Cmake Xbox 콘솔 앱

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

# 설명

이는 Microsoft GDK를 사용하여 Xbox 하드웨어에서 실행할 수 있는 “Win32 콘솔” 응용 프로그램을 생성하는 [CMake](https://cmake.org/) 플랫폼 간 빌드 시스템을 사용하는 예제입니다. ‘printf’ 스타일 출력을 사용하는 비그래픽 개발자 단위 테스트에 적합합니다.



*표준 Microsoft GDK 응용 프로그램을 빌드하기 위해 CMake를 사용하는 방법에 대한 자세한 내용은* ***CMakeExample*** *및* ***CMakeGDKExample****을 참조하세요.*

# 샘플 빌드(Visual Studio)

Visual Studio 2019를 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 CMake **XdkEditionTarget** 변수(CMakePresets.json 또는 CMakeList.txt)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. 빌드 제품은 “**out**” 하위 폴더에 있습니다.

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

*이 샘플에서는 CMake 3.13 이상이 필요하도록 target\_link\_directories를 사용합니다. Visual Studio 2017(15.9 업데이트)에는 버전 3.12가 포함되어 있으므로 Visual Studio 2019에 대한 지침이 제공됩니다. 물론 Visual Studio 통합에 의존하는 대신 CMake 도구를 직접 사용할 수 있습니다. Visual Studio 2017을 사용하는 경우 VC 런타임 DLL을 찾기 위해 CMakeList.txt에서 논리를 수정해야 합니다.*

기본 설정에는 CMake 사전 설정으로 정의된 **x64-Debug**, **x64-Release**, **x64-Clang-Debug** 및 **x64-Clang-Release** 구성이 포함됩니다.

* 이렇게 하려면 “Windows 구성 요소용 C++ Clang 컴파일러”가 설치되어 있어야 합니다.

*F5 키를 누르면 원격 콘솔이 아닌 개발 PC에서 실행하려고 시도하므로 실패할 수도 실패하지 않을 수도 있습니다. 올바르게 실행하려면 아래 지침에 따라 프로그램을 배포해야 합니다.*

# 샘플 빌드(명령줄)

*VS 2019 개발자 명령 프롬프트*를 사용하여 명령줄에서 생성하고 빌드할 수도 있습니다.

cd CMakeXboxConsoleApp

cmake -B out -DXdkEditionTarget=220300

cmake --build out

CMake 사전 설정도 제공됩니다.

cmake --list-presets

cmake --preset=x64-Debug

cmake --build out\build\x64-Debug

# 샘플 사용

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

cd CMakeXboxConsoleApp\out\build\<config>

xbcp bin\Console\\*.exe xd:\

xbcp bin\Console\\*.dll xd:\

샘플을 실행하려면

xbrun /O D:\CMakeXboxConsoleApp.exe

프로그램은 시스템 운영 체제의 컨텍스트에서 실행됩니다.

대신 게임 운영 체제의 컨텍스트에서 실행하려는 경우 비슷한 프로시저를 사용할 수 있습니다. 먼저 대상 콘솔에서 게임 운영 체제 타이틀을 실행합니다. Visual Studio 새 프로젝트 대화 상자를 사용하고 Microsoft GDK로 기본 “Direct3D 12 Xbox 게임 프로젝트”를 만들면 좋습니다. 빌드 및 배포하고 실행 중인 상태로 둡니다.

그리고 다음을 사용합니다.

xbcp /x/title bin\Console\\*.exe xd:\

xbcp /x/title bin\Console\\*.dll xd:\

샘플을 실행하려면

xbrun /x/title /O D:\CMakeXboxConsoleApp.exe

이 작업은 게임 운영 체제 VM에 프로세스를 삽입하여 작동한다는 것을 기억하세요. 다중 프로세스 게임 타이틀은 현재 지원되지 않으며, 그래픽, 오디오 및 GameRuntime을 비롯한 여러 구성 요소는 여러 프로세스 시나리오에 대해 테스트와 지원이 이루어지지 않습니다. 또한 ‘호스팅’ 파티틀을 간단하게 유지하여 CPU 리소스 사용을 제한하는 것이 좋습니다.

# 구현 세부 정보

PC 데스크톱의 경우 Win32 콘솔 exe(예: /SUBSYSTEM:CONSOLE)에 대한 **CMakeLists.txt**는 대략 다음과 같습니다.

cmake\_minimum\_required(버전 3.13)

project(CMakeExampleWindowsConsole LANGUAGES CXX)

option(BUILD\_USING\_LTCG "Enable Whole Program Optimization" ON)

set(CMAKE\_CXX\_STANDARD 14)

set(CMAKE\_CXX\_STANDARD\_REQUIRED ON)

set(CMAKE\_CXX\_EXTENSIONS OFF)

add\_executable(${PROJECT\_NAME} Main.cpp)

target\_compile\_definitions(${PROJECT\_NAME} PRIVATE "$<$<CONFIG:DEBUG>:\_DEBUG>" "$<$<CONFIG:RELEASE>:NDEBUG>")

target\_compile\_definitions(${PROJECT\_NAME} PRIVATE \_CONSOLE \_UNICODE UNICODE)

# 경고 수준 4 사용

string(REPLACE "/W3 " "/W4 " CMAKE\_CXX\_FLAGS ${CMAKE\_CXX\_FLAGS})

string(REPLACE "/W3 " "/W4 " CMAKE\_CXX\_FLAGS\_DEBUG ${CMAKE\_CXX\_FLAGS\_DEBUG})

string(REPLACE "/W3 " "/W4 " CMAKE\_CXX\_FLAGS\_RELEASE ${CMAKE\_CXX\_FLAGS\_RELEASE})

# typeid 또는 dynamic\_cast를 사용하지 않는 경우 RTTI를 사용하지 않도록 설정하여 이진 크기를 줄일 수 있습니다.

string(REPLACE "/GR " "/GR- " CMAKE\_CXX\_FLAGS ${CMAKE\_CXX\_FLAGS})

string(REPLACE "/GR " "/GR- " CMAKE\_CXX\_FLAGS\_DEBUG ${CMAKE\_CXX\_FLAGS\_DEBUG})

string(REPLACE "/GR " "/GR- " CMAKE\_CXX\_FLAGS\_RELEASE ${CMAKE\_CXX\_FLAGS\_RELEASE})

target\_compile\_options(${PROJECT\_NAME} PRIVATE /fp:fast /GS /Gy)

if(CMAKE\_CXX\_COMPILER\_ID MATCHES "MSVC")

target\_compile\_options(${PROJECT\_NAME} PRIVATE /permissive- /Zc:\_\_cplusplus)

if(CMAKE\_BUILD\_TYPE MATCHES "Debug")

elseif(BUILD\_USING\_LTCG MATCHES ON)

target\_compile\_options(${PROJECT\_NAME} PRIVATE /GL /Gw)

target\_link\_options(${PROJECT\_NAME} PRIVATE /IGNORE:4075 /LTCG)

endif()

endif()

Xbox 하드웨어의 시스템 및 게임 운영 체제의 경우 서로 다른 링크 라이브러리 집합을 사용하고 지원되지 않는 라이브러리를 선택하지 않도록 해야 합니다. 또한 지원되지 않는 API를 사용하지 않도록 적절한 API 분할을 사용하도록 설정해야 하며, 이 샘플에서는 플랫폼 헤더 및 라이브러리를 사용하여 빌드되도록 합니다.

Xbox 하드웨어에서 실행되는 응용 프로그램은 필요한 Visual C++ 런타임 DLL뿐만 아니라 디버그용으로 빌드된 경우 ucrtbased.lib도 제공해야 합니다.

이 샘플의 Xbox “콘솔” CMake는 Xbox Series X|S 또는 Xbox One 하드웨어 중 하나에서 실행되도록 EXE를 빌드하게 설정됩니다. 콘솔 앱에 Direct3D를 사용할 수 없어서 플랫폼의 주요 API 차이를 피하므로, 두 플랫폼에서 동일한 EXE가 실행될 것으로 예상할 수 있습니다. 또한 CMakeLists.txt에서 설정되지 않은 특정 XboxOne 및 Scarlett include/lib 경로에 의해서도 마찬가지입니다.

원하는 경우 Xbox Series X|S 하드웨어를 대상으로 하는 추가 컴파일러 CPU를 사용하도록 설정할 수 있습니다. 이 작업은 빌드 옵션 OPTIMIZE\_FOR\_SCARLETT을 ON으로 설정하여 수행됩니다. 그 결과인 EXE는 Xbox Series X|S에서 이전처럼 실행되지만 Xbox One에서는 실행되지 않습니다. 이를 보여주기 위해 샘플에서는 관련 CPUID 검사를 수행하는 DirectXMath XMVerifyCPUSupport 함수를 사용합니다.

# 추가 정보

이 예제에서 사용되는 모든 컴파일러와 링커 스위치에 관한 자세한 내용은 **CMakeExample**를 참조하세요.

이 샘플의 CMake 프로젝트는 BWOI(Build With/Out 설치)를 사용하는 옵트인 빌드 옵션을 지원합니다. 사용하도록 설정된 경우 *BWOIExample*의 extractgdk.cmd 스크립트에서 만든 추출된 Microsoft GDK를 가리키는 ExtractedFolder 환경 변수가 필요합니다. 필요에 따라 2020년 5월 GDK 이상에 대해 추출된 Windows 10 SDK(19041)를 가질 수도 있습니다. CMake 프로젝트는 Gaming.\*.x64 MSBuild 플랫폼을 사용하지 않으므로 vctargets.cmd 스크립트의 결과가 필요하지 않습니다.

이 빌드 옵션을 사용하도록 설정하려면 CMakeSettings.json을 사용하여 BUILD\_USING\_BWOI를 True로 설정합니다. 또는 명령줄을 사용하여 빌드하는 경우 생성 단계에 -DBUILD\_USING\_BWOI=True를 추가합니다.

자세한 내용은 **BWOIExample**을 참조하세요.

# 알려진 문제

clang/LLVM 도구 집합을 사용하는 경우 DirectXMath 3.14를 포함하는 Windows 10 SDK(19041)를 사용하고 있는지 확인합니다. DirectXMath 3.13 이전 버전에서는 XMVerifyCPUSupport 구현이 해당 도구 집합에 대해 올바르게 빌드되지 않습니다. 자세한 내용은 <https://walbourn.github.io/directxmath-3.14/>를 참조하세요.

# 버전 기록

|  |  |
| --- | --- |
| **2020년 5월** | 초기 버전. |
| **2020년 6월** | 2020년 6월 GDK FAL 릴리스에 대해 업데이트되었습니다. |
| **2020년 11월** | CMake 파일을 정리하고 \_CONSOLE 정의를 추가했습니다. |
| **2021년 2월** | 사소한 CMake 정리입니다. |
| **2021년 8월** | 도구 체인 파일의 향상된 기능입니다. |
| **2021년 10월** | BWOI에 대한 업데이트입니다. |
| **2022년 1월** | VS 2022 지원이 추가되었습니다.  CMake 정리 및 CMake 사전 설정 파일이 추가되었습니다. |