

SimpleCompute Sample for Direct3D 11.X

*This sample is compatible with the March 2016 Xbox One XDK or later*

# Description



SimpleCompute shows how to use DirectCompute™ (i.e. Direct3D Compute Shader). In addition to ‘stock’ Direct3D 11 usage, this sample also shows how to use the **ID3D11ComputeContextX** interface to submit asynchronous compute shader workloads. It updates a texture by computing the Mandelbrot set using a compute shader.

# Using the sample

|  |  |
| --- | --- |
| Action | Gamepad |
| Toggle Async Compute | A button |
| Reset Viewport to Default | Y button |
| Pan Viewport | Left stick |
| Zoom Viewport | Right stick |
| Increase Zoom Speed | Right trigger |
| Exit | View Button |
| Menu | Show/hide help |

# Implementation notes

The primary purpose of this sample is to familiarize the reader with creating and using a simple compute shader.

* **CreateDeviceDependentResources**: This is where the compiled compute shader is loaded and the various Direct3D rendering resources are created. The shaders are compiled by Visual Studio.
* **Render**: If the sample is not using asynchronous compute the compute shader is dispatched before the draw call that needs the results is dispatched. This updates the texture every frame.
* **AsyncComputeProc**: If the sample is using asynchronous compute the compute shader is dispatched from this thread as soon as it’s told to start processing. Render will wait until it’s told the asynchronous task is complete before performing the dependent draw call.

# Update history

Initial release June 2017. Updated September 2017

# Privacy Statement

When compiling and running a sample, the file name of the sample executable will be sent to Microsoft to help track sample usage. To opt-out of this data collection, you can remove the block of code in Main.cpp labeled “Sample Usage Telemetry”.

For more information about Microsoft’s privacy policies in general, see the [Microsoft Privacy Statement](https://privacy.microsoft.com/en-us/privacystatement/).