Appendix G: Shader Semantics and Annotations

You can specify semantics (on page 2) and annotations (on page 4) for Acrobat 3D Toolkit shader (.fx) files.

Shader Semantics

The following are the supported shader semantics. This list is organized by data type.

Note: Semantics are not case-sensitive.

Semantic	Description	Data Type
lightattenuation0	Defines the constant light attenuation. Light attenuation is a vector of attenuation factors.	float
lightattenuation1	Defines the linear light attenuation. Light attenuation is a vector of attenuation factors.	float
lightattenuation2	Defines the quadratic light attenuation. Light attenuation is a vector of attenuation factors.	float
lightindex	The light that is associated with the current rendering pass.	float
lightfalloff	Defines the intensity that attenuates between the bright inner cone and the outer cone of the light that emits from a spotlight.	float
lightphi	Defines the angle for the outer cone of the light that emits from a spotlight. This is calculated as cos(outer cone angle).	float
lightrange	Defines the distance, in world space, at which meshes in a scene no longer receive light emitted by that object.	float
lighttheta	Defines the radian angle for the inner cone of the light that emits from a spotlight. This is calculated as cos(inner cone angle).	float
lighttype	Defines that light type; 0=Point, 1=Spot, 2=Directional.	float
materialpower, materialspecularpower	Defines the sharpness of the specular highlights.	float
meshradius	Defines the bounding radius of an object, in scene units.	float
time	Defines the time measured in seconds.	float
materialopacity	Defines the material opacity	float
TargetWidth	Defines the active render target width	float
TargetHeight	Defines the active render target height	float
ViewportWidth	Defines the current viewport width	float
ViewportHeight	Defines the current viewport height	float
ViewportX	Defines the current viewport x position (in pixels)	float
ViewportY	Defines the current viewport y position (in pixels)	float
FOV	Defines the camera field of view	float
NearClipPlane	Defines the camera near clip plane	float
FarClipPlane	Defines the camera far clip plane	float
direction	Defines the direction of the light.	float4
materialambient	Defines the color of the ambient light.	float4
materialdiffuse	Defines the diffuse light color.	float4
materialemissive	Defines the 'glow' light color.	float4
materialspecular	Defines the specular light (reflection) color.	float4
position	Defines the position of the light.	float4
lightdirection	Defines the direction of the light.	float4
materiallightambient	Defines the material's ambient reflection characteristics. This is calculated as materialambient * lightambient.	float4
materiallightdiffuse	Defines the material's diffuse reflection characteristics. This is calculated as materialdiffuse * lightdiffuse.	float4
materiallightspecular	Defines the material's specular highlight characteristics. This is calculated as materialspecular * lightspecular.	float4
lightposition	Defines the position of the light.	float4

Semantic	Description	Data Type
lightambient	Defines the light's ambient color.	float4
lightdiffuse	Defines the light's diffuse color.	float4
lightspecular	Defines the light's specular color.	float4
eyeposition	Defines the camera eye position in world space.	float4
lookatposition	Defines the look at position in world space.	float4
updirection	Defines the up direction in world space .	float4
projection	Defines the projection matrix.	matrix
view	Defines the view matrix.	matrix
viewinverse	Defines the view-inverse matrix.	matrix
viewinversetranspose	Defines the view-inverse-transpose matrix.	matrix
viewprojection	Defines the view-projection matrix.	matrix
world	Defines the world matrix.	matrix
worldinverse	Defines the world-inverse matrix.	matrix
worldinversetranspose	Defines the world-inverse-transpose matrix.	matrix
worldtranspose	Defines the world-transpose matrix.	matrix
worldview	Defines the world-view matrix.	matrix
worldviewinverse	Defines the world-view-inverse matrix.	matrix
worldviewinversetranspose	Defines the world-view-inverse-transpose matrix.	matrix
worldviewprojection	Defines the world-view-projection matrix.	matrix
worldviewprojectioninverse	Defines the inverse world view projection matrix	matrix
worldviewprojectioninversetranspose	Defines the transpose of the inverse world view projection matrix	matrix
objectxform	Defines the object transformation matrix	matrix
objectxforminverse	Defines the inverse of the object transformation matrix	matrix
AxisRotation	Defines the rotation component of the view transfomration matrix	matrix

Shader Annotations

The following are the supported shader parameter annotations. This list is organized by data type:

Note: Annotations are not case-sensitive.

Annotation	Description	Data Type
Height	Sets the height of a renderable / procedural texture.	integer
Width	Sets the width of a renderable / procedural texture.	integer
Target	Sets the compile target for a procedurally generated texture.	integer
Depth	Sets the depth of a procedural texture.	integer
MipMap	Sets the depth of a procedural texture	bool
BumpMap	Sets the depth of a procedural texture	bool
UIMax	The maximum value to which the slider variable is set.	float
UIMin	The minimum value to which the slider variable is set.	float
Function	The entry point to a procedurally generated texture.	string
Name	The filename for a texture resource.	string
Type	The type of texture resource: 2D, cube, or volume.	string
UIName	The name that is displayed in the user interface.	string
UIType	The object to use in the user interface: color, slider, or image.	string
UIWidget	The object to use in the user interface: color, slider, or image.	string

The following are the supported shader pass annotations. This list is organized by data type:

Note: Annotations are not case-sensitive.

Annotation	Description	Data Type
CopyColorBuffer	Flag whilch can be used to copy the backbuffer contents into a render target before rendering.	bool
UseExistingDepthBuffer	Flag to indicate use of the existing backbuffer depth buffer, instead of a local depth buffer attached to each render target.	bool
UseExistingViewport	Flag to indicate use of the existing viewport settings (scaled if necessary), instead of full size of a render target.	bool
RenderPassGeometry	Flag to optionally skip any geometry rendering for this pass (normally used to copy backbuffer contents).	bool
UseReverseCullMode	Flag to use the reverse cull render state for pass geometry. Required to maintain behavior when rendering mirror objects, etc.	bool
ClearDepth	Sets the depth clear value tp optionally apply before rendering.	float
ClearColor	Sets the clear color to optionally apply before rendering.	float4
RenderTarget	Sets the name of the local render target to use.	string
RenderObject	Name of an Object type resource (normally a simple .x file) to be used as the rendered geometry for this pass.	string