CUVILib

- Image Transforms
- Feature Detectors
- Motion Estimation
- Color Operations
- Filters

How to Run CUDA In Visual Studio 2010

By Salman Ul Haq on February 24th, 2011

Microsoft has given a complete facelift to the new Visual Studio 2010. A whole bunch of new features and removal of a couple of them. With the new VS 2010, you cannot define custom build rules with .rules file as it was done before. Now there's a whole new bunch of modules that need to be specified in order to make a set of custom build rules to work with Visual Studio 2010 project. This blog explains the steps involved in making CUDA work on the new Visual Studio 2010.

I have used the following for this guide:

- **Visual Studio 2010 running on Windows 7 x64**
- **NVIDIA CUDA Toolkit 3.2** (Download from here)
- **NSight Host and Monitor 1.5** (Download from here)
- **NVIDIA CUDA SDK 3.2** (Download from here)

And yes, you need to have Visual Studio 2008 installed or at least you should have VC90 C compiler on your machine with the appropriate Windows SDK. So this is what I have in addition to the above:

- **Visual Studio 2008**
- **Windows SDK v6.0A**

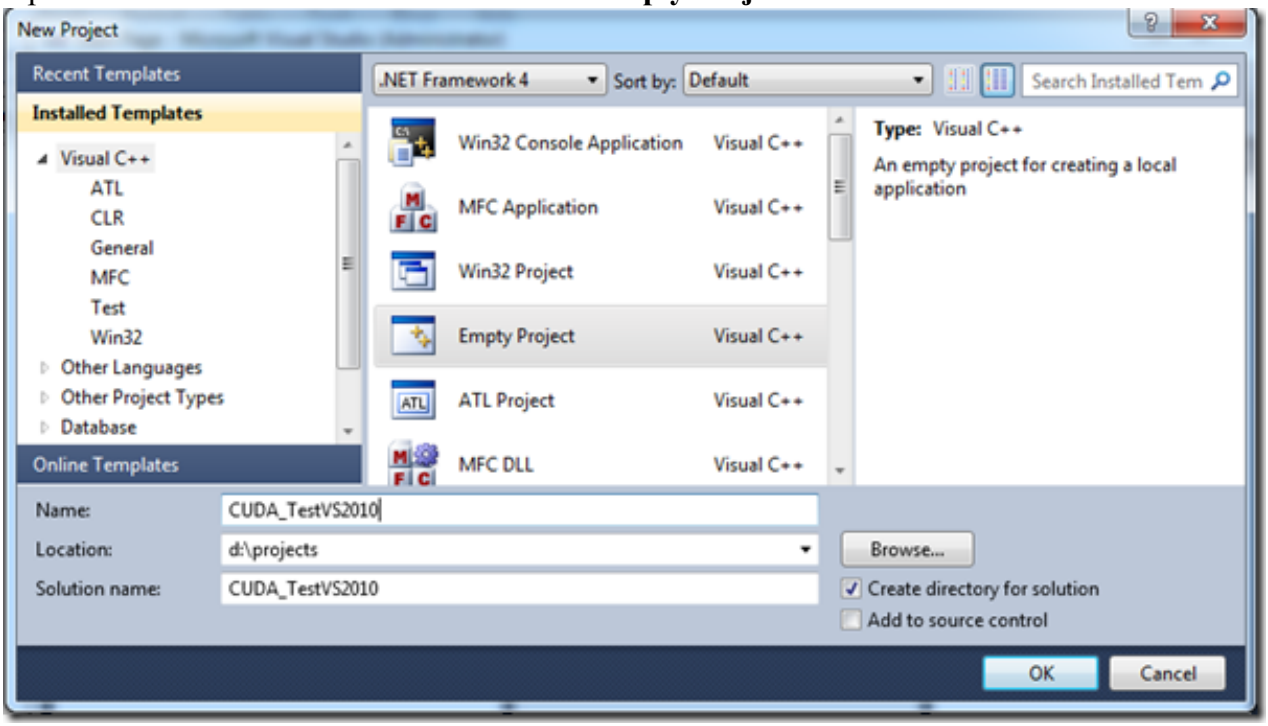
The latest CUDA Toolkit 3.2 does not has support for the VS100 C compiler and hence the reason why you still need to have Visual Studio 2008 installed on your machine. Also make sure you have the right Windows SDK (or

at least anything below Windows SDK v7.0 which is automatically installed with VS 2010). Now we are ready to get started.

Now the reason why you need to have NSight installed. Without NSight, you will have to manually specify the CUDA properties and target files so in order to avoid that, better download and install NSight (link). Make sure you enable NSight for both VS 2008 and VS 2010. Now we're ready to roll!

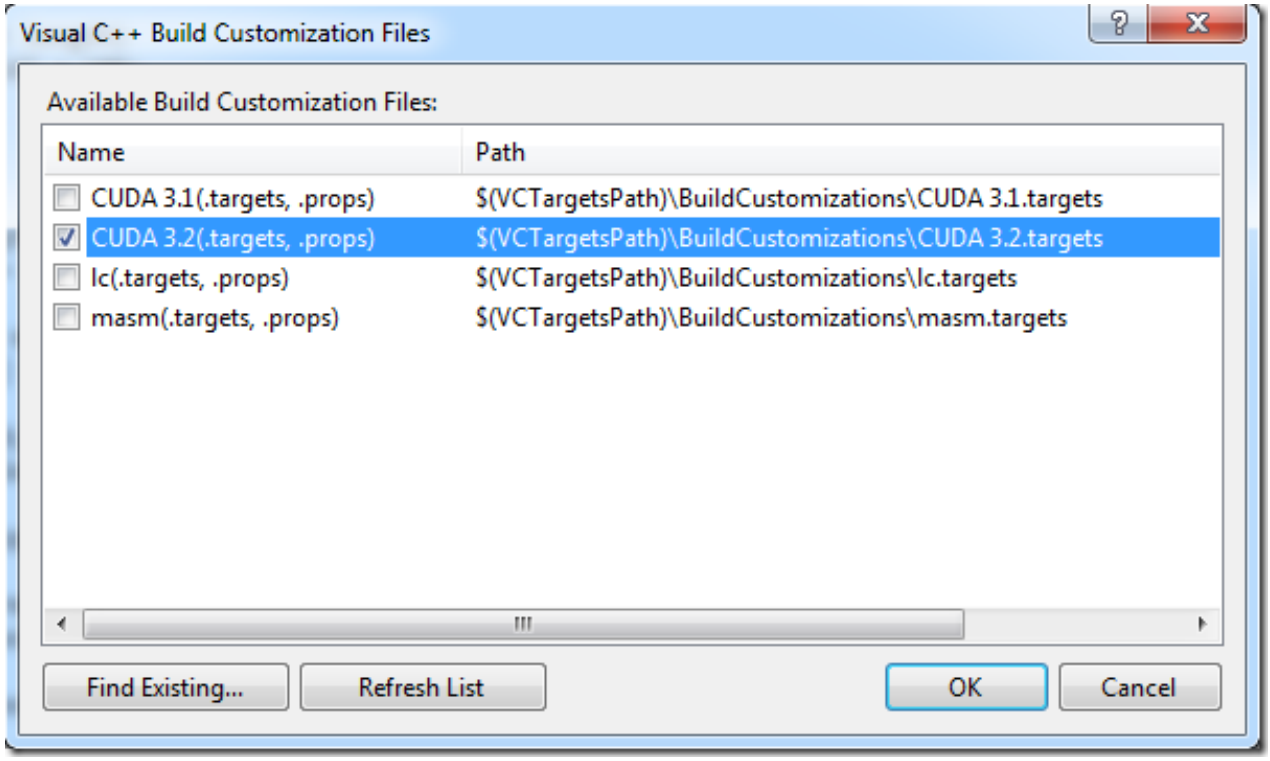
Step 1: Creating an Empty Project

Open VS 2010 and create a new **Visual C++ Empty Project:**



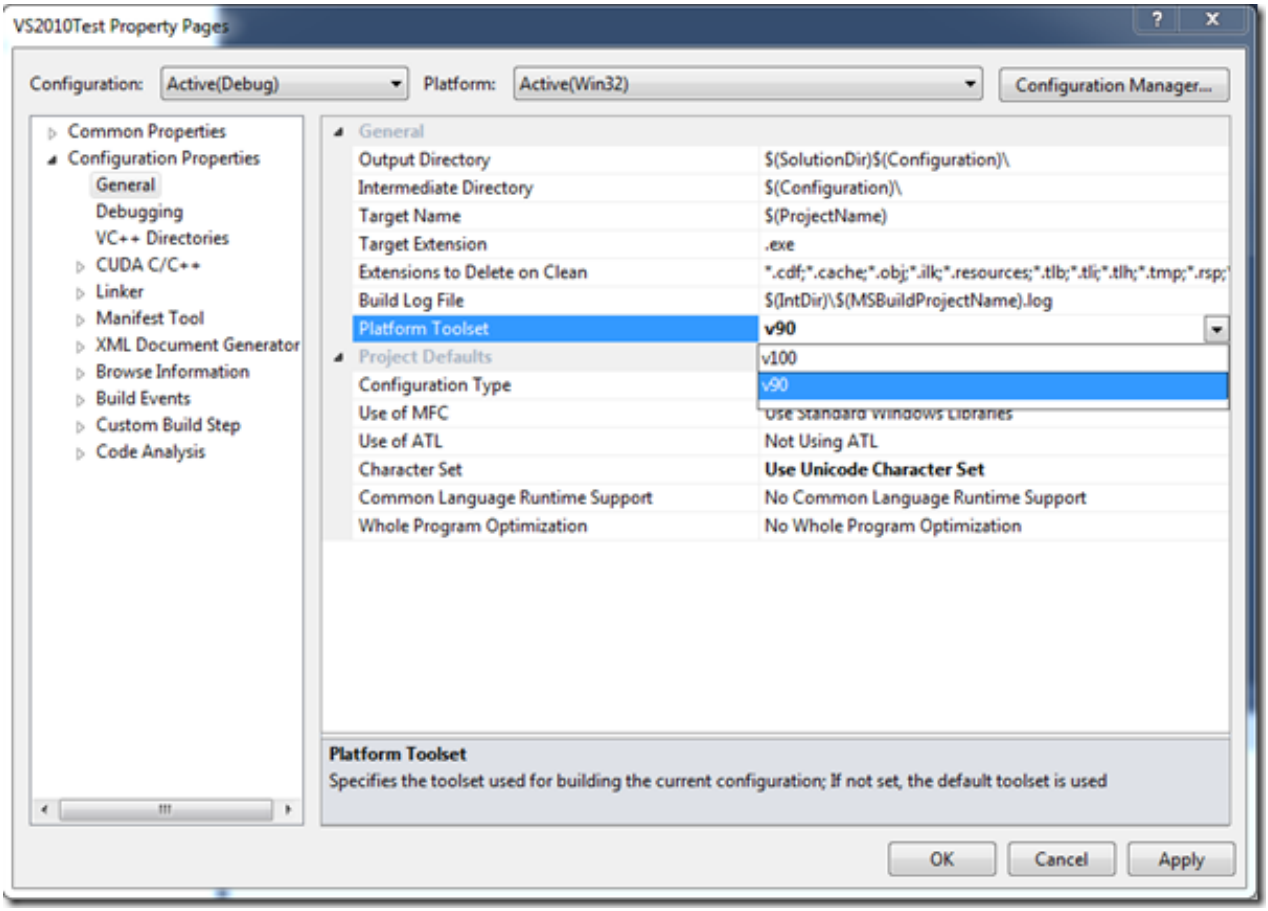
Step 2: Specify Build Customization and Add CUDA Source File

Now go to the **Project's Properties –> Build Customizations.** If you have installed CUDA Toolkit 3.2 and NSight 1.5 then you should see the following option:



I will only add a simple .cu CUDA source file which has a simple kernel that adds a constant number to an integer vector. I have the main function written in the same file that sets up the kernel launch parameters and everything in-between 😀 *I have given the download link of the complete CUDA Visual Studio 2010 template solution at the end of this blog.*

Now go to **Project –> Properties –> General** and change Platform Toolset from V100 to v90. This will specify that you are using v90 C compiler for this project. If you do not have VS 2008 on your machine then you won't even see this option so make sure you have it installed:



Now you are all set to Build your project! Just tune in the right setting of your CUDA application from **Projects –> CUDA C/C++** (Host and Device settings) and you're all set to go.

Common Issues:

In case you cannot find the CUDA 3.2 Target and Props files, look here:
C:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\BuildCustomizations

Note: I will keep this section updated with more commonly faced issues and their solutions. If you're facing an issue, post it in the comments and I will respond.

That's about it. I have uploaded the Sample VS 2010 Solution file which you can download from **here**. You can use it as a template for your CUDA projects