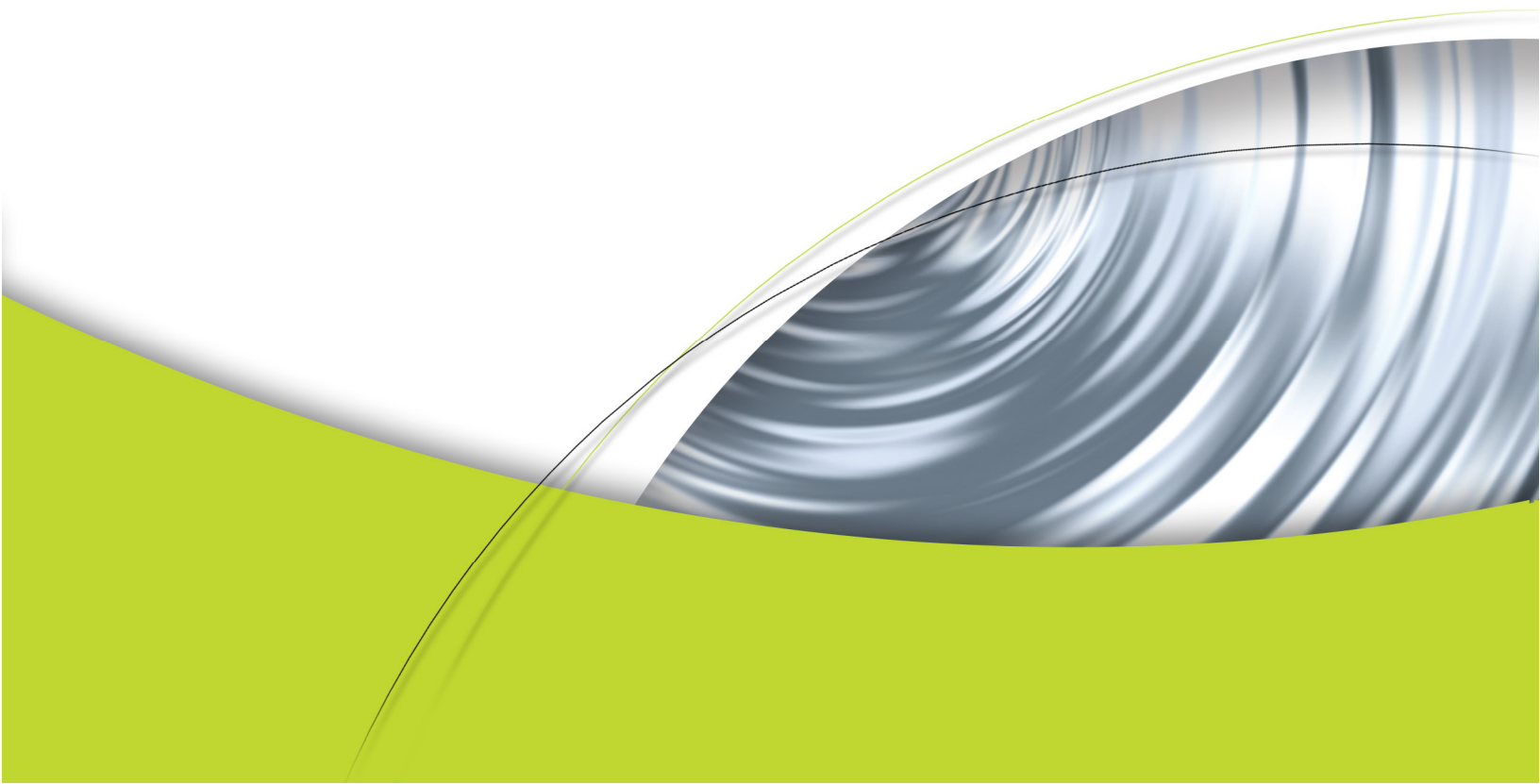




# Cg Toolkit

Cg 2.2 beta  
February 2009  
Release Notes



# Cg Toolkit Release Notes

The Cg Toolkit allows developers to write and run Cg programs using a wide variety of hardware and OS platforms and graphics APIs. Originally released in December 2002, the Toolkit now supports over 20 different DirectX and OpenGL profile targets. It provides a compiler for the Cg language, runtime libraries for use with the OpenGL and DirectX graphics APIs, support for the CgFX effect files, example applications, and extensive documentation.

The 2.2 beta release of Cg incorporates the following updates:

- ❑ DirectX10 and GLSL geometry profiles (gs\_4\_0 AND glslg)
- ❑ Support for "latest" profile keyword in CgFX compile statements
- ❑ Additional API routines (see **New API** for a complete list)
- ❑ Migrated the OpenGL examples onto GLEW
- ❑ New examples including:
  - ❑ examples/Direct3D10/advanced
    - ❑ combine\_programs
    - ❑ gs\_shrinky
    - ❑ gs\_simple
  - ❑ examples/OpenGL/advanced
    - ❑ cgfx\_latest
  - ❑ examples/Tools
    - ❑ cgfxcat
    - ❑ cginfo
- ❑ New documentation
  - ❑ Updated reference manual pages for new profiles and entry points
- ❑ Bug fixes

Visit the NVIDIA Cg website at [developer.nvidia.com/page/cg\\_main.html](http://developer.nvidia.com/page/cg_main.html) for complete availability and compatibility information.

Bug reports, issues, and feedback can be sent to [cgsupport@nvidia.com](mailto:cgsupport@nvidia.com).

---

## Supported OS/Hardware Platforms

Cg is available for these platforms:

- ❑ Windows 32
- ❑ Windows 64
- ❑ Linux x86
- ❑ Linux x86-64
- ❑ MacOS 10.4 (Tiger)
- ❑ MacOS 10.5 (Leopard)
- ❑ Solaris 10 x86

The Cg Runtime libraries include:

- ❑ The Cg core runtime library for managing parameters and loading programs
- ❑ The CgGL runtime library for OpenGL based applications
- ❑ The CgD3D10 runtime library for DirectX 10 based applications
- ❑ The CgD3D9 runtime library for DirectX 9 based applications
- ❑ The CgD3D8 runtime library for DirectX 8 based applications

---

## Supported Profiles

The Cg compiler currently supports the following hardware profiles:

### OpenGL

- ❑ **gp4gp** NV\_geomemtry\_program4
- ❑ **gp4vp** NV\_vertex\_program4
- ❑ **gp4fp** NV\_fragment\_program4
- ❑ **glslg** OpenGL Shading Language (GLSL) for OpenGL 2.0 geometry shader
- ❑ **glslv** OpenGL Shading Language (GLSL) for OpenGL 2.0 vertex shader
- ❑ **glslf** OpenGL Shading Language (GLSL) for OpenGL 2.0 fragment shader
- ❑ **arbvp1** ARB\_vertex\_program 1.0
- ❑ **arbfvp1** ARB\_fragment\_program 1.0
- ❑ **vp40** ARB\_vertex\_program + NV\_vertex\_program2 option
- ❑ **fp40** ARB\_fragment\_program + NV\_fragment\_program2 option
- ❑ **vp30** NV\_vertex\_program 2.0
- ❑ **fp30** NV\_fragment\_program 1.0
- ❑ **vp20** NV\_vertex\_program 1.0
- ❑ **fp20** NV\_register\_combiners and NV\_texture\_shader

## DirectX 10.0

- ❑ **gs\_4\_0** HLSL10 Geometry Shader
- ❑ **vs\_4\_0** HLSL10 Vertex Shader
- ❑ **ps\_4\_0** HLSL10 Fragment Shader

## DirectX 9.0c

- ❑ **hlslv** HLSL9 Vertex Shader
- ❑ **hlslf** HLSL9 Fragment Shader
- ❑ **vs\_3\_0** Vertex Shader 3.0
- ❑ **ps\_3\_0** Pixel Shader 3.0

## DirectX 9

- ❑ **vs\_2\_x** Extended Vertex Shader 2.0
- ❑ **ps\_2\_x** Extended Pixel Shader 2.0
- ❑ **vs\_2\_0** Vertex Shader 2.0
- ❑ **ps\_2\_0** Pixel Shader 2.0

## DirectX 8 & 9

- ❑ **vs\_1\_1** Vertex Shader 1.1
- ❑ **ps\_1\_3** Pixel Shader 1.3
- ❑ **ps\_1\_2** Pixel Shader 1.2
- ❑ **ps\_1\_1** Pixel Shader 1.1

---

## Improvements & Bug Fixes

### Improvements

- ❑ Added ability to query the supported profiles at runtime
- ❑ Added API to programmatically control profiles returned as “latest”
- ❑ Added API to discover the enumerants associated with a state
- ❑ Migrated OpenGL examples onto GLEW
- ❑ New program for dumping CgFX files (cgfxcat)

### Improvements: Documentation

- ❑ **Note:** The Cg Users Manual has **not** been updated for this release.
- ❑ Updated reference manual for the new profiles and entry points.

### Bug Fixes

- ❑ Various bugs have been fixed.

---

## Compatibility Notes

There aren't any known compatibility issues with programs written against Cg 2.1. For programs written against Cg 2.0 or earlier, refer to the Compatibility Notes section of the release notes for Cg 2.1.

## Known issues

### Known runtime issues

- ❑ **cgGetParameterValues** incurs a penalty in both performance and memory footprint and using it is strongly discouraged. Use **cgGetParameterValue** or **cgGetParameterDefaultValue** instead.
- ❑ **cgCopyProgram** and **cgCopyEffect** do not work.
- ❑ Loading precompiled code via **CG\_OBJECT** in **cgCreateProgramFromFile** doesn't work for shaders which use semantic type modifiers.
- ❑ The DirectX 8 runtime does not support Cg interfaces.
- ❑ The Cg runtime does not support creating shared parameters containing varying members.
- ❑ Unsized arrays and interface parameters cannot currently be used on the right-hand side of state assignments. Doing so will trigger an error.
- ❑ Values set by **cgGLSetOptimalOptions(...)** can be un-set after a call to **cgDestroyContext()**. As a work around, call **cgGLSetOptimalOptions()** after each call to **cgDestroyContext()** when more Cg contexts are going to be created.

### Known compiler issues

- ❑ Long shader programs that make heavy use of interfaces may still result in very long compilation time.
- ❑ Very little error checking is performed on the OpenGL state semantics string (**state.\***); it is just copied to the output assembly. As a result, a typo in the string may compile correctly, and no error will be apparent until the application attempts to load the assembly shader.
- ❑ Error reporting: Some error and warning messages are not as clear as they could be. Some of the issues to be aware of are:
  - ❑ Reported line numbers do not match source code lines when standard library functions are being used
  - ❑ In some cases, errors are not reported in the order they appear in the program
  - ❑ Errors are not reported when constants are out of range for untyped constants.
- ❑ Side-effects in conditional expressions ('?:') and logical expressions ('&&' and '||') are always evaluated, regardless of the condition, as specified in the Cg language specification. Hence developers need to watch out for this case.
- ❑ At most one binding semantic per uniform variable is supported by the compiler. Multiple profile-specific binding semantics per uniform variable are not supported.

- ❑ Only loops with a single induction variable are unrolled. Loops that require more than 1 induction variable will fail to compile on older profiles that do not support loops.
- ❑ Local variable arrays which are written to in one block of code, and then read via a non-constant index in a different block will fail to compile on older hardware that does not support this feature. Current hardware supports this feature.
- ❑ Invalid Cg programs can, at times, generate invalid code, instead of a compiler error.

## Known profile-specific issues

- ❑ The GLSL geometry profile has issues on OSX.
- ❑ The ps2\* profiles do not support MRTs
- ❑ Because the underlying hardware support for the fp20 and ps\_1\_\* profiles is quite limited and inflexible, it isn't always possible to compile even seemingly simple Cg programs under these profiles. For more details on these limitations, please see the NV\_register\_combiners and NV\_texture\_shader OpenGL extension specifications, or the DirectX PixelShader 1.\* specifications.
- ❑ The FOG varying input semantic is not yet supported under the fp20 profile.

---

## New API

Cg 2.2 adds new API to query supported profiles, control the latest profile keyword, discover registered state enumerants, as well as other utility functions. Here is the complete list of the new routines:

<b>cgGetDomain</b>	<b>cgGetProgramDomain</b>
<b>cgGetDomainString</b>	<b>cgGetStateEnumerant</b>
<b>cgGetNumStateEnumerants</b>	<b>cgGetStateLatestProfile</b>
<b>cgGetNumSupportedProfiles</b>	<b>cgGetSupportedProfile</b>
<b>cgGetParameterClassEnum</b>	<b>cgIsProfileSupported</b>
<b>cgGetParameterClassString</b>	<b>cgSetStateLatestProfile</b>
<b>cgGetProfileProperty</b>	<b>cgGLGetOptimalOptions</b>



---

## Release Types

Cg is released in two forms:

1. The Cg Toolkit provides a complete Cg Software Development Kit (SDK) including documentation, examples, standalone compiler, headers and libraries.
2. Cg Binary Distributions provide updated redistributable libraries that Cg-based applications can ship with.

SDK versions are released as platform specific installers containing the full toolkit (libraries, documentation, examples, etc.) They can be downloaded from

[http://developer.nvidia.com/object/cg\\_toolkit.html](http://developer.nvidia.com/object/cg_toolkit.html)

Binary distributions contain only the libraries, and all supported platforms are bundled in a single file. The libraries supplied in a binary distribution should be feature-for-feature and bug-for-bug compatible across all the platforms supported by a given distribution (meaning are all compiled from the same source code). Cross-platform software vendors are encouraged to redistribute Cg libraries from a single binary distribution to minimize platform variances in Cg.

Cg binary distributions can be found at

<http://developer.nvidia.com/object/cg-redistributable-binaries.html>

## Distribution License

The docs directory contains a file `license.pdf` providing a non-exclusive, world-wide, royalty free license for redistributing Cg with your applications. See this license for details.

## Release History

The following table summarizes release dates and library versions for Cg releases:

Cg Release Name	Release Date	Library Version
SDK2	11/20/08	2.1.0016
SDK1	10/15/08	2.1.0012
beta	08/07/08	2.1.0009
binary1	06/05/08	2.0.0016
SDK3	04/15/08	2.0.0015
SDK2	01/30/08	2.0.0012

The Cg library version is returned by `cgGetString(CG_VERSION)`

## Change History

### 2.1.0016

- ❑ Fixed crash when `cgGL` routines were called without a current GL context
- ❑ Fixed a bug in compiling CgFX files with multiple geometry shaders
- ❑ Fixed D3D10 runtime's handling of the parameter buffer (cbuffer)
- ❑ D3D10 runtime now correctly deals with true integer parameters
- ❑ Shared parameter arrays were not being updated correctly
- ❑ Corrected some minor issues in the Sun package file

### 2.1.0012

- ❑ Added translation profiles for DirectX10 Shader Model 4
- ❑ Shader source virtual file system for compilation
- ❑ Callback for providing shader source included by the compiler
- ❑ Added more 'clip' functions for improved compatibility with HLSL
- ❑ Solaris release in Sun package format
- ❑ Fixed sampler initialization by the GLSL profile
- ❑ Fixed handling of integer parameters values
- ❑ Added several new examples

## **Notice**

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

## **Trademarks**

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation.

Microsoft, Windows, the Windows logo, and DirectX are registered trademarks of Microsoft Corporation.

OpenGL is a trademark of SGI.

Other company and product names may be trademarks of the respective companies with which they are associated.

## **Copyright**

Copyright © 2004-2009 NVIDIA Corporation. All rights reserved.



**NVIDIA.**

NVIDIA Corporation  
2701 San Tomas Expressway  
Santa Clara, CA 95050  
[www.nvidia.com](http://www.nvidia.com)