CMake GDK 예제

*이 샘플은 Microsoft 게임 개발 키트(2020년 6월)와 호환됩니다.*

# 설명

[CMake](https://cmake.org/) 플랫폼 간 빌드 시스템을 사용하여 Visual Studio 생성기를 통해 Microsoft 게임 개발 키트로 실행 파일을 빌드하는 예제입니다.



*이 샘플에서는 CMake를 사용하여 Microsoft GDK를 사용하여 빌드하는 Gaming.\*.x64 플랫폼 VC++ 프로젝트 파일을 생성하는 방법을 보여 줍니다.*  *생성기를 통해 CMake를 활용하는 다른 방법은* ***CMakeExample****을 참조하세요.*

# 샘플 빌드(Visual Studio)

Visual Studio 2019 또는 2022를 사용하여 새 프로젝트 대화 상자 또는 “파일 -> 열기 -> 폴더...” 메뉴 명령에서 “로컬 폴더 열기…”를 선택하고 샘플 폴더를 엽니다.

* 이렇게 하려면 “Windows 구성 요소용 C++ CMake 도구”(Microsoft.VisualStudio.Component.VC.CMake.Project)가 설치되어 있어야 합니다.

필요한 경우 **XdkEditionTarget** 변수(CMakePresets.json 또는 gxdk\_toolchain.cmake / gxdk\_xs\_toolchain.cmake)를 편집하여 참조된 올바른 GDK 버전이 있는지 확인합니다.

CMake 도구를 열면 캐시가 자동으로 생성되어야 합니다. 그렇지 않으면 CMakeList.txt를 선택하고 오른쪽 단추 메뉴에서 “캐시 생성”을 선택합니다. 그런 다음 “빌드 -> 모두 다시 빌드” 메뉴 명령을 사용합니다.

콤보 상자에서 빌드할 플랫폼을 선택합니다([CMake 사전 설정 통합](https://devblogs.microsoft.com/cppblog/cmake-presets-integration-in-visual-studio-and-visual-studio-code/)을 지원하는 VS 2019(16.10) 이상을 사용하는 경우 다음과 같이 채워짐).

Graphical user interface, text, application

Description automatically generated

Visual Studio의 CMake에 대한 자세한 내용은 [Microsoft Docs](https://docs.microsoft.com/en-us/cpp/build/cmake-projects-in-visual-studio)를 참조하세요.

* Visual Studio 2022를 사용하는 경우 CMakePresets.json을 편집하여 다음에서 이 줄을 변경합니다.

"generator": "Visual Studio 16 2019",

to:

"generator": "Visual Studio 17 2022",

# 샘플 빌드(명령줄)

*VS x64 Native 개발자 명령 프롬프트*를 사용하여 명령줄에서 생성하고 빌드할 수도 있습니다.

cd CMakeGDKExample

cmake . -B out -DXdkEditionTarget=220300 -DCMAKE\_TOOLCHAIN\_FILE=.\gxdk\_toolchain.cmake

cmake --build out

CMake 사전 설정도 제공됩니다(CMake 3.19 이상 필요).

cmake --list-presets

cmake --preset=x64-XboxOne

cmake --build out\build\x64-XboxOne

# 샘플 사용(Visual Studio)

Visual Studio의 다른 인스턴스에서 생성된 SLN/VCXPROJ를 엽니다.

CMakeGDKExample\out\build\x64-XboxOne\CMakeGDKExample.sln

CMake 3.17 이전 버전을 사용하는 경우 먼저 구성 관리자 사용하여 *CMakeGDKExample* 프로젝트에 대한 “배포” 확인란을 선택합니다.

그런 다음 F5 키를 사용하여 배포/실행합니다.

*원래 CMakeLists.txt 컨텍스트에서 F5 키를 누르면 느슨한 레이아웃이 'bin' 디렉터리 내부에 배치되지 않으므로 시작되지 않습니다. Gaming.Xbox.\*.x64 구성의 경우 원격 콘솔이 아닌 개발 PC에서도 실행하려고 시도합니다.*

# 샘플 사용(명령줄)

샘플을 배포하려면 *Xbox 게임 명령 프롬프트* 인스턴스를 열고 샘플 디렉터리로 변경합니다.

cd CMakeGDKExample\out\build\x64-XboxOne\bin\Gaming.Xbox.XboxOne.x64

데스크톱의 경우 느슨한 레이아웃은 bin\Gaming.Desktop.x64\Debug에 있습니다.

### 푸시 배포

푸시를 수행하려면 ‘느슨한’ 레이아웃을 배포합니다.

xbapp deploy Layout\Image\Loose

### Run-from-PC

PC에서 ‘느슨한’ 레이아웃을 실행하려면 다음을 수행합니다.

xbapp launch Layout\Image\Loose\CMakeGDKExample.exe

### 패키지된 배포

Layout\Image\Loose\Microsoft.Config를 편집하여 TargetDeviceFamily 요소(“PC”, “Scarlett” 또는 ”XboxOne”)를 추가합니다.

<ExecutableList>

<Executable Name="CMakeGDKExample.exe"

**TargetDeviceFamily="XboxOne"**

Id="Game" />

</ExecutableList>

패키지를 만들려면 다음을 수행합니다.

makepkg genmap /f chunks.xml /d Layout\Image\Loose

makepkg pack /f chunks.xml /lt /d Layout\Image\Loose /pd .

그런 다음 결과 패키지를 콘솔에 설치합니다(정확한 .xvc 파일 이름은 다를 수 있음).

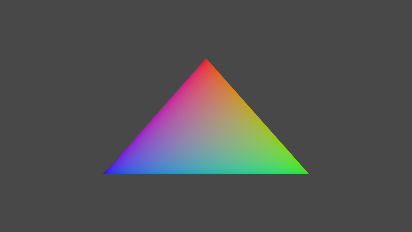
xbapp install CMakeGDKExample\_1.0.0.0\_neutral\_\_8wekyb3d8bbwe\_x.xvc

데스크톱 패키징의 경우:

makepkg genmap /f chunks.xml /d bin\Gaming.Desktop.x64\Debug

makepkg pack /pc /f chunks.xml /lt /d bin\Gaming.Desktop.x64\Debug /pd .

샘플을 실행하면 디바이스 및 swapchain이 만들어지고 색이 지정된 삼각형이 그려집니다. 컨트롤이나 다른 동작은 없습니다.



*Xbox Series X|S 및/또는 Xbox 릴리스 버전을 패키징하려면 플랫폼 및 구성에 적합한 디렉터리로 변경합니다.*

# 구현 세부 정보

**CMakeExample**에서는 “Ninja” 생성기를 사용하므로 Microsoft GDK MSBuild 규칙을 사용하지 않습니다. 이 버전에서는 “Visual Studio 16 2019 Win64” 생성기를 대신 사용하므로 Microsoft GDK MSBuild 지침을 사용합니다.

CMake 생성에서는 명령줄로 전달된 도구 체인 파일을 사용합니다.

|  |  |
| --- | --- |
| Gaming.Desktop.x64 | -DCMAKE\_TOOLCHAIN\_FILE="grdk\_toolchain.cmake" |
| Gaming.Xbox.XboxOne.x64 | -DCMAKE\_TOOLCHAIN\_FILE="gxdk\_toolchain.cmake" |
| Gaming.Xbox.Scarlett.x64 | -DCMAKE\_TOOLCHAIN\_FILE="gxdk\_xs\_toolchain.cmake" |

세 파일은 모두 사용자 지정 MSBuild 속성 파일인 gdk\_build.props를 사용합니다.

GDK에서 Gaming.\*.x64 MSBuild 규칙을 사용하면 MicrosoftGame.Config 지역화가 처리되어 CRT 파일을 레이아웃 등에 배치합니다.

CMake는 셰이더에 FXCCompile MSBuild 대상을 활용할 수 없으므로 CMakeLists.txt에서는 DXC를 사용자 지정 대상으로 실행합니다. 도구 체인은 셰이더 컴파일의 올바른 버전을 찾는 역할을 합니다. 따라서 gxdk\_toolchain.cmake 및 gxdk\_xs\_toolchain.cmake에는 **XdkTargetEdition** 변수가 필요합니다.

결과 CMake를 사용하여 *Build without Install*(BWOI)를 지원하려면 (a) 빌드 중인 플랫폼에 대해 적절한 DXC.EXE를 가리키도록 **GDK\_DXCTool**을 시적으로 설정하고 (b) BWOIExample CMake 생성 vcxproj는 Microsoft GDK에 대한 MSBuild 규칙을 사용하므로 **BWOIExample** 샘플에 자세히 설명된 Directory.Build.props 솔루션을 사용해야 합니다. CMake를 *생성*하고 결과 SLN/VCXPROJ를 빌드할 때 Directory.Build.props 파일이 있고 환경이 올바르게 설정되어야 합니다.

BWOI를 사용하여 명령줄에서 생성할 때 -DGDK\_DXCTool=<path>를 추가하여 **GDK\_DXCTool**을 지정할 수 있습니다. 여기서 <path>는 <path to GDK>\<edition number>\GXDK\bin\<XboxOne or Scarlett>\dxc.exe 형식으로 사용됩니다. 예제:

-DGDK\_DXCTool="d:\xtrctd.sdks\BWOIExample\Microsoft GDK\210600\GXDK\bin\XboxOne\dxc.exe".

## 병렬 도구 집합

[Visual C++ 블로그](https://devblogs.microsoft.com/cppblog/side-by-side-minor-version-msvc-toolsets-in-visual-studio-2019/)에 따라 최신 버전의 Visual Studio IDE와 이전 버전의 컴파일러 도구 집합을 함께 사용할 수 있습니다. CMake의 경우 **CMakeSettings.json**을 통해 이 작업을 수행합니다. 예를 들어 VS 2019(16.0) 버전의 컴파일러를 사용하려면 다음을 추가합니다.

"environment":

[

{

"ClearDevCommandPromptEnvVars": "false",

"VCToolsVersion": "14.20.27508"

}

],

Visual Studio 통합을 사용하지 않는 동안 CMake 및 VS 생성기를 직접 사용하는 경우 **set\_property**를 통해 이를 지정할 수도 있습니다.

set\_property(TARGET ${PROJECT\_NAME} PROPERTY VS\_GLOBAL\_ClearDevCommandPromptEnvVars "false")

set\_property(TARGET ${PROJECT\_NAME} PROPERTY VS\_GLOBAL\_VCToolsVersion "14.20.27508")

# 버전 기록

|  |  |
| --- | --- |
| **2020년 4월** | 이 버전 CMake 샘플의 초기 릴리스 |
| **2020년 6월** | 2006 GDK FAL 릴리스에 대해 업데이트되었습니다. |
| **2020년 8월** | 명시적으로 설정된 VS 시작 프로젝트를 추가했습니다.  병렬 세부 정보로 업데이트되었습니다. |
| **2020년 11월** | 확장 라이브러리에서 .lib 파일을 명시적으로 참조해야 합니다.  xmem.lib 및 xg\_\*.lib를 Gaming.Xbox.\*.x64 링크에 추가했습니다. |
| **2021년 2월** | Gaming.Desktop.x64 배포를 수정했습니다. |
| **2021년 4월** | Gaming.Desktop.x64의 링크 문제를 해결하기 위해 appnotify.lib를 추가했습니다.  LargeLogo.png를 추가했습니다. |
| **2021년 6월** | 버그를 수정하여 병렬 시나리오에 대해 생성된 vcxproj에서 버전 번호를 명시적으로 설정했습니다.  일반 코드를 정리했습니다. |
| **2021년 8월** | 도구 체인 파일을 개선했습니다.  BWOI 시나리오에 대한 참고 사항을 추가했습니다. |
| **2021년 10월** | BWOI 시나리오에 대한 참고 사항을 추가했습니다. |
| **2022년 1월** | VS 2022 지원이 추가되었습니다.  CMake를 정리하고 사전 설정 파일을 추가했습니다. |