

CompiledEffect Sample

 [Collapse All](#)

This sample shows how an **ID3DXEffect** object can be compiled when the project is built and loaded directly as a binary file at run time.



Path

Source	<i>SDK root\Samples\C++\Direct3D\CompiledEffect</i>
Executable	<i>SDK root\Samples\C++\Direct3D\Bin\x86 or x64\CompiledEffect.exe</i>

Overview

Large-scale projects may contain dozens or even hundreds of shader and effect files. Compiling these files at run time can cause a noticeable delay during initialization, and this delay is incurred each and every time the project is run. In spite of this, it is beneficial to compile at run time when developing or debugging effect files as this allows more debugging flexibility.

Once development and/or debugging of these shaders are completed, these files can be compiled as part of your project build process. Since these files will only need to be compiled when changes are made, no additional build time is spent on effect files which have already been debugged and compiled.

The process illustrated in this sample is identical, whether compiling a complete effect object or individual shaders. All that would change would be the shader profile passed to the compiler on the command line.

How the Sample Works

The code is virtually identical to the [HLSLwithoutEffects Sample](#) except for a couple important differences:

- The call to **D3DXCreateEffectFromFile** is passed the filename of the compiled effect `CompiledEffect.fxo` instead of the filename of the effect source `CompiledEffect.fx`.
- No debugging flags are passed to `D3DXCreateEffectFromFile`. Instead, flags are passed to the effect compiler to enable shader debugging. The choice of compiler flags depends on the Visual Studio .NET build configuration chosen.

The program used to compile shaders at build time is **fxc**. This command-line utility ships with the SDK, and wraps the `ID3DXEffectCompiler` calls (which are called to compile the shader). You could create your own shader compiler fairly easily using the `ID3DXEffectCompiler`, but `fxc` should be flexible enough for most needs. The commands used to build release and debug versions of a typical .fx file are:

Release:	<code>fxc /Tfx_2_0 /FoMyEffect.fxo MyEffect.fx</code>
Debug:	<code>fxc /Od /Zi /Tfx_2_0 /FoMyEffect.fxo MyEffect.fx</code>

Run `fxc` without any arguments for usage information.

Build Environment Configuration

Here are the steps taken to configure the build environment. These steps should be performed for every effect file you want to be compiled as part of the build process:

1. Open the `CompiledEffect` sample inside Visual Studio .NET.
2. From within the Solution Explorer window, right click on `CompiledEffect.fx` and select Properties from the context menu.
3. Select Configuration Properties/Custom Build Step to view the custom build step directives. The specific compile command issued depends on the project build configuration, as you can verify by scrolling through the Configuration combo box.

© 2010 Microsoft Corporation. All rights reserved.
Send feedback to DxSdkDoc@microsoft.com.
Version: 1962.00