

NAME

glRasterPos2d, **glRasterPos2f**, **glRasterPos2i**, **glRasterPos2s**, **glRasterPos3d**, **glRasterPos3f**, **glRasterPos3i**, **glRasterPos3s**, **glRasterPos4d**, **glRasterPos4f**, **glRasterPos4i**, **glRasterPos4s**, **glRasterPos2dv**, **glRasterPos2fv**, **glRasterPos2iv**, **glRasterPos2sv**, **glRasterPos3dv**, **glRasterPos3fv**, **glRasterPos3iv**, **glRasterPos3sv**, **glRasterPos4dv**, **glRasterPos4fv**, **glRasterPos4iv**, **glRasterPos4sv** – specify the raster position for pixel operations

C SPECIFICATION

```
void glRasterPos2d( GLdouble x,
                   GLdouble y )
void glRasterPos2f( GLfloat x,
                   GLfloat y )
void glRasterPos2i( GLint x,
                   GLint y )
void glRasterPos2s( GLshort x,
                   GLshort y )
void glRasterPos3d( GLdouble x,
                   GLdouble y,
                   GLdouble z )
void glRasterPos3f( GLfloat x,
                   GLfloat y,
                   GLfloat z )
void glRasterPos3i( GLint x,
                   GLint y,
                   GLint z )
void glRasterPos3s( GLshort x,
                   GLshort y,
                   GLshort z )
void glRasterPos4d( GLdouble x,
                   GLdouble y,
                   GLdouble z,
                   GLdouble w )
void glRasterPos4f( GLfloat x,
                   GLfloat y,
                   GLfloat z,
                   GLfloat w )
void glRasterPos4i( GLint x,
                   GLint y,
                   GLint z,
                   GLint w )
void glRasterPos4s( GLshort x,
                   GLshort y,
                   GLshort z,
                   GLshort w )
```

delim \$\$

PARAMETERS

x, y, z, w

Specify the x , y , z , and w object coordinates (if present) for the raster position.

C SPECIFICATION

```
void glRasterPos2dv( const GLdouble *v )
```

```

void glRasterPos2fv( const GLfloat *v )
void glRasterPos2iv( const GLint *v )
void glRasterPos2sv( const GLshort *v )
void glRasterPos3dv( const GLdouble *v )
void glRasterPos3fv( const GLfloat *v )
void glRasterPos3iv( const GLint *v )
void glRasterPos3sv( const GLshort *v )
void glRasterPos4dv( const GLdouble *v )
void glRasterPos4fv( const GLfloat *v )
void glRasterPos4iv( const GLint *v )
void glRasterPos4sv( const GLshort *v )

```

PARAMETERS

v Specifies a pointer to an array of two, three, or four elements, specifying *\$x\$*, *\$y\$*, *\$z\$*, and *\$w\$* coordinates, respectively.

DESCRIPTION

The GL maintains a 3D position in window coordinates. This position, called the raster position, is used to position pixel and bitmap write operations. It is maintained with subpixel accuracy. See **glBitmap**, **glDrawPixels**, and **glCopyPixels**.

The current raster position consists of three window coordinates (*\$x\$*, *\$y\$*, *\$z\$*), a clip coordinate value (*\$w\$*), an eye coordinate distance, a valid bit, and associated color data and texture coordinates. The *\$w\$* coordinate is a clip coordinate, because *\$w\$* is not projected to window coordinates. **glRasterPos4** specifies object coordinates *\$x\$*, *\$y\$*, *\$z\$*, and *\$w\$* explicitly. **glRasterPos3** specifies object coordinate *\$x\$*, *\$y\$*, and *\$z\$* explicitly, while *\$w\$* is implicitly set to 1. **glRasterPos2** uses the argument values for *\$x\$* and *\$y\$* while implicitly setting *\$z\$* and *\$w\$* to 0 and 1.

The object coordinates presented by **glRasterPos** are treated just like those of a **glVertex** command: They are transformed by the current modelview and projection matrices and passed to the clipping stage. If the vertex is not culled, then it is projected and scaled to window coordinates, which become the new current raster position, and the **GL_CURRENT_RASTER_POSITION_VALID** flag is set. If the vertex is culled, then the valid bit is cleared and the current raster position and associated color and texture coordinates are undefined.

The current raster position also includes some associated color data and texture coordinates. If lighting is enabled, then **GL_CURRENT_RASTER_COLOR** (in RGBA mode) or **GL_CURRENT_RASTER_INDEX** (in color index mode) is set to the color produced by the lighting calculation (see **glLight**, **glLightModel**, and **glShadeModel**). If lighting is disabled, current color (in RGBA mode, state variable **GL_CURRENT_COLOR**) or color index (in color index mode, state variable **GL_CURRENT_INDEX**) is used to update the current raster color.

Likewise, **GL_CURRENT_RASTER_TEXTURE_COORDS** is updated as a function of **GL_CURRENT_TEXTURE_COORDS**, based on the texture matrix and the texture generation functions (see **glTexGen**). Finally, the distance from the origin of the eye coordinate system to the vertex as transformed by only the modelview matrix replaces **GL_CURRENT_RASTER_DISTANCE**.

Initially, the current raster position is (0, 0, 0, 1), the current raster distance is 0, the valid bit is set, the associated RGBA color is (1, 1, 1, 1), the associated color index is 1, and the associated texture coordinates are (0, 0, 0, 1). In RGBA mode, **GL_CURRENT_RASTER_INDEX** is always 1; in color index mode, the current raster RGBA color always maintains its initial value.

NOTES

The raster position is modified both by **glRasterPos** and by **glBitmap**.

When the raster position coordinates are invalid, drawing commands that are based on the raster position are ignored (that is, they do not result in changes to GL state).

Calling **glDrawElements** may leave the current color or index indeterminate. If **glRasterPos** is executed while the current color or index is indeterminate, the current raster color or current raster index remains indeterminate.

To set a valid raster position outside the viewport, first set a valid raster position, then call **glBitmap** with NULL as the *bitmap* parameter.

ERRORS

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

ASSOCIATED GETS

glGet with argument **GL_CURRENT_RASTER_POSITION**

glGet with argument **GL_CURRENT_RASTER_POSITION_VALID**

glGet with argument **GL_CURRENT_RASTER_DISTANCE**

glGet with argument **GL_CURRENT_RASTER_COLOR**

glGet with argument **GL_CURRENT_RASTER_INDEX**

glGet with argument **GL_CURRENT_RASTER_TEXTURE_COORDS**

SEE ALSO

glBitmap, **glCopyPixels**, **glDrawElements**, **glDrawPixels**, **glLight**, **glLightModel**, **glShadeModel**, **glTexCoord**, **glTexGen**, **glVertex**