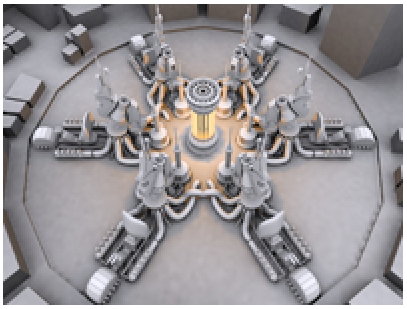
# DepthOfField10.1 Sample

⊟ Collapse All



## Path

| | |
|---|---|
| **Source** | *SDK root*\Samples\C++\Direct3D10\DepthOfField10.1 |
| **Executable** | *SDK root*\Samples\C++\Direct3D10\Bin\*x86 or x64*\DepthOfField10.1.exe |

## Overview

This sample, contributed by AMD, presents a technique for combining multi-sample anti-aliasing (MSAA) with a depth-of-field effect. It utilizes Direct3D 10.1 APIs and hardware to retrieve depth values from a multi-sampled back buffer.

## Depth of Field

To create the depth-of-field effect, the scene is first rendered into an off-screen render target; we'll call this the *color buffer*. In a second pass, only depth values from the scene are rendered into a floating-point buffer. **ID3D10Device::GenerateMips** is used to generate a full mip-chain for the color buffer. In a final pass, a full-screen quad is rendered. For each texel in the quad, the associated depth value is sampled from the depth value. A different mip-level is selected from the color buffer, depending on the distance from this depth value to the focal plane. Further values will sample lower mip-levels. Therefore, samples further from the focal plane will be blurred.

## Using MSAA

Older shader models were unable to sample from multi-sampled render targets. Therefore, when performing a technique that accessed a multi-sampled render target, MSAA either needed to be disabled, or the render target needed to be explicitly resolved (Direct3D 10). Shader model 4.1, the shader model used by Direct3D 10.1, allows shaders not only to sample MSAA render targets, but also to select which of the individual multi-samples to grab. To do this, the resource in the shader must first be declared as multi-sampled. In this case, it is a 2D multi-sampled resource, so it is declared as Texture2DMS. The <float,1> afterwards signifies that each sample contains 1 float value.

```
Texture2DMS<float,1>            g_txDepthMSAA;
```

Sampling a multi-sampled resource is similar to sampling a non-multi-sampled resource. However, a third texture coordinate is required to select which multi-sample to fetch.

```
fDepth = g_txDepthMSAA.Load( int3( iScreenCoord, 0 ), 0 );
```

On Direct3D 10.1-enabled hardware, this sample will enable MSAA and use loads from a mult-sampled render target to fetch the depth values that are needed to create the depth of field effect. On Direct3D 10 hardware, MSAA will be disabled.

Send feedback to DxSdkDoc@microsoft.com.
Version: 1962.00