Simple Triangle Sample (DX12, C++/WinRT)

*This sample is compatible with the March 2017 Xbox One XDK or later*

# Description

This sample demonstrates how to create a static Direct3D vertex buffer to render a triangle on screen.



# Using the sample

The sample has no controls other than exiting.

# Implementation notes

The primary purpose of this sample is to familiarize the reader with the ATG samples template structure, as well as provide a simple demonstration of using Direct3D 12 APIs.

**CreateDeviceDependentResources**: This is where the compiled vertex and pixel shaders blobs are loaded and the various Direct3D rendering resources are created. *The shaders are compiled by Visual Studio.*

**Render:** This is where the triangle is rendered and presented to the screen.

For details on device creation and presentation handling, see [DeviceResources](https://github.com/Microsoft/DirectXTK12/wiki/DeviceResources).

For details on the use of the loop timer, see [StepTimer](https://github.com/Microsoft/DirectXTK/wiki/StepTimer).

*This sample is set up to use Visual Studio 2017 (15.3 update). C++/WinRT is supported with VS 2015 Update 3, but VS 2017 (15.3 update) or later is strongly recommended.*

# Update history

C++/WinRT version of SimpleTriangle12 was released in March 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/).