Cg（C for Graphcis）语言，是 NVIDIA 与 Microsoft 合作研发，旨在为开发 人员提供一套方便、跨平台（良好的兼容性），控制可编程图形硬件的高级语言。 Cg 语言的语法结构与 C 语言非常类似，使用 Cg 编写的着色程序默认的文件后 缀是*. Cg。

静态编译（static coompilation）：一旦编译为可执行文件，在可执行文 件运行期间不再需要源码信息。例如C或者 C++ 编写的程序， 需要首先编译成可执行文件（.exe 文件），然后才能在 GPU 上运行，且一旦编译后，除非改变程序代码，否则不需要重新编译。
动态编译（dynamic compilation）：编译程序和源码都要参与到程序的运行过程中。

NVIDIA的网页上下载Cg Toolkit：
http://developer.nvidia.com/object/cg_toolkit.html

双击Cg-3.1_April2012_Setup.exe

readme.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

NVIDIA Cg 3.1 April 2012 README Copyright (C) 2002-2012 NVIDIA Corp.

==========

This distribution contains

- NVIDIA Cg toolkit documentation
  in the docs directory

- NVIDIA Cg compiler (cgc)
  in the bin directory

- NVIDIA Cg runtime libraries
  in the lib directory

- Example Cg applications
  in the examples directory

- Under Microsoft Windows, a Cg language syntax highlighter
  for Microsoft Visual Studio is provided in the
  msdev_syntax_highlighting directory

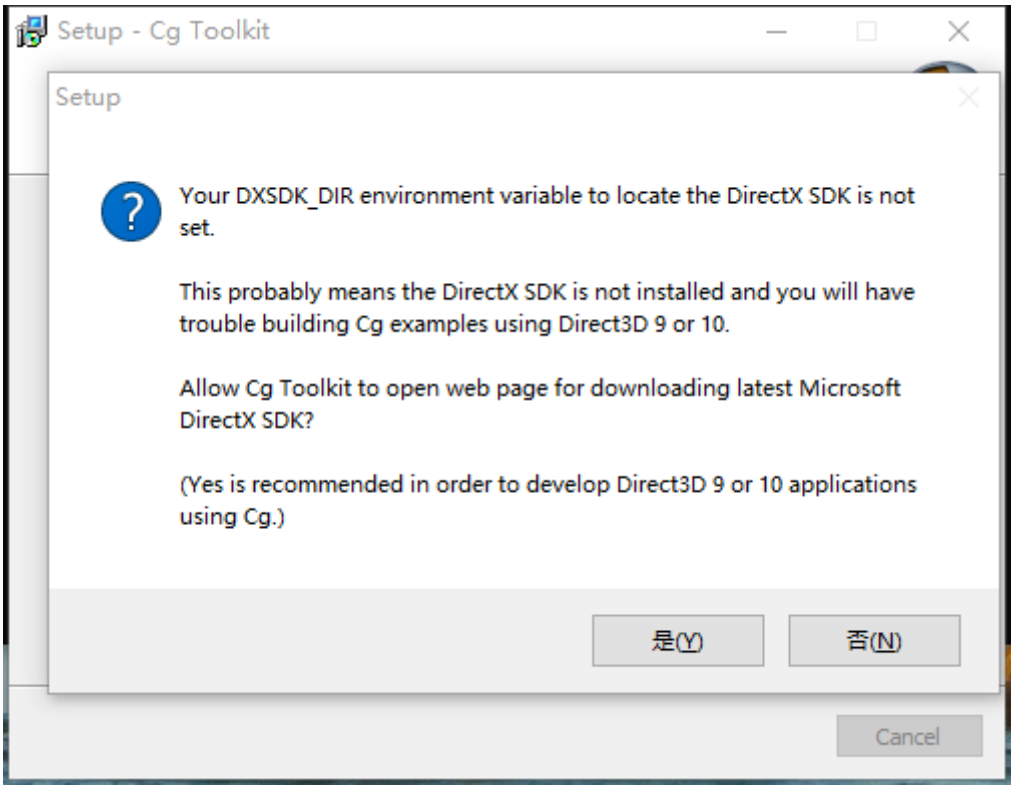
- Under Microsoft Windows, if selected at install time, 64-bit
  binaries and libraries are in the bin.x64 and lib.x64 directories.

See the release notes (docs/CgReleaseNotes.pdf) for detailed
information about this release.

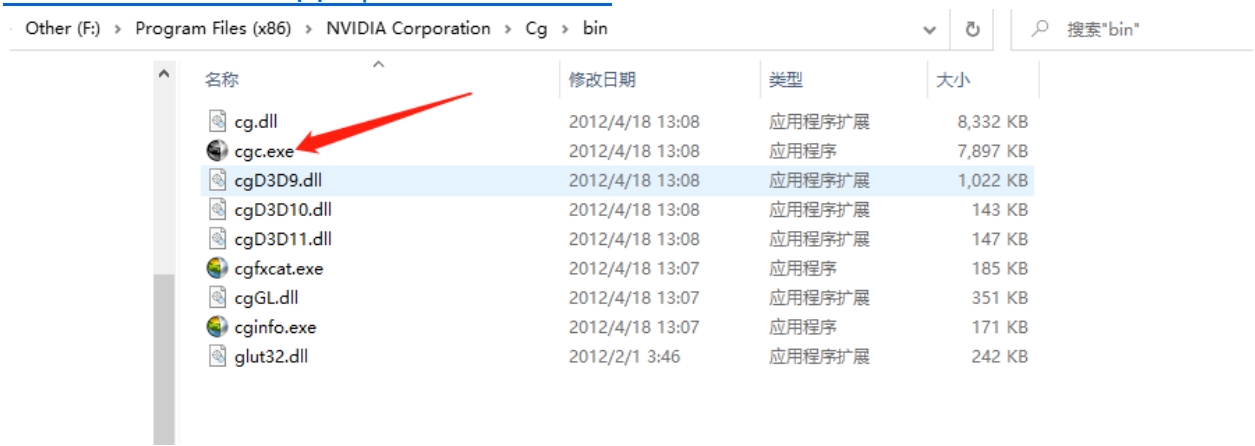
The Cg toolkit is available for a number of different hardware and
OS platforms. As of this writing, supported platforms include:

  - Windows (XP, Vista, 7) on x86/x86-64
  - Linux on x86/x86-64
  - Mac OS X (Leopard, Snow Leopard) on ppc/i386/x86_64

Visit the NVIDIA Cg website at http://developer.nvidia.com/cg-toolkit
for updates and complete compatibility information.

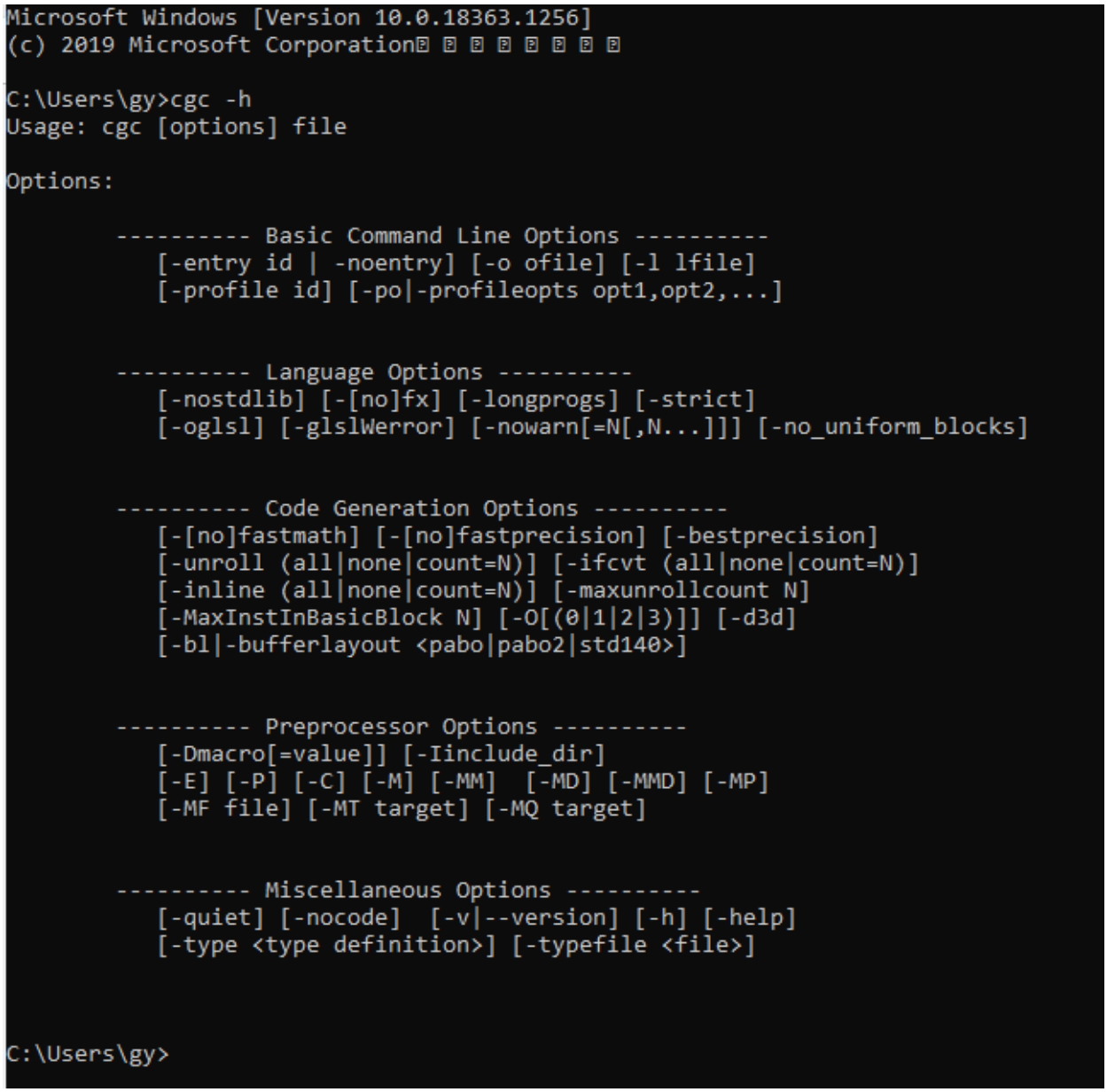


[Direct3D - Win32 apps | Microsoft Docs]



如果 Cg Toolkit 安装正确，在 NVIDIA Corporation\Cg\bin 文件夹下会看到 cgc.exe 文件。

安装完成后，使用cgc -h查看帮助



cgc [options] file
[options]表示可选配置项，file 表示 Cg 程序文件名。

cgc –profile glslv –entry main_v test.cg

-profile 是 profile 配置项名；
glslv 是当前所使用的 profile 名称；
-entry 着色程序的入口函数名称配置项；
main_v 是顶点着色程序的入口函数名； test.cg 是当前的着色程序文件名。
编译器指定的着色程序入口函数名默认为main，通常为了将顶点\片段着色程序入口函数名区别开来，而并不使用默认名 称。在下面所有的例子中，main_v 表示顶点着色程序入口函数名,main_f 表示片段着色程序入口函数名。

CG Profile

一个 Cg profile 定义了一个“被特定图形硬件或 API 所支持的 Cg 语言子集”,Profile 按照功能可以划分为顶点Profile 和片断 Profile，而顶点 profile 和片段 profile 又基于 OpenGL 和 DirectX 的不同版本或扩展，划分为各种版本
当前 Cg compiler 所支持的 profiles 有：
OpenGL ARB vertex programs Runtime profile: CG_PROFILE_ARBVP1 Compiler option: _profile arbvp1

OpenGL ARB fragment programs Runtime profile: CG_PROFILE_ARBFP1 Compiler option: _profile arbfp1

OpenGL NV40 vertex programs Runtime profile: CG_PROFILE_VP40 Compiler option: _profile vp40

OpenGL NV40 fragment programs Runtime profile: CG_PROFILE_FP40 Compiler option: _profile fp40

OpenGL NV30 vertex programs Runtime profile: CG_PROFILE_VP30 Compiler option: _profile vp30

OpenGL NV30 fragment programs Runtime profile: CG_PROFILE_FP30 Compiler option: _profile fp30

OpenGL NV2X vertex programs Runtime profile: CG_PROFILE_VP20 Compiler option: _profile vp20

OpenGL NV2X fragment programs Runtime profile: CG_PROFILE_FP20 Compiler option: _profile fp20

DirectX 9 vertex shaders Runtime profiles: CG_PROFILE_VS_2_X CG_PROFILE_VS_2_0 Compiler options:-profile vs_2_x -profile vs_2_0

DirectX 9 pixel shaders Runtime profiles: CG_PROFILE_PS_2_X CG_PROFILE_PS_2_0 Compiler options: -profile ps_2_x -profile ps_2_0

DirectX 8 vertex shaders Runtime profiles: CG_PROFILE_VS_1_1 Compiler options:-profile vs_1_1

DirectX 8 pixel shaders Runtime profiles: CG_PROFILE_PS_1_3 CG_PROFILE_PS_1_2 CG_PROFILE_PS_1_1 Compiler options: -profile ps_1_3 -profile ps_1_2 -profile ps_1_2 -profile ps_1_1