

在 Visual Studio 中使用 CUDA

CUDA 的主要工具是 `nvcc`，它會執行所需要的程式，將 CUDA 程式碼編譯成執行檔 (或 object 檔)。在 Visual Studio 下，我們透過設定 custom build tool 的方式，讓 Visual Studio 會自動執行 `nvcc`。

這裡以 Visual Studio 2005 為例：

1. 首先，建立一個 Win32 Console 模式的 project (在 Application Settings 中記得勾選 Empty project)，並新增一個檔案，例如 `main.cu`。

2. 在 `main.cu` 上按右鍵，並選擇 Properties。點選 General，確定 Tool 的部份是選擇 Custom Build Tool。

3. 選擇 Custom Build Step，在 Command Line 使用以下設定：

* Release 模式：`"$(CUDA_BIN_PATH)\nvcc.exe" -ccbin "$(VCInstallDir)bin" -c -DWIN32 -D_CONSOLE -D_MBCS -Xcompiler /EHsc,/W3,/nologo,/Wp64,/O2,/Zi,/MT -I"$(CUDA_INC_PATH)" -o $(ConfigurationName)\$(InputName).obj $(InputFileName)`

* Debug 模式：`"$(CUDA_BIN_PATH)\nvcc.exe" -ccbin "$(VCInstallDir)bin" -c -D_DEBUG -DWIN32 -D_CONSOLE -D_MBCS -Xcompiler /EHsc,/W3,/nologo,/Wp64,/Od,/Zi,/RTC1,/MTd -I"$(CUDA_INC_PATH)" -o $(ConfigurationName)\$(InputName).obj $(InputFileName)`

4. 如果想要使用軟體模擬的模式，可以新增兩個額外的設定：

* EmuRelease 模式：`"$(CUDA_BIN_PATH)\nvcc.exe" -ccbin "$(VCInstallDir)bin" -deviceemu -c -DWIN32 -D_CONSOLE -D_MBCS -Xcompiler /EHsc,/W3,/nologo,/Wp64,/O2,/Zi,/MT -I"$(CUDA_INC_PATH)" -o $(ConfigurationName)\$(InputName).obj $(InputFileName)`

* EmuDebug 模式：`"$(CUDA_BIN_PATH)\nvcc.exe" -ccbin "$(VCInstallDir)bin" -deviceemu -c -D_DEBUG -DWIN32 -D_CONSOLE -D_MBCS -Xcompiler /EHsc,/W3,/nologo,/Wp64,/Od,/Zi,/RTC1,/MTd -I"$(CUDA_INC_PATH)" -o $(ConfigurationName)\$(InputName).obj $(InputFileName)`

5. 對所有的設定檔，在 Custom Build Step 的 Outputs 中加入 `$(ConfigurationName)\$(InputName).obj`。

6. 選擇 project，按右鍵選擇 Properties，再點選 Linker。對所有的設定檔修改以下設定：

- * General/Enable Incremental Linking：No
- * General/Additional Library Directories：\$(CUDA_LIB_PATH)
- * Input/Additional Dependencies：cudart.lib

這樣應該就可以直接在 Visual Studio 的 IDE 中，編輯 CUDA 程式後，直接 build 以及執行程式了。