Client-side function

DxCreateShader

This function creates a shader element that can be used in the dxDraw functions. Successful shader creation is not guaranteed unless the Effect File contains a fallback technique which will work on every existing PC.

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

BEFORE VERSION 1.5.8 r20688:

Syntax

```
element, string dxCreateShader ( string filepath / string raw_data [, float priority = 0, float maxDistance = 0, bool layere
d = false, string elementTypes = "world,ped,vehicle,object,other,all" ] )
```

OOP Syntax Help! I don't understand this!

Method: *DxShader(...)*

Required Arguments

- **filepath / raw_data:** The filepath of the shader Effect File (.fx) file or whole data buffer of the shader file

Optional Arguments

All the following optional arguments are only relevant when the shader is used with engineApplyShaderToWorldTexture

- **priority:** If more than one shader is matched to a world texture, the shader with the highest priority will be used. If there is more than one shader with the same highest priority, the most recently created shader is used.
- **maxDistance:** If non-zero, the shader will be applied to textures nearer than maxDistance only. This can speed up rendering, but (to look good) may require the shader to fade out it's own effect as the texture reaches maxDistance.
- **layered:** When set to true, the shader will be drawn in a separate render pass. Several layered shaders can be drawn on the same world texture. (To avoid Z fighting artifacts, you may have to add **DepthBias=-0.0002;** to the technique pass, but this might cause visual artifacts when applied on vehicles)
- **elementTypes:** A comma seperated list of element types to restrict this shader to. Valid element types are:
  - world - Textures in the GTA world
  - ped - Player and ped textures
  - vehicle - Vehicles textures
  - object - Objects textures
  - other - Element textures which are not peds, vehicles or objects
  - all - Everything

Returns

- **element:** A shader element if successful, *false* if invalid arguments were passed to the function. **You should always check to see if this function has returned false.**
- **string:** The name of the technique that will be used.

Syntax

```
element, string dxCreateShader ( string filepath / string raw_data [ [, table macros = {} ], float priority = 0, float maxDis
tance = 0, bool layered = false, string elementTypes = "world,ped,vehicle,object,other,all" ] )
```

OOP Syntax Help! I don't understand this!

Method: *DxShader(...)*

Required Arguments

- **filepath / raw_data:** The filepath of the shader Effect File (.fx) file or whole data buffer of the shader file

Optional Arguments

All the following optional arguments are only relevant when the shader is used with engineApplyShaderToWorldTexture

- **macros:** A table contains macros in an ordered and/or unordered way. See example below.
- **priority:** If more than one shader is matched to a world texture, the shader with the highest priority will be used. If there is more than one shader with the same highest priority, the most recently created shader is used.
- **maxDistance:** If non-zero, the shader will be applied to textures nearer than maxDistance only. This can speed up rendering, but (to look good) may require the shader to fade out it's own effect as the texture reaches maxDistance.
- **layered:** When set to true, the shader will be drawn in a separate render pass. Several layered shaders can be drawn on the same world texture. (To avoid Z fighting artifacts, you may have to add **DepthBias=-0.0002;** to the technique pass, but this might cause visual artifacts when applied on vehicles)
- **elementTypes:** A comma seperated list of element types to restrict this shader to. Valid element types are:
  - world - Textures in the GTA world
  - ped - Player and ped textures
  - vehicle - Vehicles textures
  - object - Objects textures
  - other - Element textures which are not peds, vehicles or objects
  - all - Everything

Returns

- **element:** A shader element if successful, *false* if invalid arguments were passed to the function. **You should always check to see if this function has returned false.**
- **string:** The name of the technique that will be used.