

Ever wondered just what that enigmatic name means?

Gaming and multimedia applications are some of the most satisfying programs you can get for your PC, but getting them to run properly isn't always as easy as it could be. First, the PC architecture was never designed as a gaming platform. Second, the wide-ranging nature of the PC means that one person's machine can be different from another. While games consoles all contain the same hardware, PCs don't: the massive range of difference can make gaming a headache.

To alleviate as much of the pain as possible, Microsoft needed to introduce a common standard which all games and multimedia applications could follow – a common interface between the OS and whatever hardware is installed in the PC, if you like. This common interface is DirectX, something which can be the source of much confusion.

DirectX is an interface designed to make certain programming tasks much easier, for both the game developer and the rest of us who just want to sit down and play the latest blockbuster. Before we can explain what DirectX is and how it works though, we need a little history lesson.

DirectX history

Any game needs to perform certain tasks again and again. It needs to watch for your input from mouse, joystick or keyboard, and it needs to be able to display screen images and play sounds or music. That's pretty much any game at the most simplistic level.

Imagine how incredibly complex this was for programmers developing on the early pre-Windows PC architecture, then. Each programmer needed to develop their own way of reading the keyboard or detecting whether a joystick was even attached, let alone being used to play the game. Specific routines were needed even to display the simplest of images on the screen or play a simple sound.

Essentially, the game programmers were talking directly to your PC's hardware at a fundamental level. When Microsoft introduced Windows, it was imperative for the stability and success of the PC platform that things were made easier for both the developer and the player. After all, who would bother writing games for a machine when they had to reinvent the wheel every time they began work on a new game? Microsoft's idea was simple: stop programmers talking directly to the hardware, and build a common toolkit which they could use instead. DirectX was born.

How it works

At the most basic level, DirectX is an interface between the hardware in your PC and Windows itself, part of the Windows API or Application Programming Interface. Let's look at a practical example. When a game developer wants to play a sound file, it's simply a case of using the correct library function. When the game runs, this calls the DirectX API, which in turn plays the sound file. The developer doesn't need to know what type of sound card he's dealing with, what it's capable of, or how to talk to it. Microsoft has provided DirectX, and the sound card manufacturer has provided a DirectX-capable driver. He asks for the sound to be played, and it is – whichever machine it runs on.

From our point of view as gamers, DirectX also makes things incredibly easy – at least in theory. You install a new sound card in place of your old one, and it comes with a DirectX driver. Next time you play your favourite game you can still hear sounds and music, and you haven't had to make any complex configuration

changes.

Originally, DirectX began life as a simple toolkit: early hardware was limited and only the most basic graphical functions were required. As hardware and software has evolved in complexity, so has DirectX. It's now much more than a graphical toolkit, and the term has come to encompass a massive selection of routines which deal with all sorts of hardware communication. For example, the DirectInput routines can deal with all sorts of input devices, from simple two-button mice to complex flight joysticks. Other parts include DirectSound for audio devices and DirectPlay provides a toolkit for online or multiplayer gaming.

#### DirectX versions

The current version of DirectX at time of writing is DirectX 9.0. This runs on all versions of Windows from Windows 98 up to and including Windows Server 2003 along with every revision in between. It doesn't run on Windows 95 though: if you have a machine with Windows 95 installed, you're stuck with the older and less capable 8.0a. Windows NT 4 also requires a specific version – in this case, it's DirectX 3.0a.

With so many versions of DirectX available over the years, it becomes difficult to keep track of which version you need. In all but the most rare cases, all versions of DirectX are backwardly compatible – games which say they require DirectX 7 will happily run with more recent versions, but not with older copies. Many current titles explicitly state that they require DirectX 9, and won't run without the latest version installed. This is because they make use of new features introduced with this version, although it has been known for lazy developers to specify the very latest version as a requirement when the game in question doesn't use any of the new enhancements. Generally speaking though, if a title is version locked like this, you will need to upgrade before you can play. Improvements to the core DirectX code mean you may even see improvements in many titles when you upgrade to the latest build of DirectX. Downloading and installing DirectX need not be complex, either.

#### Upgrading DirectX

All available versions of Windows come with DirectX in one form or another as a core system component which cannot be removed, so you should always have at least a basic implementation of the system installed on your PC. However, many new games require the very latest version before they work properly, or even at all.

Generally, the best place to install the latest version of DirectX from is the dedicated section of the Microsoft Web site, which is found at [www.microsoft.com/windows/directx](http://www.microsoft.com/windows/directx). As we went to press, the most recent build available for general download was DirectX 9.0b. You can download either a simple installer which will in turn download the components your system requires as it installs, or download the complete distribution package in one go for later offline installation.

Another good source for DirectX is games themselves. If a game requires a specific version, it'll be on the installation CD and may even be installed automatically by the game's installer itself. You won't find it on magazine cover discs though, thanks to Microsoft's licensing terms.

#### Diagnosing problems

Diagnosing problems with a DirectX installation can be problematic, especially if you don't know which one of the many components is causing your newly purchased game to fall over. Thankfully, Microsoft provides a useful utility called the DirectX Diagnostic Tool, although this isn't made obvious. You won't find this tool in the Start Menu with any version of Windows, and each tends to install it in a different place.

The easiest way to use it is to open the Start Menu's Run dialog, type in dxdiag and then click OK. When the application first loads, it takes a few seconds to interrogate your DirectX installation and find any problems. First, the DirectX Files tab displays version information on each one of the files your installation uses. The Notes section at the bottom is worth checking, as missing or corrupted files will be flagged here.

The tabs marked Display, Sound, Music, Input and Network all relate to specific areas of DirectX, and all but

the Input tab provide tools to test the correct functioning on your hardware. Finally, the More Help tab provides a useful way to start the DirectX Troubleshooter, Microsoft's simple linear problem solving tool for many common DirectX issues.