Why Do We Do This?

Investing in Developers Worldwide
- Powerful tools for building games
  - Performance Analysis
  - Content Creation
  - Software Development
- Practical SDK with technical documentation
- Web Site and Newsletter [developer.nvidia.com](http://developer.nvidia.com)

Registered Developer Program
- Pre-Release Drivers
- Early Access to Developer Tools
- Online Support Forums & Bug Submission
  Sign up now at [developer.nvidia.com](http://developer.nvidia.com)
NVIDIA SDK
The Source for GPU Programming

Hundreds of code samples and effects that help you take advantage of the latest in graphics technology.

- Tons of updated and all-new DirectX and OpenGL code samples with full source code and helpful whitepapers:
  - GPU Cloth, Geometry Instancing, Rainbow Fogbow, 2xFP16 HRD, Perspective Shadow Maps, Texture Atlas Utility, ...

- Hundreds of effects, complete with custom geometry, animation and more:
  - Skin, Plastics, Flame/Fire, Glow, Gooch, Image Filters, HLSL Debugging Techniques, Texture BRDFs, Texture Displacements, HDR Tonemapping, and even a simple Ray Tracer!

SDK Browser

100+ code samples
250+ compelling effects

Search and filter to find exactly what you need
Practical Code Samples

• Video Filtering
• Antialiasing with Post-processing
• Deferred Shading
• Geometry Instancing

Compelling Effects

• Image processing shaders
• Post-processing effects
• UV & MIP level debugging shaders
• Powerful examples of DXSAS
FX Composer

- CREATE your shaders in a high powered IDE
  - Native support for HLSL .FX development
  - Render-to-texture effects
  - Save out pre-rendered ("baked") textures
- DEBUG your shaders with visual shader debugging
  - Unique real-time preview of intermediate targets
  - Import your own geometry, complete with animation
- TUNE your shader performance with advanced analysis
  - Vertex & pixel shader performance metrics
  - GPU-specific scheduling & disassembly

Full plug-in SDK and scripting support for automation

Be sure to check out the complete User Guide and helpful tutorials!

Also see the full presentation dedicated to FX Composer.

EverQuest® content courtesy Sony Online Entertainment Inc.
EverQuest is a registered trademark of Sony Computer Entertainment America Inc.
© 2004 Sony Computer Entertainment America Inc. All rights reserved.
CREATE your shaders in a high powered development environment
• Sophisticated text editing with intellisense (auto-complete) and syntax highlighting
• Work directly with HLSL .FX files, creating multiple techniques and passes
• Use the .FX files you create with FX Composer directly in your application
• Convenient, artist-friendly graphical editing of shader properties
• Supports Microsoft DirectX standard HLSL semantics and annotations
• Support for all DirectX 9.0 shader profiles
• Develop your shaders on your own models with lighting and animation

DEBUG your shaders with visual shader debugging features
• Interactive compiler helps you find and fix problems
• Visible preview of source and intermediate textures targets
• Interactive jump-to-error feature helps you fix problems quickly

TUNE your shader performance with advanced analysis and optimization
• Enables performance tuning workflow for vertex and pixel shaders
• Simulates performance for the entire family of GeForce FX GPUs
• Capture of pre-calculated functions to texture look-up table
• Provides empirical performance metrics such as GPU cycle count, register usage, pixels per second, and more
• Optimization hints notify you of performance bottlenecks
**FX Composer In Your Pipeline**

HLSL Shaders → Optimized Shaders
Textures → Property Sets
Scene Data → Generated Textures
→ Package Files

**FX Composer**
Create – Debug – Tune

Scripting automation & SDK for custom importer/exporter plug-ins

---

**Goes-in**
- Supports all DirectX texture formats
- Import scene data from .obj, .x, .ply, or .x files

**Comes-out**
- Use the Shader Perf panel to optimize your shaders
- Save properties sets as XML files (re-apply in engine)
- Save textures generated by HLSL functions (e.g. noise)
- Package shaders, textures & full scene in one file for review/collaboration

**Scripting automation & SDK for custom importers/exporters**
- Sample code for custom importers & exporters
- Save properties bundles in your own format
- Automate just about anything using .NET scripting
  (e.g. screenshots for all shaders in a directory)
Head model comparison
Left: 18k poly reference model
Right: 4k poly working model
(auto simplified using progressive mesh deformation algorithm)

Note various visualization modes (top to bottom):
1. Filled
2. Textured
3. Chart View
4. Object Space Normals
Dawn’s Bustier
Left 38K polygons
Right 2K polygons with Melody normal map

*Can you tell the difference?*

**Visualization Options Include:**
- Vertex or per-pixel normals (textured or not)
- Charts & attribute groups
- Tangent to object space
- Tangent or object space light mapping
- Combined with Filled, Wireframe or Outlined mode
Goes-in
• Import models from .obj, .3ds or .ply files.
• Import a low-res model or generate LODs using Melody

Comes-out
• Multiple LODs ()
• Normal map file for each low-res model (tons of options)
• Object space and tangent space normal maps
• FP support (8 or 32 bits / channel)
• Displacement maps
• Re-sampled color map (normal & color)
• Greyscale Height map
• Generates ambient occlusion term

Visual Preview
• Simple object manipulation
• Powerful visualization modes
Photoshop Plug-ins
• Allows saving in various formats, incl. DXTC (.dds files)
• Generate mip-maps in variety of ways
  (box filter to Kaiser gamma-space filter)
• Preview w/ 3D rendering options
  (e.g. anisotropy, format comparisons etc.)
• Create normal maps from height maps

nvDXT & mip map utilities
• Command line interface for pipeline automation
• Available .lib for tools integration
• Detach & Stitch utilities to manipulate MIP maps

DDS Thumbnail Viewer
• Provides convenient preview of DXT compressed files.

Also check out our Texture Atlas whitepaper...
Be sure to check out the User Guide for full details!

When NVPerfHUD is activated, you can perform graphics pipeline experiments, display graphs of performance metrics, and explore potential problems using several performance visualization modes. You can also switch to the Debug Console or Frame Analysis Mode for deeper analysis. Use these shortcut keys to switch modes:

**F5 Performance Analysis Mode**
Use timing graphs and directed experiments to identify bottlenecks.

**F6 Debug Console Mode**
Review messages from the DirectX Debug runtime, NVPerfHUD warnings and custom messages from your application.

**F7 Frame Analysis Mode**
Freeze the current frame and step through your scene one draw call at a time, using advanced State Inspectors for state of the graphics pipeline.
Configure the information displayed on the screen and perform several graphics pipeline experiments:

- **F1** Cycle display of helpful information
- **B** Show Batch Size Histogram
- **F** Fade Background
- **H** Hide Graphs
- **W** Show Wireframe
- **D** Show Depth Complexity
- **T** Isolate the texture unit by forcing the GPU to use 2x2 textures
- **V** Isolate the vertex unit by using a 1x1 scissor rectangle to clip all rasterization and shading work
- **N** Eliminate the GPU (and state change overhead) by ignoring all draw calls
When NVPerfHUD is activated, you can perform graphics pipeline experiments, display graphs of performance metrics, and explore potential problems using several performance visualization modes. You can also switch to the Debug Console or Frame Analysis Mode for deeper analysis.

**Index Unit State Inspector**
Verify all the information used to fetch the vertex data and make sure the geometry associated with the current draw call is correct.

**Vertex and Pixel Shader State Inspectors**
Verify that the shader program constants and textures are correct for the current draw call. Make sure the constants are not #NAN or #INF.

**Raster Operations State Inspector**
Confirm that the back buffer format has an alpha component when blending doesn’t seem to be working properly, verify that opaque objects are not drawn with blendEnable, etc.
This handy Quick Reference sheet is included in the NVPerfHUD installer.

Be sure to read through the complete User Guide with chapters on:

• Enabling NVPerfHUD
• Bottleneck Identification
• Crushing Bottlenecks
• Advanced State Inspectors
• Troubleshooting
Identify shader performance issues:
- Run once across all your shaders to create a baseline report
- Run the same version again later to identify which shaders have improved / degraded performance

Identify driver performance issues:
- Run once across all your shaders to create a baseline report
- Then run a different version on the same shaders to identify improvements and/or regressions

*Please let us know if you find any driver performance issues!!*
Utilities, libraries and more...

- **NVMeshMender** (C++ src code)
  - Fixes problem geometry
  - Prepares meshes for per-pixel lighting

- **NVTriStrip** (.lib & src code)
  - Cache-aware creation of optimized tri lists or strips

**NVMeshMender**:
- Creates tangent basis for per-pixel lighting
- Creates smoothed normals
- Creates u, v coordinates
- Handles mirrored (u,v) and cylindrical wrapping

**NVTriStrip library**:
- Library to stripify arbitrary geometry meshes
- Flexibly optimizes for postTnL vertex caches
- Option to stitch strips using degenerate triangles
- Option to remap indices to improve vertex buffer locality
- Outputs lists or strips
Questions / Feature Requests?

All of this and more, available now at

developer.nvidia.com
The Source for GPU Programming

Please send questions, feature requests & comments about our SDK and developer tools to:

sdkfeedback@nvidia.com
The Source for
GPU Programming

developer.nvidia.com

- Latest News
- Developer Events Calendar
- Technical Documentation
- Conference Presentations
- GPU Programming Guide
- Powerful Tools, SDKs and more ...

Join our FREE registered developer program for early access to NVIDIA drivers, cutting edge tools, online support forums, and more.

developer.nvidia.com

©2005 NVIDIA Corporation. NVIDIA, the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation, in the U.S. and/or other countries. All rights reserved.