

Teaching Myself DirectX

The Device

A device is used to create resources and to enumerate the capabilities of a display adapter. A Direct3D device allocates and destroys objects, **renders primitives**, and **communicates with a graphics driver and the hardware**. In Direct3D 11, a device is separated into a device object for creating resources and a device-context object, which performs rendering. A device is represented with an [**ID3D11Device**](https://docs.microsoft.com/en-us/windows/desktop/api/D3D11/nn-d3d11-id3d11device) interface.

**Each application must have at least one device.**

[**D3D11CreateDevice**](https://docs.microsoft.com/en-us/windows/desktop/api/D3D11/nf-d3d11-d3d11createdevice) or [**D3D11CreateDeviceAndSwapChain**](https://docs.microsoft.com/en-us/windows/desktop/api/D3D11/nf-d3d11-d3d11createdeviceandswapchain) can be used to create a device.

Device Context

A device context is used to set **pipeline** state and generate rendering commands using the resources owned by a device.

Swap Chain

A swap chain is a collection of buffers that are used for displaying frames to the user. Each time an application presents a new frame for display, the first buffer in the swap chain takes the place of the displayed buffer. This process is called swapping or flipping.

A graphics adapter holds a **pointer** to a surface that represents the image being displayed on the monitor, called a **front buffer**. As the monitor is refreshed, the graphics card sends the contents of the front buffer to the monitor to be displayed. However, this leads to a problem when rendering real-time graphics. The heart of the problem is that monitor refresh rates are very slow in comparison to the rest of the computer. Common refresh rates range from 60 Hz (60 times per second) to 100 Hz. If your application is updating the front buffer while the monitor is in the middle of a refresh, the image that is displayed will be cut in half with the upper half of the display containing the old image and the lower half containing the new image. **This problem is referred to as tearing**.

Back buffering is the process of drawing a scene to an off-screen surface, called a back buffer. Note that any surface other than the front buffer is called an **off-screen surface because it is never directly viewed by the monitor**. By using a back buffer, an application has the freedom to render a scene whenever the system is idle (that is, no windows messages are waiting) without having to consider the monitor's refresh rate. Back buffering brings in an additional complication of how and when to move the back buffer to the front buffer.