GX Toolkit Lighting
Version 1.0

Description
The GX Toolkit provides a set of constants, data structures and regular functions that wrapper the functionality of the underlying Direct3D interface. For details of the lighting model and how lighting is calculated refer to the DirectX SDK Documentation for the fixed function pipeline (DirectX 9).

Table 1. The specific constants are listed here.

Constant	Description
gx3d_LIGHT_TYPE_POINT	A point light
gx3d_LIGHT_TYPE_SPOT	A spot light
gx3d_LIGHT_TYPE_DIRECTION	A directional light
gx3d_FILL_MODE_POINT	Render objects vertices only
gx3d_FILL_MODE_WIREFRAME	Render objects edges only
gx3d_FILL_MODE_SMOOTH_SHADED	Render objects with flat-shaded triangles
gx3d_FILL_MODE_GOURAUD_SHADED	Render objects with Gouraud-shaded triangles

Table 2. The specific data structures are listed here.

Data Structure	Description
gx3dLight	Represents a dynamic light (directional, point or spot light)
gx3dLightData	Attributes of a dynamic light
gx3dColor	A RGBA color value where each component value is 0-1
gx3dVector	A XYZ vector

Table 3. The specific functions are listed here.

Function	Description
gx3d_EnableLighting	Enable dynamic lighting
gx3d_DisableLighting	Disable dynamic lighting
gx3d_SetAmbientLight	Set ambient lighting to a gx3dColor
gx3d_EnableSpecularLighting	Enable specular highlights
gx3d_DisableSpecularLighting	Disable specular highlights
gx3d_EnableVertexLighting	Enable per-vertex lighting (part of the 3D model data)
gx3d_DisableVertexLighting	Disable per-vertex lighting
gx3d_InitLight	Create a dynamic light
gx3d_UpdateLight	Update the attributes of an existing dynamic light
gx3d_FreeLight	Destroy a dynamic light
gx3d_EnableLight	Enable a dynamic light
gx3d_DisableLight	Disable a dynamic light
gx3d_SetFillMode	Sets render mode for triangles

Fill Mode
Rendering 3D objects using point or wireframe is generally only useful for debugging or other special purposes. Objects can be rendered using smooth (flat) shading in which case lighting is calculated on a per-triangle basis. This mode is also typically only useful for debugging or in situation where you need to be able to clearly distinguish between different triangles of an object. For most situations Gouraud shading is the standard. In Gouraud shading lighting is calculated on a per-vertex basis with the resulting values interpolated across the surface of each triangle as they are rendered to the screen. Set the desired fill mode using gx3d_SetFillMode() and pass as a parameter one of the constants listed in Table 1.

Ambient lighting
Ambient lighting can be used when dynamic lighting is not required. Ambient lighting can also be used in conjunction with dynamic lights. Ambient lighting can be set with gx3d_SetAmbientLight(). Alternatively, each dynamic light includes an ambient component so ambient lighting can be implicitly enabled when enabling a dynamic light that includes ambient.

Dynamic lighting
Three types of dynamic lights are supported (see Table 1). To enable dynamic light calculations call gx3d_EnableLighting(). To create a dynamic light call gx3d_InitLight() and specify the attributes of the new light in a gx3dLightData structure. The attributes of a dynamic light can be changed at any time by calling gx3d_UpdateLight() and passing it a new gx3dLightData structure. Free the memory used by the light and destroy it by calling gx3d_FreeLight().

A maximum of 8 dynamic lights can be used when rendering any 3D object. This is a limitation of the underlying Direct3D fixed function pipeline. Call gx3d_EnableLight() or gx3d_DisableLight() to enable or disable a dynamic light for use in rendering. Any lights enabled beyond 8 will not affect rendering.

Specular lighting
Specular lighting calculations can be enabled and disabled by calling gx3d_EnableSpecularLighting() or gx3d_DisableSpecularLighting(). The intensity of specular lighting is controlled by setting an appropriate value in the material property of the rendered object (see the gx3dMaterial struct). Recall, that a material must be set before an object can be rendered.

Vertex lighting
If model data contains per-vertex color information it can be used to modify the color value used in lighting calculations. Enable or disable per-vertex colors by calling gx3d_EnableVertexLighting() or gx3d_DisableVertexLighting().