

Ignacio Castaño
Daniel Horowitz



NVIDIA®

State of the Art
Cross Platform Shader
Development with FX Composer 2

Agenda



- **FX Composer 2.0 Overview**
- **Cross-Platform Shader Authoring**
- **Production Pipeline Integration**
- **Conclusion**
- **Q&A**

FX Composer 2.0

What and Who?



● What is it for?

- Shader Authoring IDE
- Debugging and Profiling
- Scene Integration
- Asset Management

● Who is it for?

- Graphics Programmers
- Technical Directors
- Technical Artists
- Artists



Your Requirements

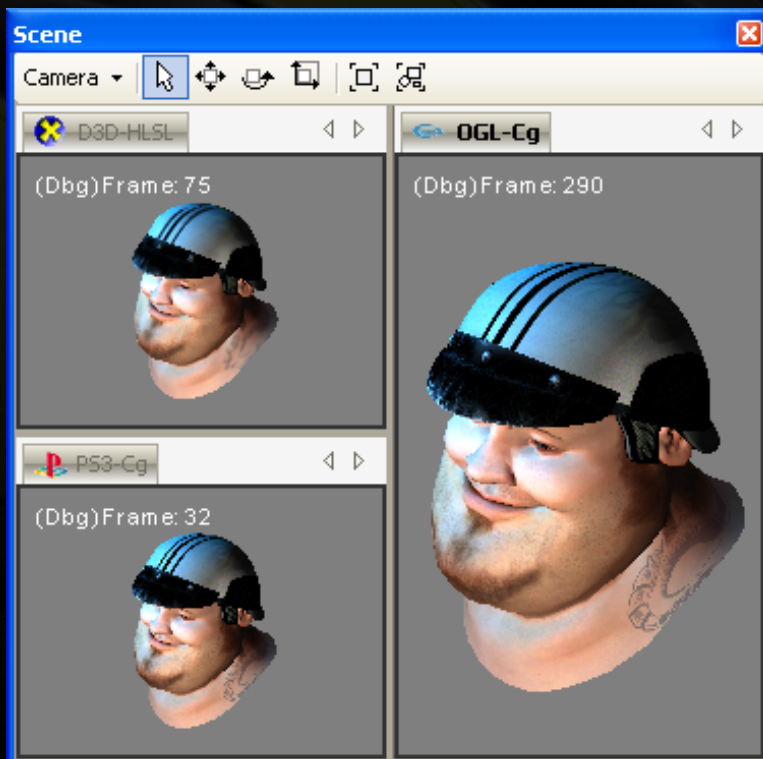


- **Handling of complex rendering**
- **Highly customizable layout management**
- **Powerful user interface**
- **Shader performance profiling**
- **Plug-in based architecture**
- **Scriptable**

Flexible Render



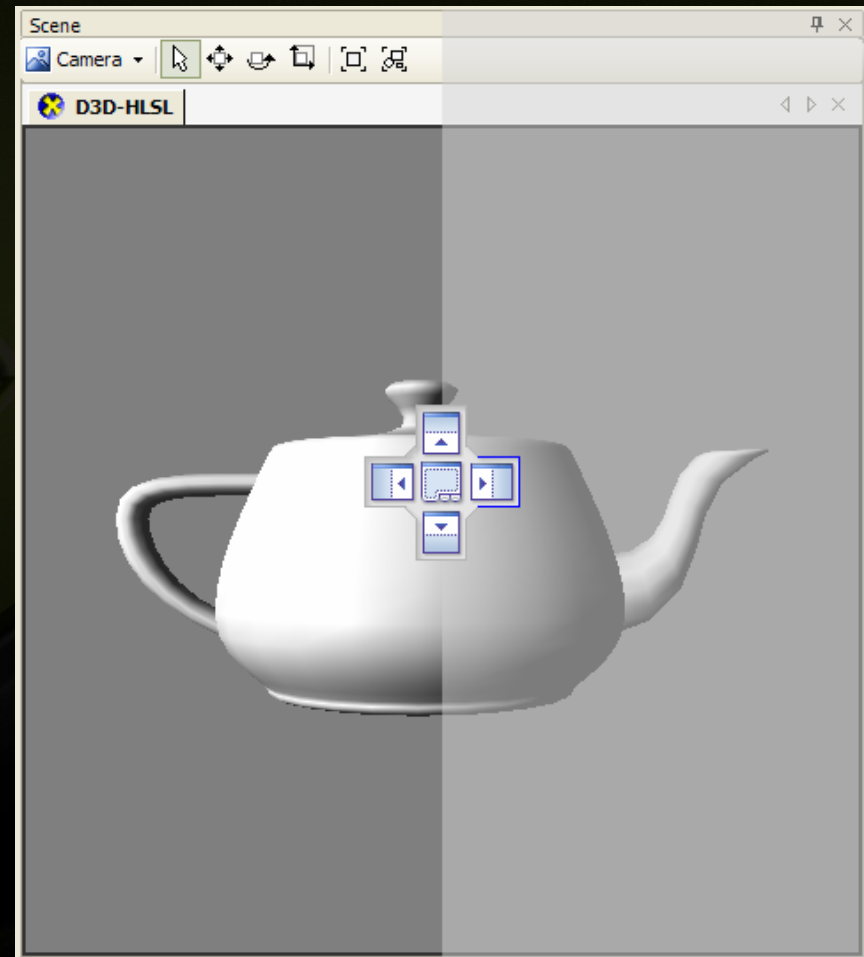
- Many API & shading language combinations
 - OpenGL, Direct3D, PlayStation3™, OpenGL ES*
 - Cg, HLSL, GLSL*
- Surface and fullscreen shaders



Improved User Interface



- **Customized user layouts**
 - Show or hide panels
 - VC2005 docking style
 - Save/Load layouts



Improved User Interface



The screenshot displays a 3D software interface with several key components:

- Code Editor:** Shows a C++ shader function `float4 fresPS_t(vertexOutput IN) : COLOR` with lines 246-301. The code includes texture sampling and material property assignments.
- Scene Hierarchy:** A tree view on the left lists categories: Cameras, Effects, Geometries, Images, Lights, Materials, and Scenes.
- Material Scene Bindings:** A panel on the right shows a `PointLight` material bound to `<Unbound>`. It includes a table for `Node Properties: No` with columns for `Lights` and `Uri`.
- Property Editor:** A central panel shows material properties for `Bright Surface Color` (0.8 0.5 0.1), `Dark Surface Color` (0.5 0.4 0.05), and `Gooch cool tone` (0.05 0.05 0.6).
- Color Picker:** A circular color wheel is visible, with RGB values: R=0.500, G=0.375, B=0.234, and A=1.000.
- Material Parameters:** A slider for `Minimum for Gloss Dr` is set to 0.7, and a `Strength of Glossy` parameter is also visible.
- 3D Viewport:** The bottom right shows a 3D scene with a central object and several point lights, rendered in a wireframe style.

Typical FX Composer Layout



The screenshot displays the NVIDIA FX Composer 2 interface, organized into several key panels:

- Management:** The top-left panel shows the Library/Viewer and Project Explorer. The Library/Viewer contains a tree structure with folders like 'Cameras', 'Effects', and 'scene26_ms_eye_left-fx'. The Project Explorer shows the current project structure.
- Coding:** The central panel is the Material Editor, displaying a Cg shader file named 'MikeShared.cgfx'. The code includes preprocessor directives for debugging and material properties, such as `#define PERMIT_BUMP` and `#define DEFAULT_EXP 25.0`.
- Properties:** The top-right panel shows the Properties window for the selected material, 'scene26_ms_helmet'. It lists various parameters like Position, Look Direction, and World Bounds ABB, along with their values.
- Textures:** The bottom-left panel is the Texture Explorer, showing a grid of texture thumbnails. The current frame is 43.
- Info, Scripting, Errors:** The bottom-middle panel is the Python console, displaying the output of a script execution, including the path to the material file and the extension name.
- Preview:** The bottom-right panel is the Scene preview window, showing a 3D render of a character's head and helmet. It includes a camera view and a timeline showing the current frame (34).



DEMO: Shader Authoring

- Loading Project
- Compiling
- Errors and Tasks

```
163     float4 diffContrib;  
164     float4 specContrib;  
165     gooch_shared(IN,diffContrib, specContrib)  
166     float4 result = diffContrib + specContrib;  
167     return result;  
168 }  
169  
170 float4 goochT_PS(vertexOutput IN) :COLOR  
171
```

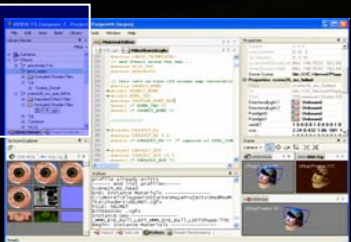
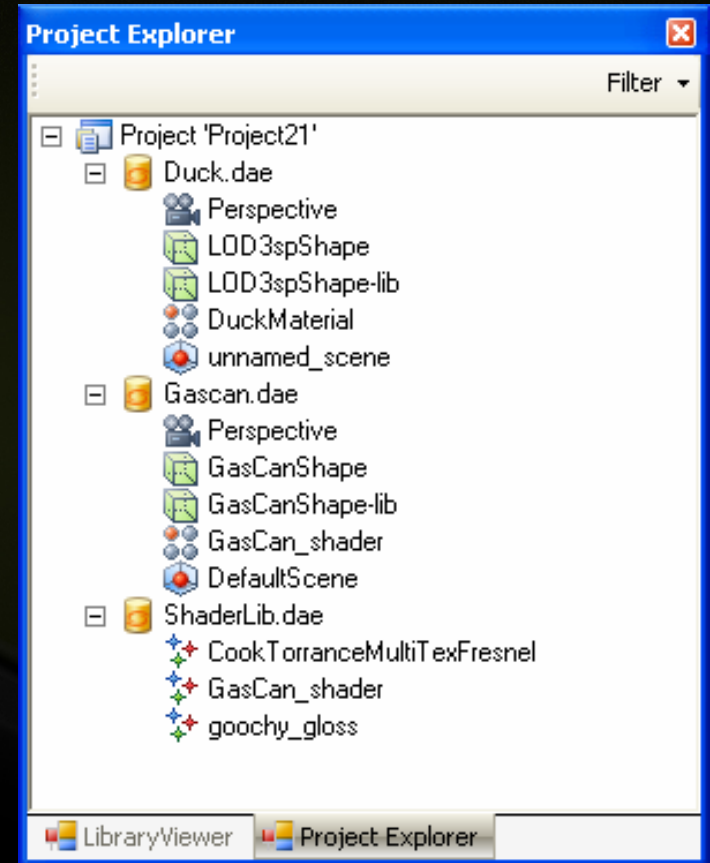
Task List

!	Description	File	Line
⚠	warning C7011: implicit cast from "float4" to "float3"	c:\documents and ...	133
⊗	error C0501: type name expected at token ")"	c:\documents and ...	154
⊗	error C0000: syntax error, unexpected ')', expecting ';' or '.' at token ")"	c:\documents and ...	154
⊗	error C0501: type name expected at token "float4"	c:\documents and ...	166
⊗	error C0000: syntax error, unexpected type identifier, expecting ';' or '.' at token "float4"	c:\documents and ...	166

Project Explorer



- **Manage multiple documents**
- **Documents contain multiple assets**
- **Assets are effects, materials, meshes, and other scene elements**
- **Organize you assets**
 - Move, copy, delete, rename
 - Drag and drop
 - One or many documents
 - Effect Libraries
- **COLLADA**





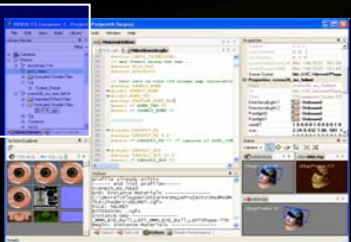
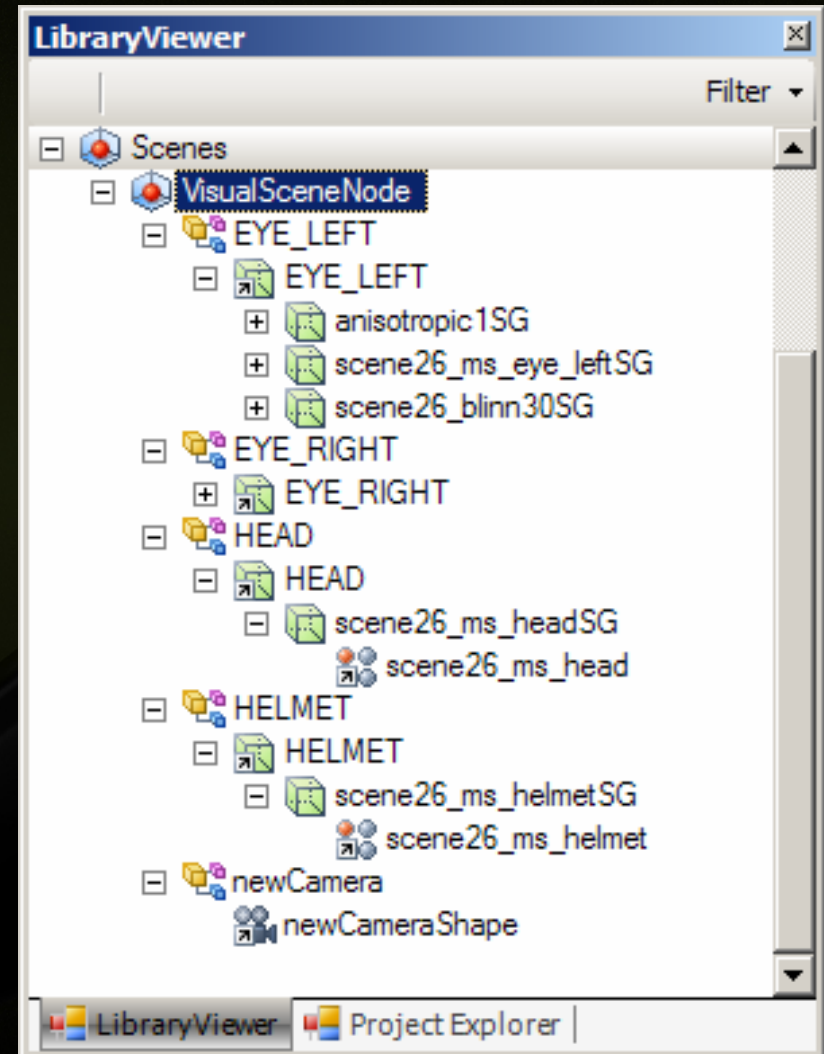
- **Open asset exchange format (.dae)**
- **Governed by the Khronos Group**
 - Includes numerous ISVs and IHVs
 - Mature DCC plugins for extensive support
- **Supported by FX Composer 2**
 - Import & Export
 - Supports *effects and materials*
 - Facilitates asset exchange with DCC apps
 - Other file formats supported



Library View



- Organize across documents
- Sort assets by type
- Visualize Assets
 - Scenes
 - Effect
- Authoring



Effect Authoring

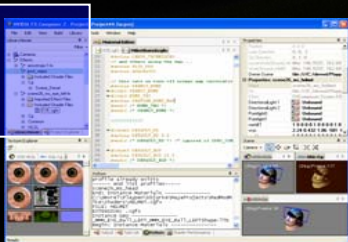
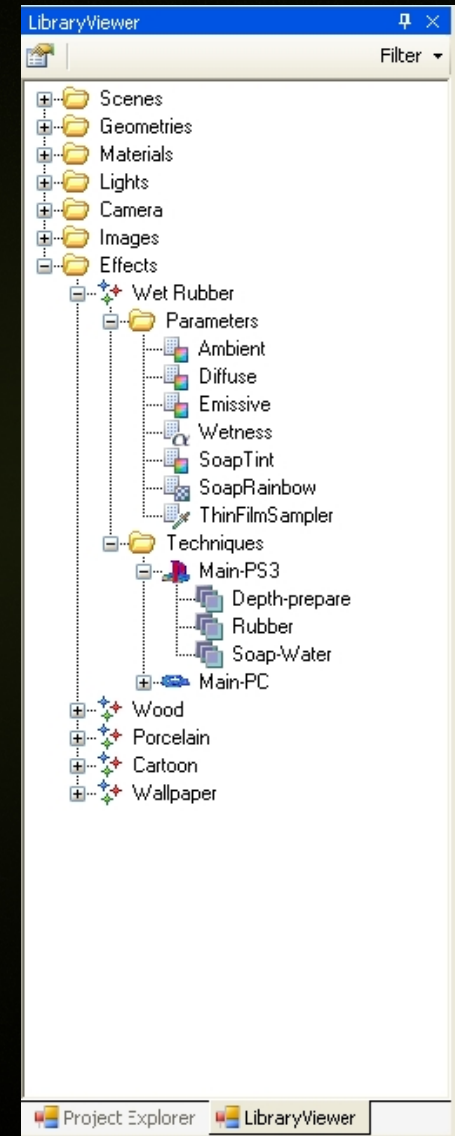


View effect structure

- Techniques
- Passes
- Parameters

Authoring using toolbars and context menus

- Add children
- Remove children
- Advanced options



CgFX & COLLADA FX Cg

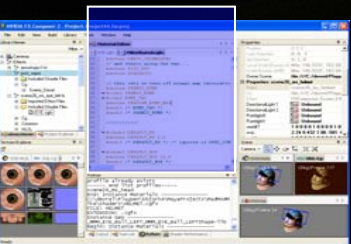


● CgFX

- Hand coded
- Less user interface assistance

● COLLADA FX for Cg authoring

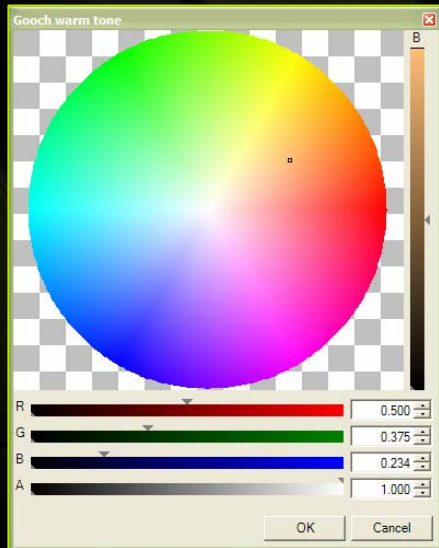
- Most user friendly experience
- Fully editable via user interface
- Can migrate your CgFX assets
- Less hand coding
 - Zero XML
 - Cg, GLSL only when writing the GPU shader code



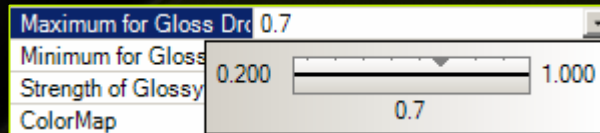


DEMO: Material Authoring

- Creating materials
- Assigning effects
- Tweaking material parameters



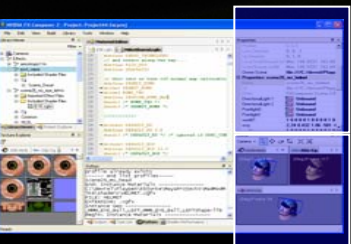
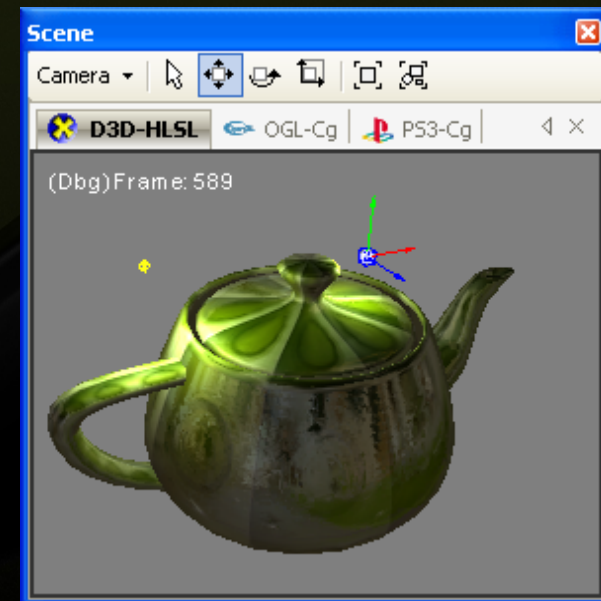
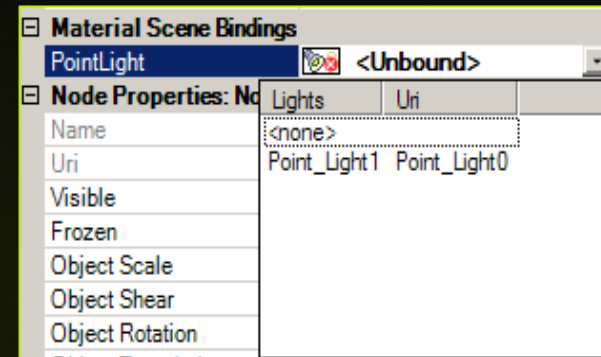
Bright Surface Color	0.8 0.5 0.1
Bright Surface Color	0 0 0
Dark Surface Color	0.5 0.4 0.05
Gooch cool tone	0.05 0.05 0.6



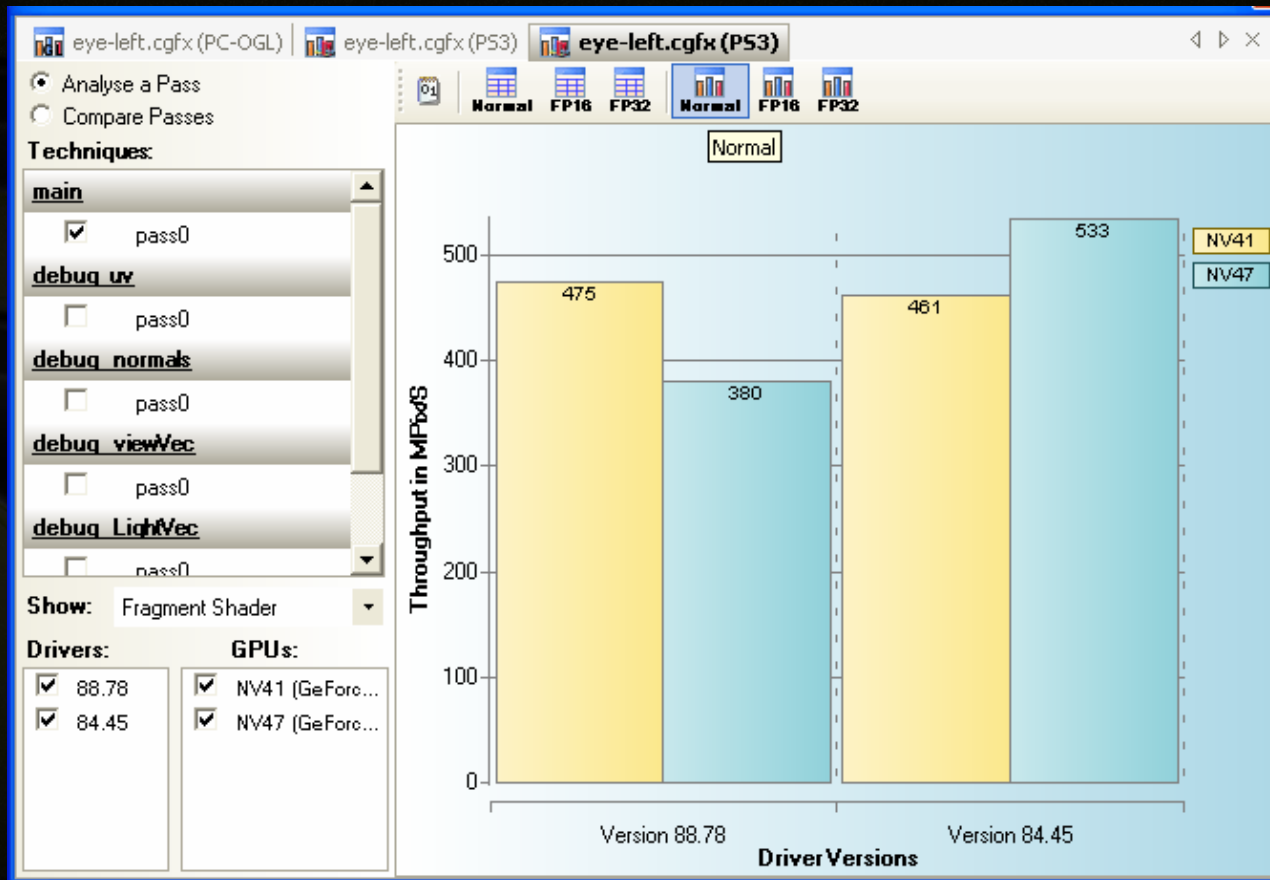


DEMO: Scene Integration

- Light creation
- Scene traversal
- Bind light to material
- Realtime manipulation

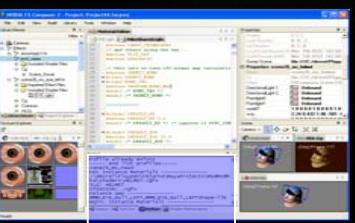


Shader Performance Simulation



The table displays performance metrics for Version 84.45, comparing NV41 and NV47 GPUs. The metrics include MPix/s, Regs., and Cycles.

Version 84.45			
MPix/s	Regs.	Cycles	MPix/s
475	3	16	461
380	3	2	533



Speaker Transition



- Daniel Horowitz

Production Pipeline Integration



- **Scene binding expressions**
- **Automation via scripting**
- **Plug-in based architecture**
- **Source control integration**
- **Production Pipeline Samples**

Scene binding expressions



Semantic

```
float4 LightPos : Position
<
  string Object = "PointLight";
  string Space = "Object";
> = {-10.0f, 10.0f, -10.0f, 0.0f};
```

Annotations



Properties	
Parameter: LightPos	
Name	LightPos
Semantic	POSITION
Rows	1
Columns	4
Value	100 100 100 0

Custom Semantics and Annotations



- Hook parameter to scene and system data
- Expressions via xml configuration file
- Extensive Library of Operators
 - dot & cross products, mux, demux, matrix ops, ...
- Custom operators
 - Built from XML using operators
 - Via plug-in

```
1 :<RemappedSemantic name="myWorldView">
2 :   <MatrixMultiply description="World * View">
3 :     <input type="internalsemantic" value="world"/>
4 :     <input type="internalsemantic" value="view"/>
5 :   </MatrixMultiply>
6 :</RemappedSemantic>
```

Automation Via Scripting



- **Automatic assignment of**
 - **Materials to geometry**
 - **Shader parameters to scene objects**
(nearest lights, cameras, etc...)
 - **A model's accessories to attachment points**

- **Common-tasks toolbar (ala Maya/MEL)**

Automation Via Scripting



```
#Python scripting

# Convert any Possible Profile to COLLADA FX
def ConvertToCOLLADA():
    effects = FXRuntime.Instance.Library.FindLibraryItems(FXEffect)
    for effect in effects:
        for profile in effect.Profiles:
            if profile.CanConvertToColladaFX() == True:
                profile.ConvertToColladaFX()

# Create an effect
def bindMMM():
    CmdGroupBegin.Do("script: assign cgfx files to MMM ")

    SelectRenderPort("OpenGL")
    ForceRedraw()

    CmdGroupEnd.Do()

# get the cgfx files to assign to MMM

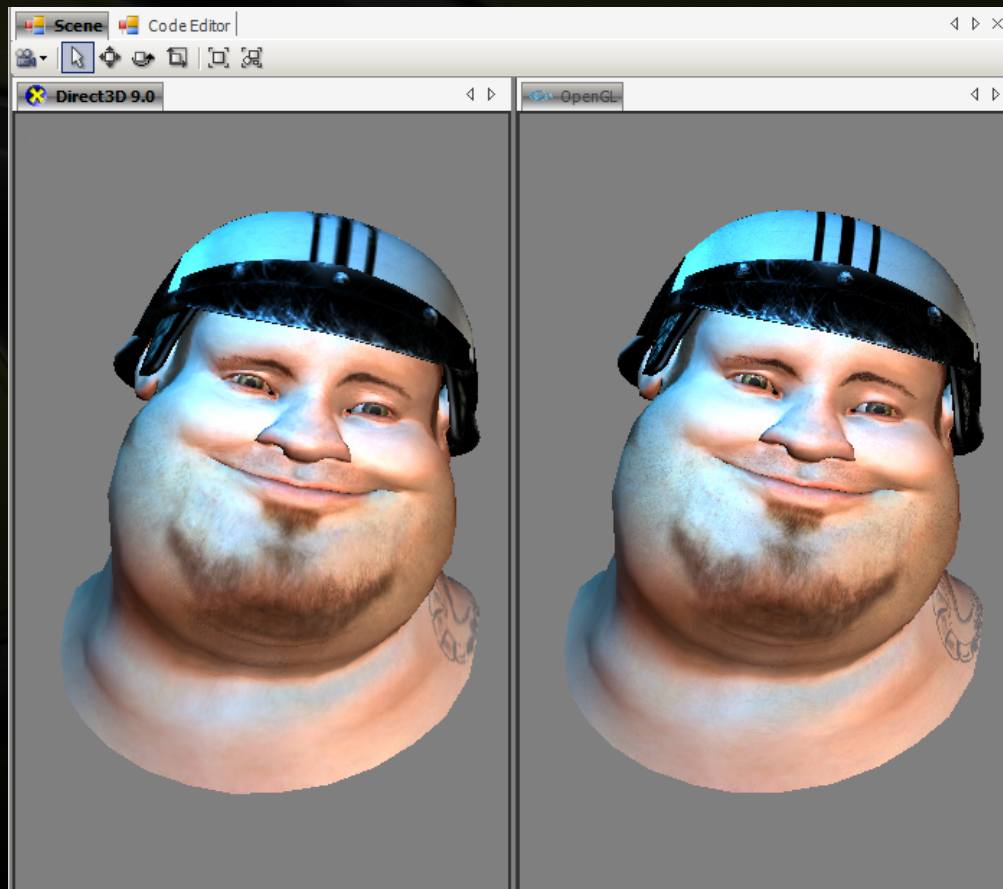
files = FXEffectUtils.GetEffectFiles()
for pathname in files:
```

...

DEMO: Scripting



- Automatic effect loading & assignment



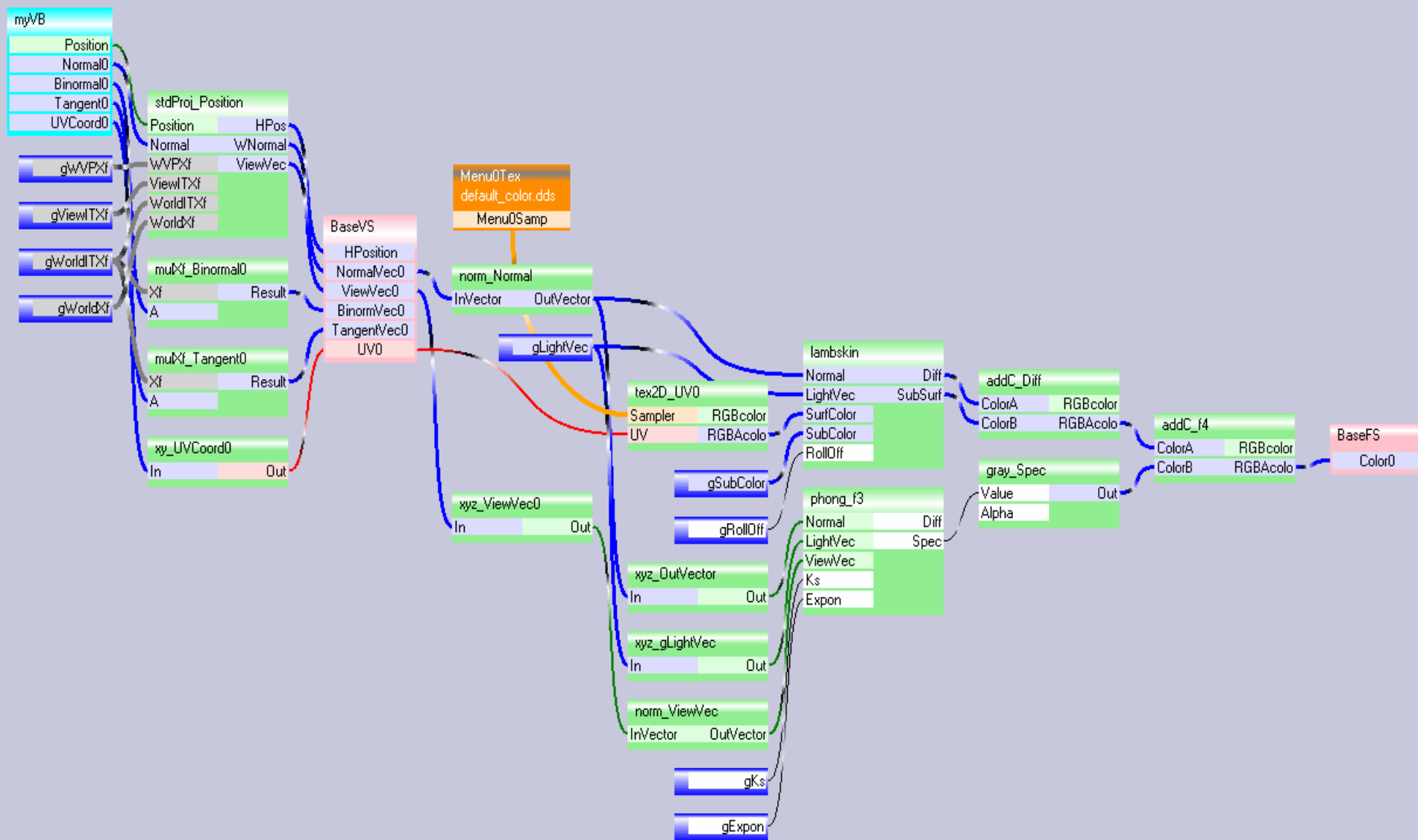
Custom Plug-ins



- Importers
- Exporters
- Semantic expression operators
- Rendering devices
- Procedural geometry generation (fins, hair, etc...)
- Custom authoring environment

...etc...

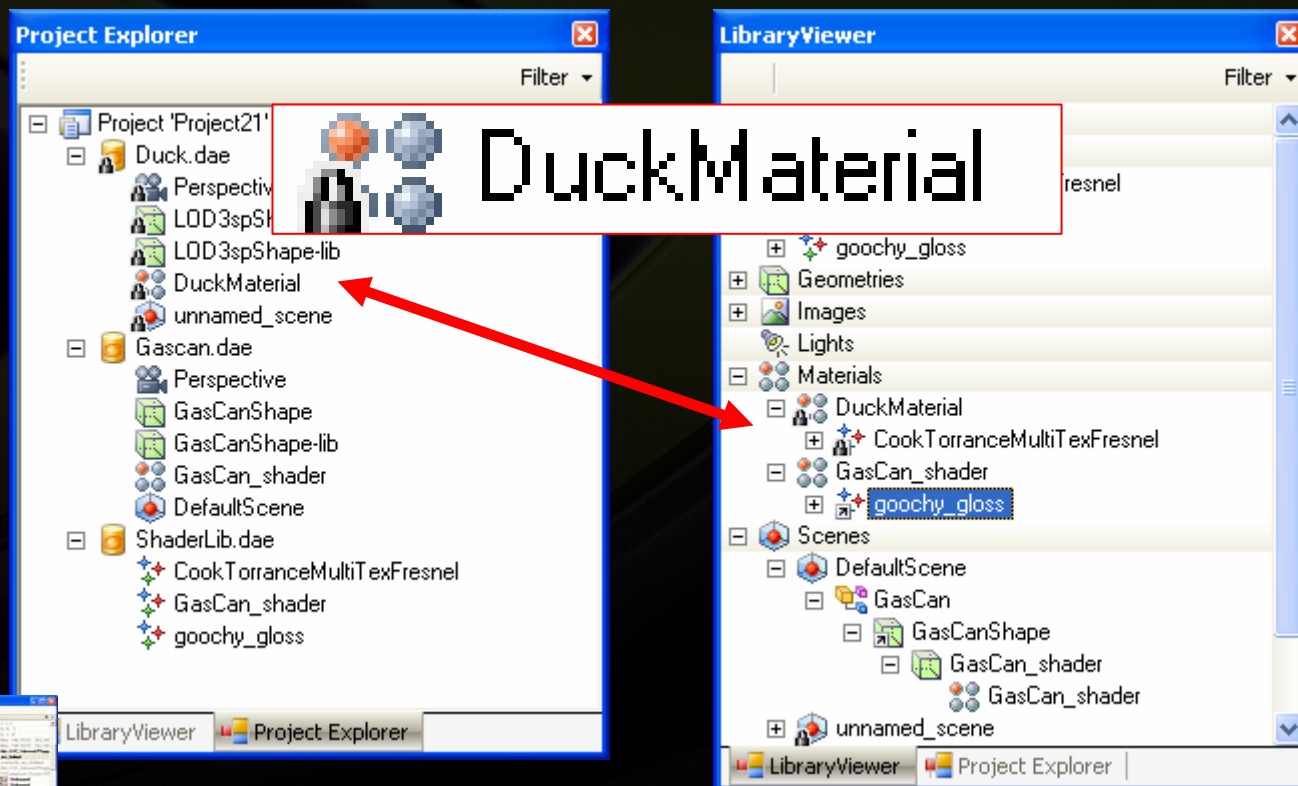
Custom Plug-ins: Sky is the limit



Source Control Integration



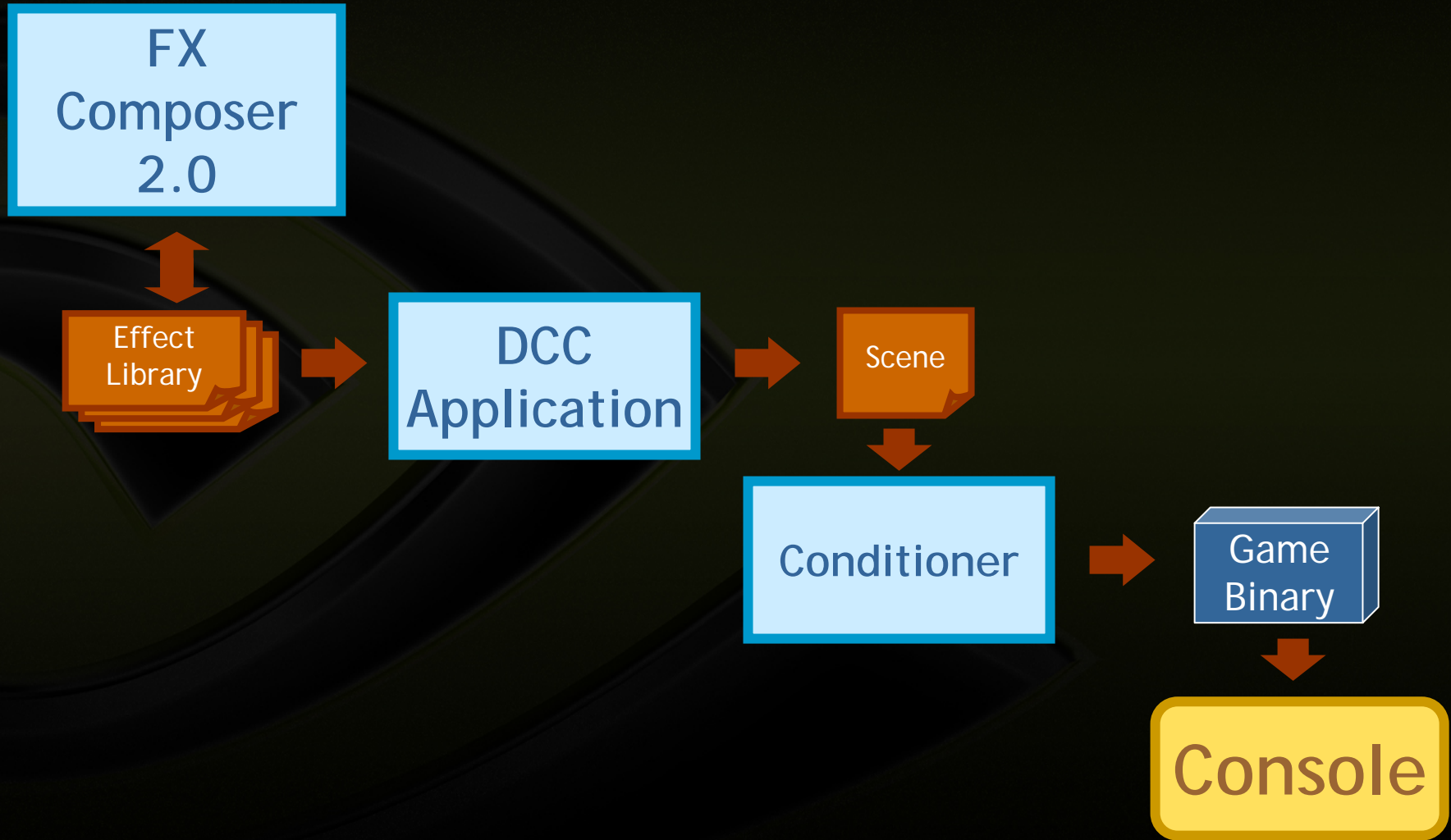
- Seamless integration into source control software
- Documents and assets reflect file-based state



Pipelines: FXC2 centric



Pipelines: DCC centric



Pipelines: Handheld



Host PC

Target Handheld

FX Composer 2.0 ES

Remote
Render

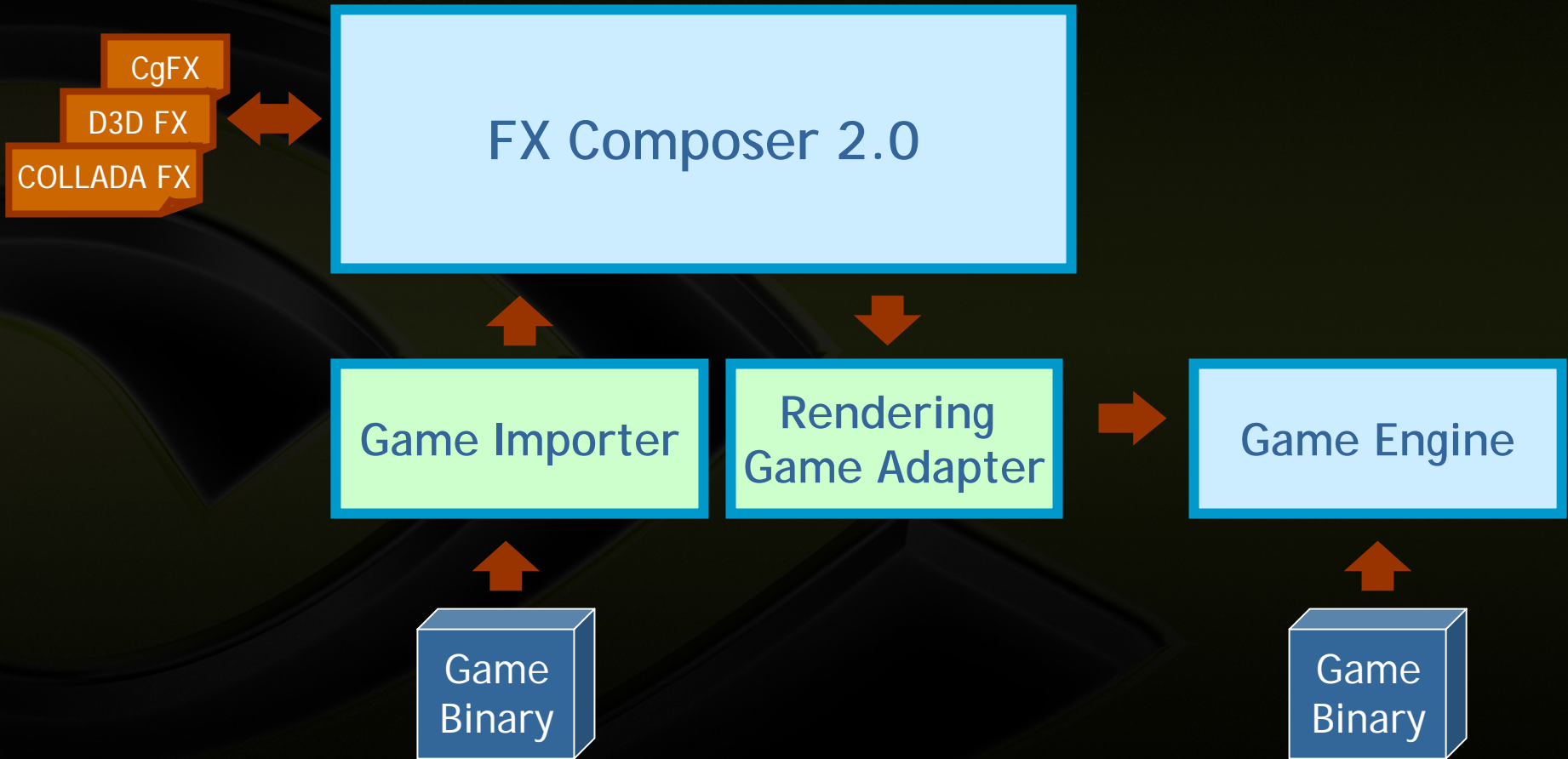
DCC Tools
XSI, Max Maya

COLLADA

Data Conditioner
and Compiler

Game
Engine

Pipelines: Engine



FX Composer 2.0 Alpha 5



- **Alpha5 release ETA end of summer '06**
 - Document and asset management
 - COLLADA FX authoring
 - Shader parameter scene binding
 - Custom semantic and annotation support
 - Python scripting
 - Shader performance
 - Available to Sony PS3 developers and limited partners

- **Beta release ETA end of fall '06**
 - Open to public

Conclusion



- **Next-generation of shader IDE is on its way**
- **Production-ready with powerful features**
- **NVIDIA is closely working with Khronos and others to deliver a professional-grade authoring tool**

Q&A



- Send us emails for early alpha and beta releases

fxcomposer@nvidia.com

- Thanks

Ignacio Castaño (icastano@nvidia.com)

Daniel Horowitz (dhorowitz@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.

nVIDIA

developer.nvidia.com

©2004 NVIDIA Corporation. NVIDIA, and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation. Nalu is ©2004 NVIDIA Corporation. All rights reserved.