

Configuring Cg for Use in the Crux Lab

The following steps assume that the Cg Toolkit has been installed onto the Crux machines and only needs to be configured for use within Microsoft Visual C++ 6.0. You will need administrative permissions to install the Cg Toolkit, so contact UT labstaff if you have any questions concerning installation. After Cg is installed, you can configure your environments as follows:

- The paths of Cg's **include** and **lib** directories need to be added to Visual C++'s search path. Open Visual C++ 6.0 and choose Tools->Options. Then click on the [Directory] tab. Select "Include files" from the dropdown menu labeled [Show directories for:]. Now add the Cg include path to the directories list (i.e. C:\NVIDIA Corporation\Cg\include). Now select the "Library files" option from the dropdown menu and add the Cg library path to the directories list (i.e. C:\NVIDIA Corporation\Cg\lib).

NOTE: You will need to consult UT labstaff for the installed location of Cg, or you may have to select the [...] option in the directories list and browse the directory tree to find Cg. Alternatively, you can copy all the header files in the "lib" directory to Visual C++'s "lib" directory and all the "include" header files into Visual C++'s "include" directory; but you would need administrative privileges to do this.

- To allow your system to know where the Cg compiler resides, you need to set up an environmental variable in Windows XP, using the following steps:
 1. Click Start -> Settings -> Control Panel
 2. Double click the **System** icon
 3. Select the **Advanced** tab
 4. Click the Environment Variables button
 5. Click the **New** button in the **System variables** area
 6. Type CG_COMPILER_EXE in the Variable name textbox
 7. Type C:\NVIDIA Corporation\Cg\bin\cgc.exe in the Variable value textbox

Again, consult labstaff if you have trouble finding the Cg install path.

- Next, you have to set up your Visual Studio workspace. The first thing you have to do is to add the path to the CgGL using the following steps:
 1. Click **Project -> Settings** and select the C/C++ tab
 2. Select **All Configurations** from the **Settings For** drop-down list box
 3. Select **Preprocessor** from the **Category** drop-down list box
 4. Add **Cg-runtime\Cg\include** to the **Additional include directories** textbox (separate each directory in your list with a comma)

- Now you have to add the CgGL and/or CgD3D static libraries to your project's linker settings, using the following steps:
 1. Click **Project -> Settings** and select the **Link tab**
 2. Select **Win32 Release** from the **Settings For** drop-down list box
 3. Add CgGL.lib to the list of libraries in the **Object/library modules** text field
 4. Select **Input** from the **Category** drop-down list box
 5. Add **cg-runtime\cg\cgGL\Release** to the **Additional library path** textbox (separate each item with a comma)
 6. Select **Win32 Debug** from the **Settings For** drop-down list box
 7. Add **cg-runtime\cg\cgGL\Debug** to the **Additional library path** textbox (separate each item with a comma)

- Next, you need to create a folder for all your Cg shaders. To do this:
 1. Select **File View** in the Workspace pane on the left side of Visual Studio
 2. Right click on your project and select **New Folder...** in the popup menu
 3. Set the name of the folder to **Cg Shaders** and the extension to .cg and click OK
 4. Right click on the newly created folder and select **Add Files to Folder...** from the popup menu
 5. Add any Cg shaders that you have into the folder.

- Finally, a Custom Build Step allows you to use a specific tool to build your Cg shaders. In this case, we're going to use the Cg compiler, cgc.exe. The following are the steps you have to take to accomplish this:
 1. Right click on a shader in the **Cg Shaders** folder created in the last step and select **Settings** from the popup menu
 2. Select **All Configurations** from the **Settings For** drop-down list box
 3. Add `cgc $(InputPath) -o $(InputName).vp -profile vp20` to the Commands textbox
 4. Add `$(InputName).vp` to the **Outputs** textbox

- If you like to increase the readability of your shaders, you can turn on syntax highlighting when editing your Cg files in Visual Studio 6.0. Here's how:
 1. Copy usertype.dat
 - (C:\NVIDIA Corporation\Cg\msdev_syntax_highlighting\usertype.dat) to the Visual Studio bin directory (typically **C:\Program Files\Microsoft Visual Studio\Common\MSDev98\Bin**)
 2. Start regedit (**Start -> Run -> regedit**) and go to **HKEY_CURRENT_USER\Software\Microsoft\DevStudio\6.0\Text**

Editor\Tabs\Language Settings\C/C++

3. Add cg to the end of the **FileExtensions** key (each extension in the list should be separated with a semicolon)
4. Restart Visual Studio

Your shaders should now have syntax highlighting. Of course, you will need to have administration privileges to perform this step, so consult labstaff if this is desirable.