

NAME

glCopyTexSubImage2D - copy a two-dimensional texture subimage

C SPECIFICATION

```
void glCopyTexSubImage2D( GLenum target,
                          GLint level,
                          GLint xoffset,
                          GLint yoffset,
                          GLint x,
                          GLint y,
                          GLsizei width,
                          GLsizei height )
```

PARAMETERS

- target* Specifies the target texture. Must be **GL_TEXTURE_2D**
- level* Specifies the level-of-detail number. Level 0 is the base image level. Level *n* is the *n*th mipmap reduction image.
- xoffset* Specifies a texel offset in the x direction within the texture array.
- yoffset* Specifies a texel offset in the y direction within the texture array.
- x, y* Specify the window coordinates of the lower left corner of the rectangular region of pixels to be copied.
- width* Specifies the width of the texture subimage.
- height* Specifies the height of the texture subimage.

DESCRIPTION

glCopyTexSubImage2D replaces a rectangular portion of a two-dimensional texture image with pixels from the current **GL_READ_BUFFER** (rather than from main memory, as is the case for **glTexSubImage2D**).

The screen-aligned pixel rectangle with lower left corner at (*x, y*) and with width *width* and height *height* replaces the portion of the texture array with x indices *xoffset* through *xoffset + width - 1*, inclusive, and y indices *yoffset* through *yoffset + height - 1*, inclusive, at the mipmap level specified by *level*.

The pixels in the rectangle are processed exactly as if **glCopyPixels** had been called, but the process stops just before final conversion. At this point, all pixel component values are clamped to the range [0, 1] and then converted to the texture's internal format for storage in the texel array.

The destination rectangle in the texture array may not include any texels outside the texture array as it was originally specified. It is not an error to specify a subtexture with zero width or height, but such a specification has no effect.

If any of the pixels within the specified rectangle of the current **GL_READ_BUFFER** are outside the read window associated with the current rendering context, then the

values obtained for those pixels are undefined.

No change is made to the *internalformat*, *width*, *height*, or *border* parameters of the specified texture array or to *texel* values outside the specified subregion.

NOTES

glCopyTexSubImage2D is available only if the GL version is 1.1 or greater.

Texturing has no effect in color index mode.

glPixelStore and **glPixelTransfer** modes affect texture images in exactly the way they affect **glDrawPixels**.

ERRORS

GL_INVALID_ENUM is generated if *target* is not **GL_TEXTURE_2D**.

GL_INVALID_OPERATION is generated if the texture array has not been defined by a previous **glTexImage2D** or **glCopyTexImage2D** operation.

GL_INVALID_VALUE is generated if *level* is less than 0.

GL_INVALID_VALUE may be generated if *level* is greater than $\log \max$, where \max is the returned value of **GL_MAX_TEXTURE_SIZE**.

GL_INVALID_VALUE is generated if $x < -b$ or if $y < -b$, where b is the border width of the texture array.

GL_INVALID_VALUE is generated if $xoffset < -b$, $(xoffset + width) > (w - b)$, $yoffset < -b$, or $(yoffset + height) > (h - b)$, where w is the **GL_TEXTURE_WIDTH**, h is the **GL_TEXTURE_HEIGHT**, and b is the **GL_TEXTURE_BORDER** of the texture image being modified. Note that w and h include twice the border width.

GL_INVALID_OPERATION is generated if **glCopyTexSubImage2D** is executed between the execution of **glBegin** and the corresponding execution of **glEnd**.

ASSOCIATED GETS

glGetTexImage
glIsEnabled with argument **GL_TEXTURE_2D**

SEE ALSO

glCopyPixels, **glCopyTexImage1D**, **glCopyTexImage2D**,
glCopyTexSubImage1D, **glPixelStore**, **glPixelTransfer**,
glTexEnv, **glTexGen**, **glTexImage1D**, **glTexImage2D**,
glTexParameter, **glTexSubImage1D**, **glTexSubImage2D**

