NVIDIA VIDEO CODEC SDK

Read Me
Table of Contents

Chapter 1. Read Me..............................................................................................................1
  1.1. System Requirements.................................................................................................1
  1.2. Building Samples........................................................................................................3
Chapter 1. Read Me

1.1. System Requirements

- NVIDIA Maxwell/Pascal/Volta/Turing/Ampere GPU with hardware video accelerators
  - Video Codec SDK is available in GitLab at [https://gitlab.com/nvidia/video/video-codec-sdk](https://gitlab.com/nvidia/video/video-codec-sdk)
  - Documents can be browsed online at [https://docs.nvidia.com/video-technologies/video-codec-sdk/index.html](https://docs.nvidia.com/video-technologies/video-codec-sdk/index.html)
  - Windows: Driver version 471.41 or higher
  - Linux: Driver version 470.57.02 or higher
  - CUDA 11.0 or higher Toolkit
  - Visual Studio Solution and Linux Makefiles can now be generated using CMake. CMake 3.9 or later is required for SDK 10.0 and higher. Self-extracting scripts or installers for CMake can be downloaded from [https://cmake.org/download/](https://cmake.org/download/).

> **Note:** NVIDIA Video Codec SDK is now supported on IBM Power9 class server with NVIDIA Tesla V100 (SXM2) GPU.

Windows Configuration Requirements

- DirectX SDK is needed. You can download the latest SDK from Microsoft’s DirectX website.
- The Vulkan SDK needs to be installed in order to build and run the AppMotionEstimationVkCuda sample application.
- In order to build and run AppEncD3D12 sample application, Windows 20H1 or later is required. Visual Studio 2017 and above should be used for building and running this application.
In Windows, the following environment variables must be set to build the sample applications included with the SDK

- DXSDK_DIR: pointing to the DirectX SDK root directory.
- VULKAN_SDK: pointing to Vulkan SDK install directory.
- The CUDA Toolkit and the related environment variables are optional to install if the client has Video Codec SDK 8.0. However, they are mandatory if client has Video Codec SDK 8.1 or above on his/her machine.

Plus all the requirements under System Requirements and Common to all OS platforms

Linux Configuration Requirements

- X11 and OpenGL, GLUT, GLEW libraries for video playback and display
- CUDA Toolkit is mandatory if client has Video Codec SDK 8.1 or above on his/her machine.
- Libraries and headers from the FFmpeg project which can be downloaded and installed using the distribution’s package manager or compiled from source.
- The sample applications have been compiled and tested against the libraries and headers from FFmpeg- 4.4. The source code of FFmpeg- 4.4 has been included in this SDK package. While configuring FFmpeg on Linux, it is recommended not to use ‘disable-decoders’ option. This configuration is known to have a channel error (XID 31) while executing sample applications with certain clips and/or result in an unexpected behavior.
- To build/use sample applications that depend on FFmpeg, users may need to
  - Add the directory [/usr/local/lib/pkgconfig by default] to the PKG_CONFIG_PATH environment variable. This is required by the Makefile to determine the include paths for the FFmpeg headers.
  - Add the directory where the FFmpeg libraries are installed to the LD_LIBRARY_PATH environment variable. This is required for resolving runtime dependencies on FFmpeg libraries.
- Stub libraries (libnvcuvid.so and libnvidia-encode.so) have been included as part of the SDK package, in order to aid development of applications on systems where the NVIDIA driver has not been installed. The sample applications in the SDK will link against these stub libraries as part of the build process. However, users need to ensure that the stub libraries are not referenced when running the sample applications. A driver compatible with this SDK needs to be installed in order for the sample applications to work correctly.
- The Vulkan SDK needs to be installed in order to build and run the AppMotionEstimationVkCuda sample application.
- Plus all the requirements under System Requirements and Common to all OS platforms

Windows Subsystem for Linux (WSL) Configuration Requirements

- CUDA Toolkit is mandatory if client has Video Codec SDK 8.1 or above on his/her machine.
- Libraries and headers from the FFmpeg project which can be downloaded and installed using the distribution’s package manager or compiled from source.

- The sample applications have been compiled and tested against the libraries and headers from FFmpeg- 4.4. The source code of FFmpeg- 4.4 has been included in this SDK package. While configuring FFmpeg on WSL, it is recommended not to use ‘disable-decoders’ option. This configuration is known to have a channel error (XID 31) while executing sample applications with certain clips and/or result in an unexpected behavior.

- To build/use sample applications that depend on FFmpeg, users may need to
  - Add the directory (/usr/local/lib/pkgconfig by default) to the PKG_CONFIG_PATH environment variable. This is required by the Makefile to determine the include paths for the FFmpeg headers.
  - Add the directory where the FFmpeg libraries are installed to the LD_LIBRARY_PATH environment variable. This is required for resolving runtime dependencies on FFmpeg libraries.

- Plus all the requirements under System Requirements and Common to all OS platforms

Common to all OS platforms

- Vulkan SDK can be downloaded from [https://vulkan.lunarg.com/sdk/home](https://vulkan.lunarg.com/sdk/home). Alternatively, it can be installed by using the distribution’s package manager.

1.2. Building Samples

Video Codec SDK uses CMake for building the samples. To build the samples, follow these steps:

**Windows:**

1. Install all dependencies for Windows, as specified in [Windows Configuration Requirements](#).
2. Extract the contents of the SDK into a folder.
3. Create a subfolder named “build” in Video_Codec_SDK_x.y.z/Samples
4. Open a command prompt in the “build” folder and run the following command, depending upon the version of Visual Studio on your computer.

This command will generate the necessary Visual Studio project files in the “build” folder. You can open NvCodec.sln file in Visual Studio and build. Alternatively, following command can be used to build the solution:
cmake --build . --target install --config Release

The application binaries will be available in Samples/build. Please note that the applications are validated only for x64 platform.

**Linux:**

1. Install all dependencies for Linux, as specified in [Linux Configuration Requirements](#).
2. Extract the contents of the SDK into a folder.
3. Create a subfolder named “build” in Video_Codec_SDK_x.y.z/Samples
4. Use the following command to build samples in release mode.
   
   ▶ cmake -DCMAKE_BUILD_TYPE=Release ..
   ▶ make
   ▶ make install

This will build and install the binaries of the sample applications. The application binaries will be available in Samples/build.

**Windows Subsystem for Linux:**

1. Install all dependencies for Windows Subsystem for Linux, as specified in [Windows Subsystem for Linux (WSL) Configuration Requirements](#).
2. Follow the build and installation steps provided above for Linux. Applications using OpenGL and Vulkan will not be built.