

DxDrawMaterialPrimitive

This function draws a 2D primitive shape with material applied to it across the screen - rendered for one frame. This should be used in conjunction with `onClientRender` in order to display continuously. If image file is used, it should ideally have dimensions that are a power of two, to prevent possible blurring. Power of two: 2px, 4px, 8px, 16px, 32px, 64px, 128px, 256px, 512px, 1024px...

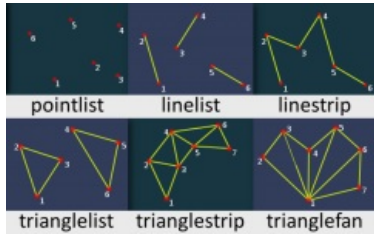
Syntax

```
bool dxDrawMaterialPrimitive ( primitiveType pType, mixed material, bool postGUI, table vertex1 [, table vertex2, ...] )
```

Required Arguments

- **pType:** Type of primitive to be drawn.
- **image:** Either a material element or a filepath of the image which is going to be drawn. (.dds images are also supported). Image files should ideally have dimensions that are a power of two, to prevent possible blurring. Use a texture created with `dxCreateTexture` to **speed up drawing**.
- **postGUI:** A bool representing whether the line should be drawn on top of or behind any ingame GUI (rendered by CEGUI).
- **vertices:** Tables representing each primitive vertex, required amount of them is determined by primitive type.

Allowed types



Available primitive types.

More info on primitives may be found on this MSDN site

- **pointlist:** Renders the vertices as a collection of isolated points.
- **linelist:** Renders the vertices as a list of isolated straight line segments.
- **linestrip:** Renders the vertices as a single polyline.
- **trianglelist:** Renders the specified vertices as a sequence of isolated triangles. Each group of three vertices defines a separate triangle.
- **trianglestrip:** Renders the vertices as a triangle strip.
- **trianglefan:** Renders the vertices as a triangle fan.

Vertices format

- **posX:** An float representing the absolute X position of the vertex, represented by pixels on the screen.
- **posY:** An float representing the absolute Y position of the vertex, represented by pixels on the screen.
- **color (optional):** An integer of the hex color, produced using `tocolor` or 0xAARRGGBB (AA = alpha, RR = red, GG = green, BB = blue). If it's not specified, white color is used.
- **u:** An float representing the relative X coordinate of the top left corner of the material which should be drawn from image
- **v:** An float representing the relative Y coordinate of the top left corner of the material which should be drawn from image

Returns

Returns a *true* if the operation was successful, *false* otherwise.