VideoMemory

<https://github.com/walbourn/directx-sdk-samples>

This is the DirectX SDK's VideoMemory sample updated to use Visual Studio 2012 and the Windows SDK 8.0 without any dependencies on legacy DirectX SDK content.

Applications often need to know how much video memory is available on the system. This is used to scale content to ensure that it fits in the allotted video memory without having to page from disk. This sample demonstrates the various methods to obtain video memory sizes which works on a wide range of Windows OS releases.



This sample is a Win32 desktop sample to cover a wide range of Windows OS release. Windows Store apps can utilize the DXGI API to obtain reliable video memory information.

# Video Memory Classifications

Video memory can fall into one of two categories. The first is dedicated video memory. Dedicated video memory is available for exclusive use by the video hardware. Dedicated video memory typically has very fast communication with the video hardware. It can also be referred to as on-board or local video memory, because such memory is often found on a video card. However, video hardware that is integrated into a motherboard often does not have dedicated video memory.

The second category of video memory is shared system memory. Shared system memory is part of the main system memory. The video hardware can use this memory in the same way as it would dedicated video memory, but that memory is also used by the operating system and other applications. For discrete video cards that use system memory, this can also be called non-local video memory. Integrated video cards often only have shared system memory. Shared system memory usually has slower communication with video hardware than dedicated video memory does.

# Ways to Get Video Memory

There are several ways to get the size of video memory on a system. This sample demonstrates 5 methods. The first 4 are available on Windows XP, Windows Vista, Windows 7, and Windows 8.x. DirectX Graphics Infrastructure (DXGI) is only available on Windows Vista, Windows 7, and Windows 8.x. Those methods are:

## GetVideoMemoryViaDirectDraw

This method queries the DirectDraw 7 interfaces for the amount of available video memory. On a discrete video card, this is often close to the amount of dedicated video memory and usually does not take into account the amount of shared system memory.

## GetVideoMemoryViaWMI

This method queries the Windows Management Instrumentation (WMI) interfaces to determine the amount of video memory. On a discrete video card, this is often close to the amount of dedicated video memory and usually does not take into account the amount of shared system memory.

## GetVideoMemoryViaDxDiag

DxDiag internally uses both DirectDraw 7, and WMI and returns the rounded WMI value, if WMI is available. Otherwise, it returns a rounded from DirectDraw 7.

Note that DxDiag is supported on Windows XP, but the required headers are not present in the Windows SDK 7.1 used for "v110\_xp" Platform Toolset builds. Therefore, this sample doesn't implement the DxgDiag version for the Windows XP compataible configurations. It is possible to build this for Windows XP by combing with the legacy DirectX SDK.

## GetVideoMemoryViaD3D9

This method queries DirectX 3D 9 for the amount of available texture memory. On Windows Vista or later, this number is typically the dedicated video memory, plus the shared system memory, minus the amount of memory in use by textures and render targets.

## GetVideoMemoryViaDXGI

DXGI is not available on Windows XP, but is available on Windows Vista or later. This method returns the amount of dedicated video memory, the amount of dedicated system memory, and the amount of shared system memory. DXGI provides numbers that more accurately reflect the true system configuration than the previous 4 methods in this list.

# More Information

[Where is the DirectX SDK?](http://blogs.msdn.com/b/chuckw/archive/2012/03/22/where-is-the-directx-sdk.aspx)

[Where is the DirectX SDK (2013 Edition)?](http://blogs.msdn.com/b/chuckw/archive/2013/07/01/where-is-the-directx-sdk-2013-edition.aspx)

[Games for Windows and DirectX SDK blog](http://blogs.msdn.com/b/chuckw/)

[Visual Studio 2012 Update 1](http://blogs.msdn.com/b/chuckw/archive/2012/11/26/visual-studio-2012-update-1.aspx)