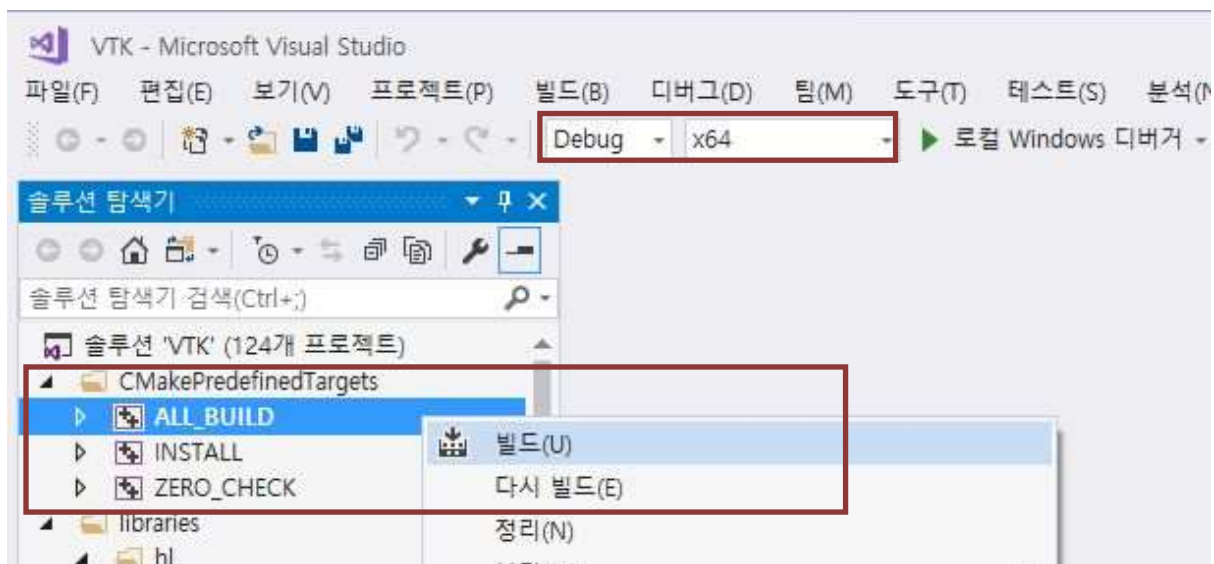
작업이며, Generate 버튼을 눌러 생성된 코드는 ~\cmake-bin 폴더에서 확인할 수 있다.

4. VTK 빌드

~\cmake-bin 폴더에 생성된 소스 파일을 빌드하는 과정이다.

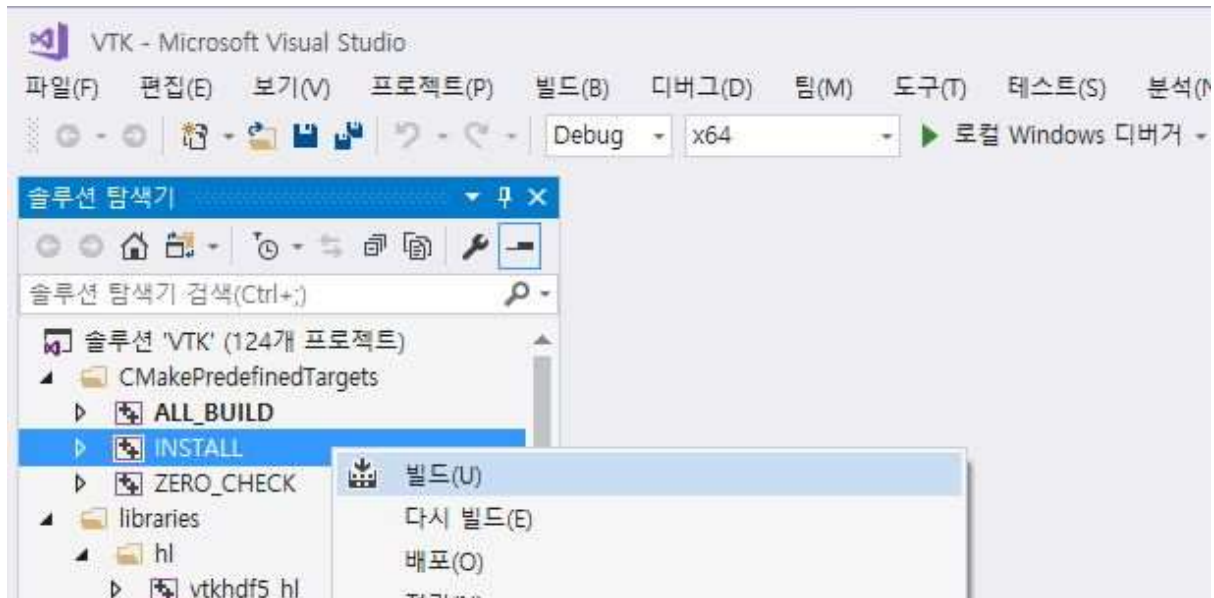
(1) ~\cmake-bin 폴더에 생성된 "VTK.sln" 파일을 더블 클릭하여 VisualStudio를 실행한다.

(2) VisualStudio에서 프로젝트 컴파일 모드를 Debug 모드로 한다. VisualStudio 좌측의 Solution Explorer에서 "ALL_BUILD"를 찾아 오른쪽 클릭해서 빌드(Build)한다. 시간이 꽤 소요되며 빌드가 완료될 때까지 기다린다.



<그림 부록1.6 VTK 소스코드 빌드>

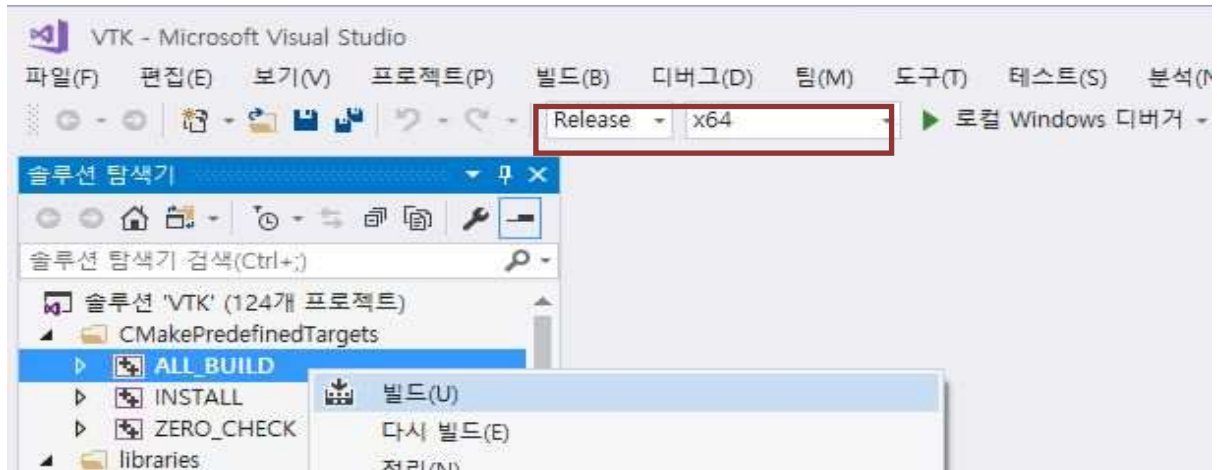
(3) 에러 없이 VTK 프로젝트의 빌드가 완료되면, 좌측의 Solution Explorer에서 "INSTALL" 프로젝트를 선택하고 오른쪽 클릭하여 빌드 한다.



<그림 부록1.7 VTK 소스코드 인스톨>

※ INSTALL 빌드 과정을 통하여 ~\cmake-bin 폴더에 build된 결과 파일들이 각각 ~\Debug 폴더 (예: D:\SDK\vtk-8.0.0\Debug) 하위의 bin, cmake, include, lib, share 에 복사된다.

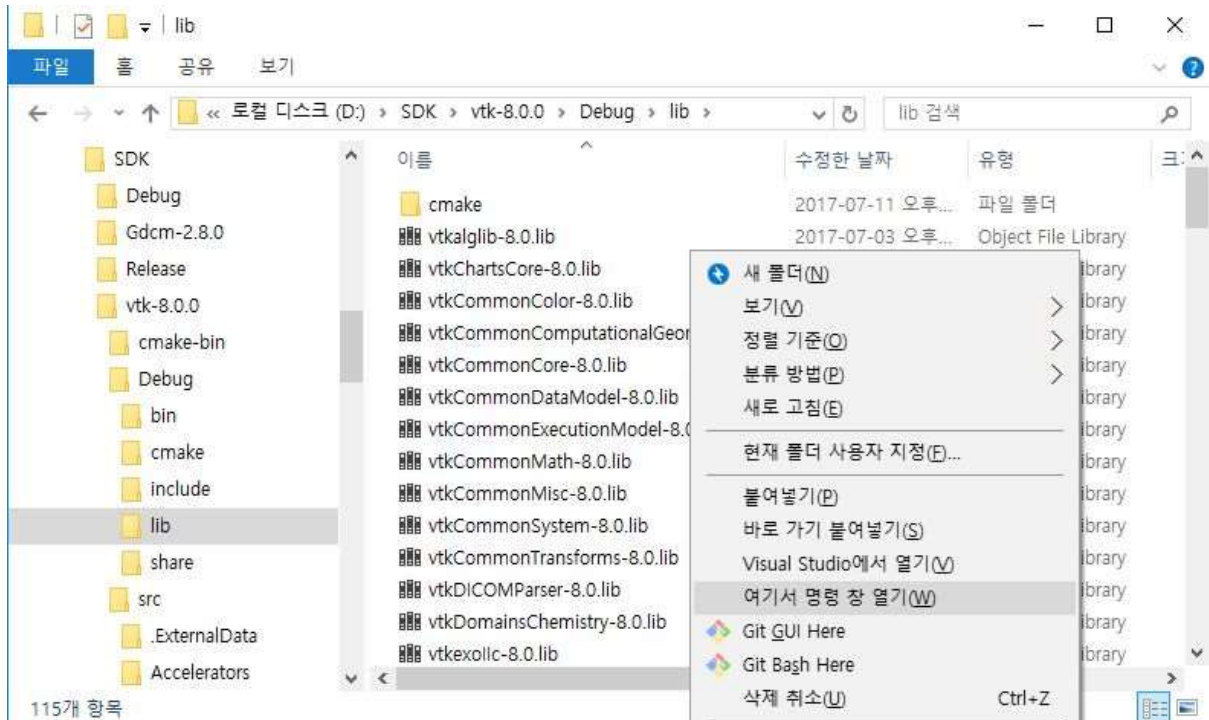
(4) VTK 프로젝트를 Release 모드로 변경하고 ALL_BUILD 및 INSTALL의 빌드를 수행하면, ~\Release 폴더 하위에 결과 파일들이 생성된다.



<그림 부록1.8 릴리즈 모드용 라이브러리를 위한 VTK 빌드>

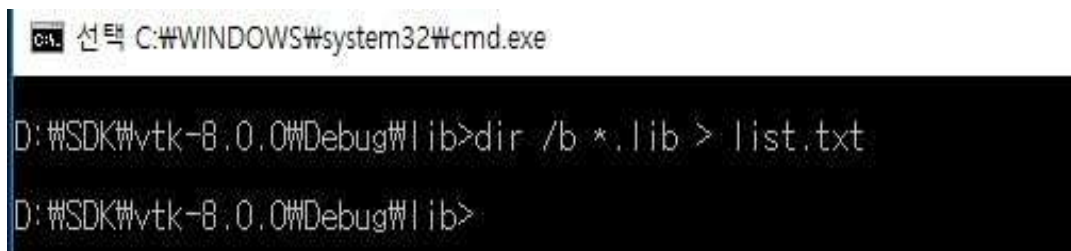
※ 만일 옵션을 변경하고 싶으면 CMake에서 옵션을 변경하여 프로젝트를 다시 빌드한다.

(5) VTK설치는 완료되었지만, 프로젝트에서 VTK 라이브러리 링크를 설정할 때 유용한 팁을 소개한다. VTK 설치 시 100개 이상의 라이브러리 파일이 생성되는데, 다음과 같은 방법을 쓰면 일일이 타이핑하여 적지 않아도 한번에 라이브러리 목록을 만들 수 있다. 윈도우 탐색기에서 ~\Debug\lib 폴더로 이동하여 파일이 아닌 빈 공간에 Shift+마우스 우클릭을 하여 "여기서 명령 창 열기"라는 메뉴를 실행한다.



<그림 부록1.9 명령 창 열기 메뉴>

명령창에 "dir /b *.lib > list.txt"라는 명령을 입력하면 모든 라이브러리 파일 목록을 포함하는 list.txt 파일이 생성된다. 이 파일을 열어서 텍스트를 복사하면, 이후 독자가 원하는 개별 프로젝트 설정 시, 손쉽게 라이브러리 목록을 프로젝트 속성에 추가할 수 있다.



<그림 부록1.10 라이브러리 목록 파일 생성 명령>