Simple MSAA for DirectX 12

# *This sample is compatible with the Windows 10 April 2018 Update SDK (17134)*



# Description

This sample implements an MSAA render target & depth/stencil buffer for a 3D scene using DirectX 12.

# Using the sample

|  |  |  |
| --- | --- | --- |
| Action | Gamepad | Keyboard |
| Toggle MSAA vs. single-sample | A button | Space |
| Exit | View Button | Esc |

# Implementation notes

The UI is drawn without MSAA, and makes use of an explicit resolve. Note that DirectX12 does not support using an implicit resolve of an MSAA swap chain as was the case with ‘classic’ DirectX 11.

## UWP on Xbox

To support DirectX 12 on Xbox One, a UWP app must have its app type set to ‘Game’. Otherwise, only the software device (WARP12) is available on developer consoles, which is unsupported for retail consoles. During development the app type can be set via DevHome. This sample uses the [expandedResources](https://msdn.microsoft.com/en-us/library/windows/desktop/mt808808.aspx) restricted capability to achieve this by default, but can be removed after setting the package app type to ‘Game’. Note that apps submitted to the Windows Store will fail validation if using this restricted capability.

# Known issues

* Due to a bug in the Windows 10 validation layer prior to the Windows 10 Creators Update (15063), a DirectX 12 Resolve with an sRGB format using new “flip-style” swapchain would fail. This has been fixed in the newer versions of Windows 10.

# Update history

Initial release May 2017. Updated October 2017 for UWP on Xbox One.

# 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/).