# CUDA安装

## CUDA环境变量设置

Cuda8.0安装完成之后，需要配置7个环境变量。其中CUDA自动配置了2个系统变量，所以手动配置5个即可。自动配置的两个环境变量分别是：

CUDA\_PATH：C:\Program Files\NVIDIA GPU ComputingToolkit\CUDA\v8.0

CUDA\_PATH\_V8\_0：C:\Program Files\NVIDIA GPU ComputingToolkit\CUDA\v8.0

需要手动配置的5个环境变量分别是：

CUDA\_BIN\_PATH：%CUDA\_PATH%\bin

CUDA\_LIB\_PATH：%CUDA\_PATH%\lib\Win32

CUDA\_SDK\_BIN：%CUDA\_SDK\_PATH%\bin\Win64

CUDA\_SDK\_LIB：%CUDA\_SDK\_PATH%\common\lib\x64

CUDA\_SDK\_PATH：C:\ProgramData\NVIDIA Corporation\CUDASamples\v8.0

以上操作完成后，测试一下是否成功，在cmd中输入：nvcc -V，如下图所示：

在环境变量配置成功后，记得要将电脑重启！！！

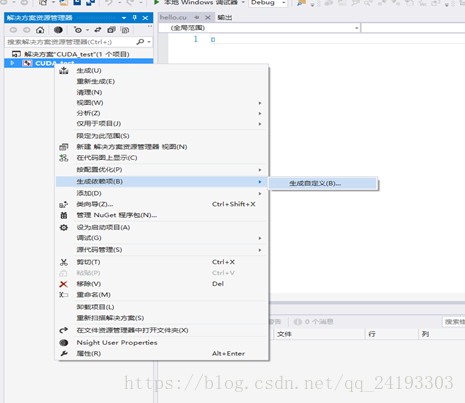
## CUDA在VS中的设置

3、接下来是在VS2013中建立CUDA工程。首先新建项目，选择Visual C++，选择win32控制台，填写名称与位置，最后确定。

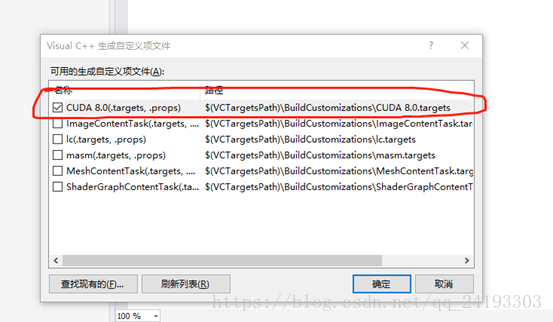
右键“源文件”，选择“添加”，选择“新建项”；

或者，添加溢已有的源代码。

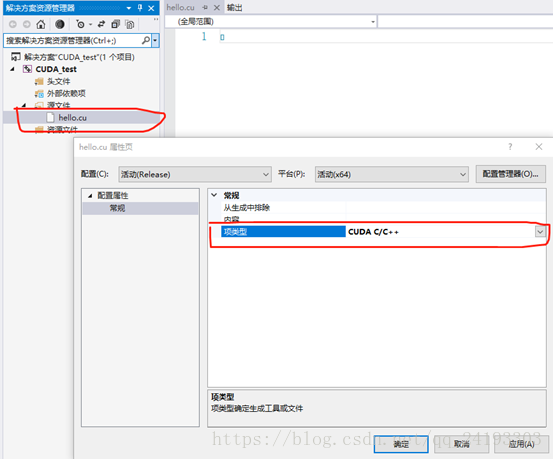
右键工程》生成依赖项》生成自定义



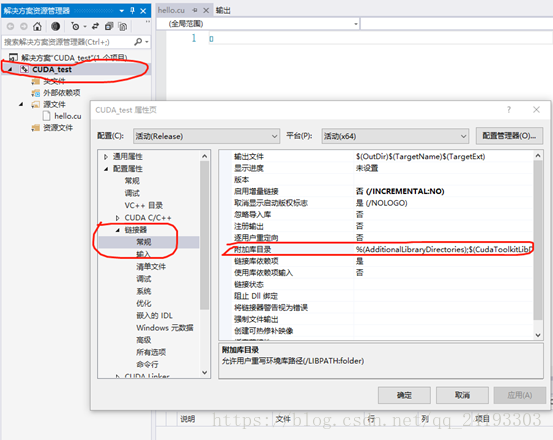
选择CUDA8.0



右键hello.cu文件，属性》常规》项类型》CUDAC/C++

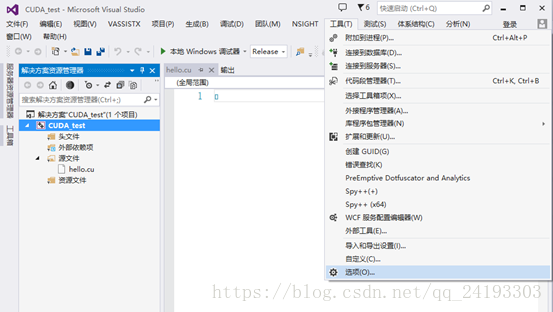


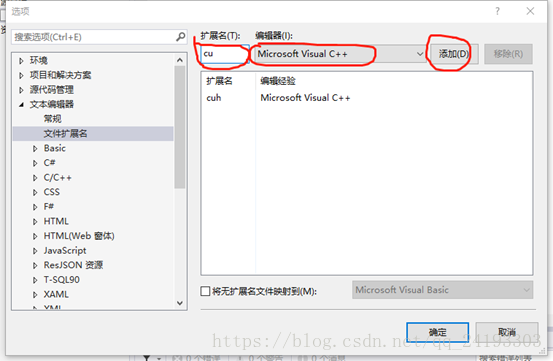
右键项目，属性>链接器>常规》附加库目录，添加$(CUDA\_PATH\_V8\_0)\lib\$(Platform);

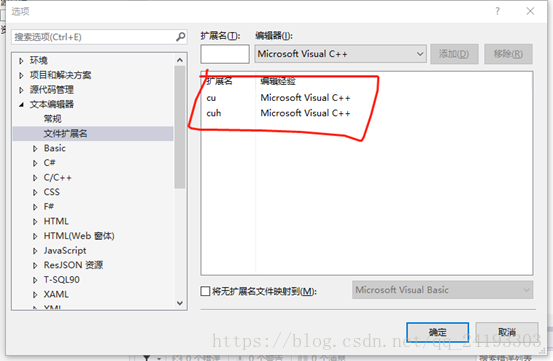


在链接器》输入》附加依赖项：添加cudart.lib。

选择工具》选项》文本编辑器》文件拓展名：添加cu和cuh两个文件拓展名







**5、最重要的错误：进阶篇，实现VS中直接新建CUDA解决方案的功能（因为离成功就差一步，所以很重要，这么重要还错了真不应该啊）**  
将“CudaProjectVsWizards”文件整体拷贝到“D:\ProgramFiles\Microsoft Visual Studio 14.0\Common7\IDE\Extensions”下，这里要注意两点，  
第一，VS安装的时候在C盘和D盘都会创建一个“Microsoft Visual Studio 14.0”文件夹，一开始我想把“CudaProjectVsWizards”拷贝到C盘的哪个“Microsoft Visual Studio 14.0”文件夹中，但是找不到“Extensions”，其实“Extensions”在D盘，就是你最开始安装vs的那个盘

第二，并不是拷贝过去重启VS就可以的，详情看：https://stackoverflow.com/questions/50042186/cannot-install-cuda-toolkit-9-1-85-on-windows-7-64-bit，解决方案：在路径 D:\ProgramFiles\Microsoft Visual Studio 14.0\Common7\IDE下，打开cmd（管理员身份）: “devenv /setup”

至此，编译环境的相关搭建就完成了 。在hello.cu文件中添加如下测试代码，并运行。

#include<stdio.h>

#include"cuda\_runtime.h"

#include"device\_launch\_parameters.h"

boolInitCUDA()

{

       intcount;

       cudaGetDeviceCount(&count);

       if(count == 0){

              fprintf(stderr, "Thereis no device.\n");

              returnfalse;

       }

       inti;

       for(i = 0; i < count; i++){

              cudaDevicePropprop;

              if(cudaGetDeviceProperties(&prop, i) == cudaSuccess)

              {

                     if(prop.major >= 1)

                     {

                            break;

                     }

              }

       }

       if(i == count){

              fprintf(stderr, "Thereis no device supporting CUDA 1.x.\n");

              returnfalse;

       }

       cudaSetDevice(i);

       returntrue;

}

intmain()

{

       if(!InitCUDA()){

              return0;

       }

       printf("HelloWorld,CUDA has been initialized.\n");

       system("pause");

       return0;

}