

# SkyCube RT Shader Libs Reference

---

## Introduction:

Skycube RT shader library provide functions to let you to use RGBM and different panoramic cubemap in your shader directly.

Basically, to use RGBM HDR map to your own shader, you just need to copy **SC\_Common\_Libs.cginc** from Skycube RT package to your shader folder. And include this file under **CGPROGRAM** in your shader:

```
#include "SC_Common_Libs.cginc"
```

And make sure you work in D3D9 or higher. You can put following keyword to keep your shader compile for ShaderMode 3.0:

```
#pragma target 3.0
```

Then you can use the SC's lookup functions to replace standard texture sampler functions, like `tex2D()`, `texCUBE()`.

Please note, there are some SC lookup functions need keyword:

```
#pragma glsl
```

## Functions

All library function will have prefix **SC\_**:

```
half4 SC_texCUBEMap(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))
```

Advanced cubemap function for all kinds of panoramic cubemap format as follow:

- **HorizontalCross** : **SC\_HC**
- **VerticalCross** : **SC\_VC**
- **NvidiaCube** : **SC\_NV**
- **XSIcube** : **SC\_XSI**
- **Spherical** : **SC\_SP**
- **LightProbe** : **SC\_LP**

To use this function you need defined the cubemap type inside your shader first, like **#define SC\_HC** for Horizontal Cross.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMap\_HC(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))**

The function Horizontal Cross panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMap\_VC(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))**

The function Vertical Cross panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMap\_NV(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))**

The function Nvidia panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMap\_XSI(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))**

The function XSI panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMap\_SP(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))**

The function Spherical panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMap\_LP(sampler2D tex, half3 dir, half4 adj = half4(1,0,1,0))**

The function Light Probe panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_tex2DRGBM(sampler2D tex, half2 uv, half range, half ev = 1)**

Tex2D RGBM Decode function. It decode RGBM to float HDR Color.

**sampler2D tex:** 2D RGBM encoding texture

**half2 uv:** Texture UV

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**Return:** HDR Color

**half4 SC\_texCUBERGBM(samplerCUBE cube, half3 dir, half range, half ev = 1)**

TexCUBE RGBM Decode function. It decode RGBM cubemap to float HDR Color cubemap.

**samplerCUBE cube:** RGBM encoding cubemap texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**Return:** HDR Color

**half4 SC\_texCUBERGBM(samplerCUBE cube, half3 dir, half range, half ev , int mplevel)**

TexCUBE RGBM Decode function with mipmap level (Specular chain). It decode RGBM cubemap to float HDR Color cubemap in specific mipmap level.

**samplerCUBE cube:** RGBM encoding cubemap texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**int mplevel:** Mipmap level

**Return:** HDR Color

**half4 SC\_texCUBEMapRGBM(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

Advanced RGBM cubemap function for all kinds of RGBM encoding panoramic cubemap format as follow:

- **HorizontalCross : SC\_HC**
- **VerticalCross : SC\_VC**
- **NvidiaCube : SC\_NV**
- **XSIcube : SC\_XSI**
- **Spherical : SC\_SP**
- **LightProbe : SC\_LP**

To use this function you need defined the cubemap type inside your shader first, like **#define SC\_HC** for Horizontal Cross.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMapRGBM\_HC(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

The function Horizontal Cross RGBM encoding panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMapRGBM\_VC(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

The function Vertical CrossRGBM encoding panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMapRGBM\_NV(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

The function Nvidia Cube RGBM encoding panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMapRGBM\_XSI(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

The function XSI Cube RGBM encoding panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMapRGBM\_SP(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

The function Spherical RGBM encoding panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

**half4 SC\_texCUBEMapRGBM\_LP(sampler2D tex, half3 dir, half range, half ev = 1, half4 adj = half4(1,0,1,0))**

The function LightProbe RGBM encoding panoramic cubemap format.

**sampler2D tex:** 2D panoramic texture

**half3 dir:** Reflection Vector

**half range:** The Max HDR range which setup when encoding.

**half ev:** Exposure value

**half4 adj:** Offset adjustment for solve edge seam, Vector4( U scale, U offset, V scale, V offset )

**Return:** Color

Any question about this tool please contact me in mail: [support@atomsdev.com](mailto:support@atomsdev.com)

[Copyright 2015 SkyCube\\_atomsdev.wordpress.com](http://Copyright 2015 SkyCube_atomsdev.wordpress.com)