NVIDIA VIDEO CODEC SDK

v10.0 | June 2020

Read Me

TABLE OF CONTENTS

Chapter 1. READ ME

1.1. System Requirements

- NVIDIA Kepler/Maxwell/Pascal/Volta/Turing/GA100 GPU with hardware video accelerators
    - Refer to the NVIDIA Video SDK developer zone web page (https://developer.nvidia.com/nvidia-video-codec-sdk) for GPUs which support video encoding and decoding acceleration.
- Video Codec SDK. It can be downloaded from https://developer.nvidia.com/nvidia-video-codec-sdk
- Windows: Driver version 445.87 or higher
- Linux: Driver version 450.51 or higher
- CUDA 10.1 or higher Toolkit
    - CUDA 11.0 or higher is needed if GA100 GPU is used
- Visual Studio Solution and Linux Makefiles can now be generated using CMake. CMake 3.9 or later is required for SDK 10.0. Self-extracting scripts or installers for CMake can be downloaded from https://cmake.org/download/.

> 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 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- 4.1. The source code of FFmpeg- 4.1 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

Common to all OS platforms

- CUDA toolkit can be downloaded from http://developer.nvidia.com/cuda/cuda-toolkit
- Vulkan SDK can be downloaded from 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.

    - Visual Studio 2019: `cmake -G"Visual Studio 16 2019" -A"x64" -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=. ..`
    - Visual Studio 2017: `cmake -G"Visual Studio 15 2017" -A"x64" -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=. ..`
    - Visual Studio 2015: `cmake -G"Visual Studio 14 2015" -A"x64" -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=. ..`

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.

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.