NAME

glPushAttrib, glPopAttrib – push and pop the server attribute stack

C SPECIFICATION

void **glPushAttrib**(GLbitfield *mask*)

PARAMETERS

mask Specifies a mask that indicates which attributes to save. Values for *mask* are listed below.

C SPECIFICATION

void **glPopAttrib**(void)

DESCRIPTION

glPushAttrib takes one argument, a mask that indicates which groups of state variables to save on the attribute stack. Symbolic constants are used to set bits in the mask. *mask* is typically constructed by ORing several of these constants together. The special mask **GL_ALL_ATTRIB_BITS** can be used to save all stackable states.

The symbolic mask constants and their associated GL state are as follows (the second column lists which attributes are saved):

GL_ACCUM_BUFFER_BIT	Accumulation buffer clear value
GL_COLOR_BUFFER_BIT	**GL_ALPHA_TEST** enable bit Alpha test function and reference value **GL_BLEND** enable bit Blending source and destination functions Constant blend color Blending equation **GL_DITHER** enable bit **GL_DRAW_BUFFER** setting **GL_COLOR_LOGIC_OP** enable bit **GL_INDEX_LOGIC_OP** enable bit Logic op function Color mode and index mode clear values Color mode and index mode writemasks
GL_CURRENT_BIT	Current RGBA color Current color index Current normal vector Current texture coordinates Current raster position **GL_CURRENT_RASTER_POSITION_VALID** flag RGBA color associated with current raster position Color index associated with current raster position Texture coordinates associated with current raster position **GL_EDGE_FLAG** flag
GL_DEPTH_BUFFER_BIT	**GL_DEPTH_TEST** enable bit Depth buffer test function Depth buffer clear value

	GL_DEPTH_WRITEMASK enable bit
GL_ENABLE_BIT	**GL_ALPHA_TEST** flag
	GL_AUTO_NORMAL flag
	GL_BLEND flag
	Enable bits for the user-definable clipping planes
	GL_COLOR_MATERIAL
	GL_CULL_FACE flag
	GL_DEPTH_TEST flag
	GL_DITHER flag
	GL_FOG flag
	GL_LIGHT*i* where 0 <= *i*<**GL_MAX_LIGHTS**
	GL_LIGHTING flag
	GL_LINE_SMOOTH flag
	GL_LINE_STIPPLE flag
	GL_COLOR_LOGIC_OP flag
	GL_INDEX_LOGIC_OP flag
	GL_MAP1_*x* where *x* is a map type
	GL_MAP2_*x* where *x* is a map type
	GL_NORMALIZE flag
	GL_POINT_SMOOTH flag
	GL_POLYGON_OFFSET_LINE flag
	GL_POLYGON_OFFSET_FILL flag
	GL_POLYGON_OFFSET_POINT flag
	GL_POLYGON_SMOOTH flag
	GL_POLYGON_STIPPLE flag
	GL_SCISSOR_TEST flag
	GL_STENCIL_TEST flag
	GL_TEXTURE_1D flag
	GL_TEXTURE_2D flag
	Flags **GL_TEXTURE_GEN_***x* where *x* is S, T, R, or Q
GL_EVAL_BIT	**GL_MAP1_***x* enable bits, where *x* is a map type
	GL_MAP2_*x* enable bits, where *x* is a map type
	1D grid endpoints and divisions
	2D grid endpoints and divisions
	GL_AUTO_NORMAL enable bit
GL_FOG_BIT	**GL_FOG** enable bit
	Fog color
	Fog density
	Linear fog start
	Linear fog end
	Fog index
	GL_FOG_MODE value
GL_HINT_BIT	**GL_PERSPECTIVE_CORRECTION_HINT** setting
	GL_POINT_SMOOTH_HINT setting
	GL_LINE_SMOOTH_HINT setting
	GL_POLYGON_SMOOTH_HINT setting

	GL_FOG_HINT setting
GL_LIGHTING_BIT	**GL_COLOR_MATERIAL** enable bit **GL_COLOR_MATERIAL_FACE** value Color material parameters that are tracking the current color Ambient scene color **GL_LIGHT_MODEL_LOCAL_VIEWER** value **GL_LIGHT_MODEL_TWO_SIDE** setting **GL_LIGHTING** enable bit Enable bit for each light Ambient, diffuse, and specular intensity for each light Direction, position, exponent, and cutoff angle for each light Constant, linear, and quadratic attenuation factors for each light Ambient, diffuse, specular, and emissive color for each material Ambient, diffuse, and specular color indices for each material Specular exponent for each material **GL_SHADE_MODEL** setting
GL_LINE_BIT	**GL_LINE_SMOOTH** flag **GL_LINE_STIPPLE** enable bit Line stipple pattern and repeat counter Line width
GL_LIST_BIT	**GL_LIST_BASE** setting
GL_PIXEL_MODE_BIT	**GL_RED_BIAS** and **GL_RED_SCALE** settings **GL_GREEN_BIAS** and **GL_GREEN_SCALE** values **GL_BLUE_BIAS** and **GL_BLUE_SCALE** **GL_ALPHA_BIAS** and **GL_ALPHA_SCALE** **GL_DEPTH_BIAS** and **GL_DEPTH_SCALE** **GL_INDEX_OFFSET** and **GL_INDEX_SHIFT** values **GL_MAP_COLOR** and **GL_MAP_STENCIL** flags **GL_ZOOM_X** and **GL_ZOOM_Y** factors **GL_READ_BUFFER** setting
GL_POINT_BIT	**GL_POINT_SMOOTH** flag Point size
GL_POLYGON_BIT	**GL_CULL_FACE** enable bit **GL_CULL_FACE_MODE** value **GL_FRONT_FACE** indicator **GL_POLYGON_MODE** setting **GL_POLYGON_SMOOTH** flag **GL_POLYGON_STIPPLE** enable bit **GL_POLYGON_OFFSET_FILL** flag **GL_POLYGON_OFFSET_LINE** flag **GL_POLYGON_OFFSET_POINT** flag **GL_POLYGON_OFFSET_FACTOR** **GL_POLYGON_OFFSET_UNITS**

GL_POLYGON_STIPPLE_BIT	Polygon stipple image
GL_SCISSOR_BIT	**GL_SCISSOR_TEST** flag Scissor box
GL_STENCIL_BUFFER_BIT	**GL_STENCIL_TEST** enable bit Stencil function and reference value Stencil value mask Stencil fail, pass, and depth buffer pass actions Stencil buffer clear value Stencil buffer writemask
GL_TEXTURE_BIT	Enable bits for the four texture coordinates Border color for each texture image Minification function for each texture image Magnification function for each texture image Texture coordinates and wrap mode for each texture image Color and mode for each texture environment Enable bits **GL_TEXTURE_GEN_***x*, *x* is S, T, R, and Q **GL_TEXTURE_GEN_MODE** setting for S, T, R, and Q **glTexGen** plane equations for S, T, R, and Q Current texture bindings (for example, **GL_TEXTURE_2D_BINDING**)
GL_TRANSFORM_BIT	Coefficients of the six clipping planes Enable bits for the user-definable clipping planes **GL_MATRIX_MODE** value **GL_NORMALIZE** flag
GL_VIEWPORT_BIT	Depth range (near and far) Viewport origin and extent

glPopAttrib restores the values of the state variables saved with the last **glPushAttrib** command. Those not saved are left unchanged.

It is an error to push attributes onto a full stack, or to pop attributes off an empty stack. In either case, the error flag is set and no other change is made to GL state.

Initially, the attribute stack is empty.

NOTES

Not all values for GL state can be saved on the attribute stack. For example, render mode state, and select and feedback state cannot be saved. Client state must be saved with **glPushClientAttrib**.

The depth of the attribute stack depends on the implementation, but it must be at least 16.

ERRORS

GL_STACK_OVERFLOW is generated if **glPushAttrib** is called while the attribute stack is full.

GL_STACK_UNDERFLOW is generated if **glPopAttrib** is called while the attribute stack is empty.

GL_INVALID_OPERATION is generated if **glPushAttrib** or **glPopAttrib** is executed between the execution of **glBegin** and the corresponding execution of **glEnd**.

ASSOCIATED GETS

glGet with argument **GL_ATTRIB_STACK_DEPTH**
glGet with argument **GL_MAX_ATTRIB_STACK_DEPTH**

SEE ALSO

glGet, **glGetClipPlane**, **glGetError**, **glGetLight**, **glGetMap**, **glGetMaterial**, **glGetPixelMap**, **glGetPolygonStipple**, **glGetString**, **glGetTexEnv**, **glGetTexGen**, **glGetTexImage**, **glGetTexLevelParameter**, **glGetTexParameter**, **glIsEnabled**, **glPushClientAttrib**