

DirectX 9.0 SDK and the DirectX Utility Headers

The DirectX 9.0 SDK

The industry standard 3D API, OpenGL, is not the only 3D API that you can use on the .NET platform in the Microsoft Windows environment. Another choice is DirectX, which has been steadily increasing its functionality, stability, and market share among developers writing all sorts of Windows multimedia software, games, and other applications that make use of 2D and 3D graphics. While earlier versions of DirectX often left a lot to be desired, its latest incarnation, DirectX 9.0, is a serious product worth a longer look, especially if you want to target multiple Microsoft platforms: Microsoft Windows, PocketPC, or X-Box.

With DirectX 9.0, you have a choice of two 3D APIs on the Microsoft Windows platform, but that doesn't mean that it must be one or the other. Even if you stay with the OpenGL for 3D API (it's an industry standard after all, and there's a lot of information and talent to draw upon) you may still need DirectX's other components that you will simply not find in OpenGL. Direct3DX is only a part of the DirectX puzzle, and other pieces are equally interesting.

Just like OpenGL, DirectX too used to be aimed at developers coding in C/C++. That's natural, because compiled C/C++ code is faster than interpreted Visual Basic. However, Microsoft knows that in order to make .NET a success, they need to attract as many developers as they can. To do that, all of their crucial APIs must be available in managed form, so they can be easily incorporated into applications written in C#, Visual Basic .NET, and other .NET languages.

*Definition: **Managed DirectX 9.0**, the .NET interface for using DirectX 9.0 with .NET languages*

The recently released DirectX 9.0 SDK is a clear sign that Microsoft wants to broaden the use of DirectX among developers who do not speak C++ and MFC programmers. The official release of the DirectX 9.0 SDK is available in three forms:

DirectX 9.0 SDK: The whole package for C/C++, C#, Visual Basic .NET, and other languages available on the .NET platform.

DirectX 9.0 SDK for C#: A cut-down, managed version of the DirectX 9.0 SDK for programmers writing C# code.

DirectX 9.0 SDK for Visual Basic .NET: A cut-down, managed version of the DirectX 9.0 SDK for programmers writing Visual Basic .NET code.

Despite their best efforts to provide managed versions of the whole of DirectX, Microsoft does not yet provide managed versions of DirectMusic, DirectShow, or DirectSetup. One additional component that you will find is a basic support for audio and video playback.

Check Out the DXSDK code samples: DXSDK/Samples/C++/

The samples are available for other languages as well.

DirectX Utility Headers

The DirectX Utility Toolkit or DXUtil as it is known is a set of helper functions and typing shortcuts for DirectX programming.

It is made up of a header file (dxutil.h) and a source file (dxutil.cpp). You can view these files at:

http://if.dynsite.net/t-pot/program/29_appwizard/dxutil_cpp.html

http://if.dynsite.net/t-pot/program/29_appwizard/dxutil_h.html

Or you can download them from my site.

[1] Primitive - A low-level object or operation from which higher-level, more complex objects and operations can be constructed. In graphics, primitives are basic elements, such as lines, curves, and polygons, which you can combine to create more complex graphical images. In programming, primitives are the basic operations supported by the programming language. A programmer combines these primitives to create more complex operations, which are packaged as functions, procedures, and methods.