DxCreateRenderTarget

This function creates a render target element, which is a special type of texture that can be drawn on with the dx functions. Successful render target creation is not guaranteed, and may fail due to hardware or memory limitations.

To see if creation is likely to fail, use dxGetStatus. (When **VideoMemoryFreeForNRP** is zero, failure *is* guaranteed.)

> **Tip:** Use dxSetBlendMode to get better quality

> **Tip:** It is highly recommended that dxSetTestMode is used when writing and testing scripts using dxCreateRenderTarget.

> **Note:** Render targets are usually cleared when the player minimizes NRP (i.e. alt-tab). See onClientRestore for details on when to restore any fixed content.

Syntax

```
element dxCreateRenderTarget ( int width, int height [, bool withAlpha = false ] )
```

BETA: NEW FEATURE (BUILD: 1.6.0 r21938)

```
element dxCreateRenderTarget ( int width, int height, surface-format surfaceFormat )
```

OOP Syntax Help! I don't understand this!

Method: *DxRenderTarget(...)*

Required Arguments

- **width :** The width of the texture in pixels.
- **height :** The height of the texture in pixels.
- **withAlpha:** The render target will be created with an alpha channel. 'false' will turn images' alpha channels to black color

BETA: NEW FEATURE (BUILD: 1.6.0 r21938)

- **surfaceFormat :** See Surface formats

Returns

Returns a texture element if successful, *false* if the system is unable to create a render target.

You should always check to see if this function has returned false.

Explanation

What is a render target?

A render target is like a blank canvas. You can draw on the render target as many times as you like - and even clear it.

If your dxDraw* calls are static (meaning the appearance doesn't change), or only update periodically, then a render target can be useful not only for cleaner code - but for performance reasons too. Instead of making possibly hundreds of dxDraw* calls every frame, you can simply make those calls on a single frame and draw directly to the render target, then use a **single** dxDrawImage call every frame afterwards to display the render target.

Render targets can also be used to create and display the same thing multiple times, as shown in the example below.