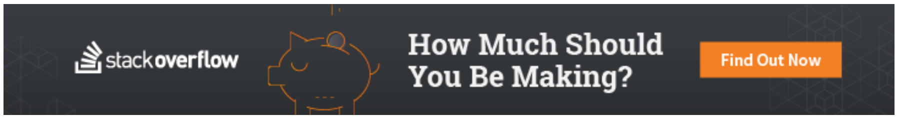
How to choose OpenCL implementation with CMake?



I have a Windows OpenCL application that uses some of the AMD extensions. Additionally, my application has some optional CUDA components. When building the visual studio solution with CMake, the FindOpenCL module picks the Nvidia OpenCL implementation instead of AMD's. Is there an easy way to make CMake use the AMD version?

I tried commenting the Nvidia sections in the FindOpenCL module code, but that did not work. Is there some registry setting I can modify? I installed CUDA after installing AMD's SDK. Would the installation order make a difference?

cmake opencl

edited Jul 5 at 21:57

asked Jul 5 at 19:06
Damian
39 6

query platform name, if it has amd in it, select it. or you can try renaming/moving nvidia's icd file. – huseyin tugrul buyukisik Jul 5 at 22:01

Do you mean query platform name on CMake? or query it with OpenCL code? If it's the latter, that is different from my use case since I need the AMD implementation at compile time due to the extensions – Damian Jul 5 at 22:52

I meant in OpenCL code. Maybe you can temporarily remove icd file location from environment variables such as PATH? – huseyin tugrul buyukisik Jul 6 at 1:33

Which set of include files you use is *independent* of which platform and device you select at runtime. For example, I always use an old AMD SDK for my headers, but the code runs on AMD, NVIDIA, and Intel. In other words, you don't select the implementation at compile time. – Dithermaster Jul 7 at 21:52

To select a specific platform & device at runtime, just search for their known name (or substring like "NVIDIA"). Or, as HTB suggests above, you can manually remove items from the ICD (in the Registry, if on Windows; HLM\SOFTWARE\Khronos\OpenCL\Vendors), but that is only if it's your own machine (don't programmatically do that to a user's machine!). – Dithermaster Jul 7 at 21:54

1 Answer

In the end, what worked was to remove the NVIDIA environment variables from the `PATHS` suggestions in the `find_library` and `find_path` commands and adding the `NO_DEFAULT_PATH` . The issue was that, even though I removed the suggestions, CMake was adding the default path which included NVIDIA's OpenCL implementation. Both commands ended up like shown below:

```
find_path(OpenCL_INCLUDE_DIR
  NAMES
    CL/cl.h OpenCL/cl.h
    NO_DEFAULT_PATH
  PATHS
    ENV "PROGRAMFILES(X86)"
    ENV AMDAPPSDKROOT
    ENV ATISTREAMSDKROOT
  PATH_SUFFIXES
    include
    OpenCL/common/inc
```