Client-side function

DxDrawImageSection

Differing from dxDrawImage, this function only draws a part of an image on the screen for a single frame. In order for the image to stay visible continuously, you need to call this function with the same parameters on each frame update (see onClientRender).

Image files 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...

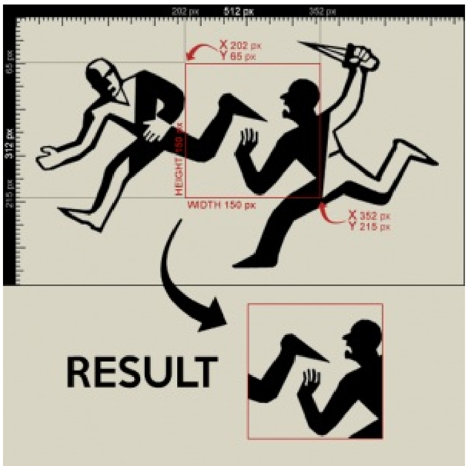
> **Tip:** Use a texture created with dxCreateTexture to **speed up drawing**.

> **Tip:** To help prevent edge artifacts when drawing textures, set **textureEdge** to **"clamp"** when calling dxCreateTexture

Syntax

```
bool dxDrawImageSection ( float posX, float posY, float width, float height,
                          float u, float v, float usize, float vsize, mixed image,
                        [ float rotation = 0, float rotationCenterOffsetX = 0, float rotationCenterOffsetY = 0,
                          int color = white, bool postGUI = false ] )
```

Required Arguments



An example of how dxDrawImageSection function works in practice.

- **posX:** the absolute X coordinate of the top left corner of the image
- **posY:** the absolute Y coordinate of the top left corner of the image
- **width:** the absolute width of the image
- **height:** the absolute height of the image
- **u:** the absolute X coordinate of the top left corner of the section which should be drawn from image
- **v:** the absolute Y coordinate of the top left corner of the section which should be drawn from image
- **usize:** the absolute width of the image section
- **vsize:** the absolute height of the image section
- **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**.

Optional Arguments

- **rotation:** the rotation, in degrees for the image.
- **rotationCenterOffsetX:** the absolute X offset from the image center for which to rotate the image from.
- **rotationCenterOffsetY:** the absolute Y offset from the image center for which to rotate the image from.
- **color:** the color of the image, a value produced by tocolor or hexadecimal number in format: 0xAARRGGBB (AA = alpha, RR = red, GG = green, BB = blue).
- **postgui :** A bool representing whether the image should be drawn on top of or behind any ingame GUI (rendered by CEGUI).

Returns

Returns *true* if successful, *false* otherwise.