

GL: 3-D Graphics

Version 8.17.0.3

Scott Owens

May 18, 2025

The `sgl` libraries provide access to the rendering functions of OpenGL 1.5 and GLU 1.3 libraries. The `sgl` libraries do not address system-level concerns, such as the attachment of GL rendering contexts to displays. Instead, the libraries should work with any Racket extension that provides GL with access to the system (such as a binding for `glx`). Notably, the `racket/gui/base` library provides support for rendering contexts via the `canvas%` class and its `with-gl-context` method.

Contents

1	Using OpenGL	3
2	C-Style OpenGL	4
3	Racket-Style OpenGL	40
4	OpenGL Vectors	45
5	Bitmaps	51
6	Initialization	52
	Index	53
	Index	53

1 Using OpenGL

The `sgl/gl` library provides direct access to the C-style OpenGL API, whereas the `sgl` library provides a more Racket-like interface. The `sgl/gl` library provides a binding for each `#defined` constant and for most functions in OpenGL 1.5 and GLU 1.3. The functions perform comparable checking to their C-language counterparts; they check the types of their arguments, but do not check the length of array arguments. The `sgl` library provides wrappers around many of the functions in the `sgl/gl` library to present a more Racket-friendly interface, including function names that follow Racket conventions, and checked, symbolic enumeration arguments, and array-length checks.

Warning on Safety: OpenGL programming is inherently unsafe, even when using only the `sgl` library. Although `sgl` checks the arguments to each function call, violation of higher-level assumptions of the system's OpenGL library can cause it to crash, bringing the entire Racket system down. For example, sending a large number of vertices in a single `glBegin` causes at least some GL implementations to crash.

Some examples are available in the "examples" directory of the "sgl" collection in the Racket installation. For "alpha.rkt", try choosing the "sk.jpg" image distributed with Racket in the "icons" collection; you may have to press the `t` key a few times if the spinning cube is blank.

2 C-Style OpenGL

```
(require sgl/gl)      package: sgl
```

The `sgl/gl` module provides a direct interface to the system's GL library closely following the conventions of the C-language OpenGL API. It provides a binding for each `#defined` constant (these start with `GL_`) and for the functions in the GL 1.5 and GLU 1.3 specifications, except for the following:

- Vertex arrays (GL 1.5, Section 2.8)
- Buffer objects (GL 1.5, Section 2.9)
- `glGetPointerv` (GL 1.5, Section 6.1.11)
- Buffer object queries (GL 1.5, Section 6.1.13)
- Polygon tessellation (GLU 1.3, Section 5)
- `gluQuadricCallback` (GLU 1.3, Section 6.2)
- NURBS callbacks (GLU 1.3, Section 7.2)

If one of the provided functions is not present on your system (e.g. if your system supports only GL 1.3), then the corresponding `sgl/gl` function raises a run-time exception when invoked.

The functions provided by `sgl/gl` perform comparable checking to their C-language counterparts; they check the types of their arguments, but do not check the length of array arguments. The following details the kinds of Racket values that can be provided for each primitive OpenGL type:

- `GLbyte`, `GLshort`, `GLint`: exact integer in the proper range
- `GLubyte`, `GLushort`, `GLuint`: exact non-negative integer in the proper range
- `GLsizei`, `GLenum`, `GLbitfield`: exact non-negative integer in the proper range
- `GLfloat`, `GLdouble`: real number
- `GLclampf`, `GLclampd`: real number
- `GLboolean`: any value, where `#f` means `GL_FALSE` and all other values mean `GL_TRUE`; do not use `GL_FALSE` or `GL_TRUE`, since they are bound to integers, both will end up being converted to `GL_TRUE`.

OpenGL functions that take vector arguments accept `cvector` values. The type of the `cvector` is checked; for example, `glVertex3fv` expects a vector of `GLfloat`s, so `glVertex3fv` accepts only a `cvector` containing reals. See also `sgl/gl-vectors`. Functions that accept arrays of type `void*` accept any `cvector`; you must ensure that you supply the proper kind of vector, as in the C-language OpenGL API.

Examples:

```
(require sgl/gl
         sgl/gl-vectors)
(glBegin GL_TRIANGLES)
(glVertex3i 1 2 3)
(glVertex4fv (gl-float-vector 1 2 3 4))
(glEnd)
```

```
glPixelMapfv : procedure?
glPixelMapuiv : procedure?
glPixelMapusv : procedure?
glDeleteTextures : procedure?
glDeleteQueries : procedure?
```

These functions do not take a size argument, because it is derived from the length of the argument vector.

```
glGenTextures : procedure?
glGenQueries : procedure?
```

These functions do not take vector arguments. Instead, they allocate a vector of the requested size and return it.

```
glAreTexturesResident : procedure?
```

This function takes in a `GLuint` vector and textures, and it returns 2 values: the specified boolean and a boolean vector of residences.

```
glGetBooleanv : procedure?
glGetIntegerv : procedure?
glGetFloatv : procedure?
glGetDoublev : procedure?
glGetLightfv : procedure?
glGetLightiv : procedure?
glGetMaterialfv : procedure?
glGetMaterialiv : procedure?
glGetTexEnvfv : procedure?
glGetTexEnviv : procedure?
```

```

glGetTexGendv : procedure?
glGetTexGenfv : procedure?
glGetTexGeniv : procedure?
glGetTexParameterfv : procedure?
glGetTexParameteriv : procedure?
glGetTexLevelParameterfv : procedure?
glGetTexLevelParameteriv : procedure?
glGetPixelMapfv : procedure?
glGetPixelMapuiv : procedure?
glGetPixelMapusv : procedure?
glGetMapdv : procedure?
glGetMapfv : procedure?
glGetMapiv : procedure?
glGetBufferParameteriv : procedure?
glGetConvolutionParameterfv : procedure?
glGetConvolutionParameteriv : procedure?
glGetHistogramParameterfv : procedure?
glGetHistogramParameteriv : procedure?
glGetMinmaxParameterfv : procedure?
glGetMinmaxParameteriv : procedure?
glGetQueryiv : procedure?
glGetQueryObjectiv : procedure?
glGetQueryObjectuiv : procedure?

```

Instead of taking a vector argument, these function take an integer argument that specifies the size of the vector that is returned.

```

glGetClipPlane : procedure?

```

This function does not take a vector argument and returns a GLdouble vector of length 4.

```

glGetString : procedure?
gluCheckExtension : procedure?
gluErrorString : procedure?
gluGetString : procedure?

```

These functions deal with strings instead of GLubyte vectors.

```

gluProject : procedure?
gluUnProject : procedure?
gluUnProject4 : procedure?

```

Instead of taking pointers to GLdoubles for return values, these function directly return GLdouble vectors.

```
glSelectBuffer : procedure?  
glFeedbackBuffer : procedure?
```

These functions do not take vectors, instead they return a `selection-buffer-object` or `feedback-buffer-object`. The `select-buffer->gl-uint-vector` and `feedback-buffer->gl-float-vector` functions copy the contents of the buffer into a vector. Because the OpenGL library writes to the buffer-object on OpenGL function calls after `glSelectBuffer` or `glFeedbackBuffer` has returned, if the buffer is garbage collected before OpenGL is finished writing to it, the entire Racket system can crash. The `gl-process-selection` function in `sgl` helps interpret the results of `glSelectBuffer` in a Racket-friendly format.

```
glAccum : procedure?  
glActiveTexture : procedure?  
glAlphaFunc : procedure?  
glBegin : procedure?  
glBeginQuery : procedure?  
glBindTexture : procedure?  
glBitmap : procedure?  
glBlendColor : procedure?  
glBlendEquation : procedure?  
glBlendFunc : procedure?  
glBlendFuncSeparate : procedure?  
glCallList : procedure?  
glCallLists : procedure?  
glClear : procedure?  
glClearAccum : procedure?  
glClearColor : procedure?  
glClearDepth : procedure?  
glClearIndex : procedure?  
glClearStencil : procedure?  
glClipPlane : procedure?  
glColor3b : procedure?  
glColor3bv : procedure?  
glColor3d : procedure?  
glColor3dv : procedure?  
glColor3f : procedure?  
glColor3fv : procedure?  
glColor3i : procedure?  
glColor3iv : procedure?  
glColor3s : procedure?  
glColor3sv : procedure?  
glColor3ub : procedure?  
glColor3ubv : procedure?
```

```
glColor3ui : procedure?
glColor3uiv : procedure?
glColor3us : procedure?
glColor3usv : procedure?
glColor4b : procedure?
glColor4bv : procedure?
glColor4d : procedure?
glColor4dv : procedure?
glColor4f : procedure?
glColor4fv : procedure?
glColor4i : procedure?
glColor4iv : procedure?
glColor4s : procedure?
glColor4sv : procedure?
glColor4ub : procedure?
glColor4ubv : procedure?
glColor4ui : procedure?
glColor4uiv : procedure?
glColor4us : procedure?
glColor4usv : procedure?
glColorMask : procedure?
glColorMaterial : procedure?
glColorSubTable : procedure?
glColorTable : procedure?
glColorTableParameterfv : procedure?
glColorTableParameteriv : procedure?
glCompressedTexImage1D : procedure?
glCompressedTexImage2D : procedure?
glCompressedTexImage3D : procedure?
glCompressedTexSubImage1D : procedure?
glCompressedTexSubImage2D : procedure?
glCompressedTexSubImage3D : procedure?
glConvolutionFilter1D : procedure?
glConvolutionFilter2D : procedure?
glConvolutionParameterf : procedure?
glConvolutionParameterfv : procedure?
glConvolutionParameteri : procedure?
glConvolutionParameteriv : procedure?
glCopyColorSubTable : procedure?
glCopyColorTable : procedure?
glCopyConvolutionFilter1D : procedure?
glCopyConvolutionFilter2D : procedure?
glCopyPixels : procedure?
glCopyTexImage1D : procedure?
```

```
glCopyTexImage2D : procedure?
glCopyTexSubImage1D : procedure?
glCopyTexSubImage2D : procedure?
glCopyTexSubImage3D : procedure?
glCullFace : procedure?
glDeleteLists : procedure?
glDepthFunc : procedure?
glDepthMask : procedure?
glDepthRange : procedure?
glDisable : procedure?
glDrawBuffer : procedure?
glDrawPixels : procedure?
glEdgeFlag : procedure?
glEdgeFlagv : procedure?
glEnable : procedure?
glEnd : procedure?
glEndList : procedure?
glEndQuery : procedure?
glEvalCoord1d : procedure?
glEvalCoord1dv : procedure?
glEvalCoord1f : procedure?
glEvalCoord1fv : procedure?
glEvalCoord2d : procedure?
glEvalCoord2dv : procedure?
glEvalCoord2f : procedure?
glEvalCoord2fv : procedure?
glEvalMesh1 : procedure?
glEvalMesh2 : procedure?
glEvalPoint1 : procedure?
glEvalPoint2 : procedure?
glFinish : procedure?
glFlush : procedure?
glFogCoordd : procedure?
glFogCoorddv : procedure?
glFogCoordf : procedure?
glFogCoordfv : procedure?
glFogf : procedure?
glFogfv : procedure?
glFogi : procedure?
glFogiv : procedure?
glFrontFace : procedure?
glFrustum : procedure?
glGenLists : procedure?
glGetColorTable : procedure?
```

```
glGetCompressedTexImage : procedure?  
glGetConvolutionFilter : procedure?  
glGetError : procedure?  
glGetHistogram : procedure?  
glGetMinmax : procedure?  
glGetPolygonStipple : procedure?  
glGetSeparableFilter : procedure?  
glGetTexImage : procedure?  
glHint : procedure?  
glHistogram : procedure?  
glIndexMask : procedure?  
glIndexd : procedure?  
glIndexdv : procedure?  
glIndexf : procedure?  
glIndexfv : procedure?  
glIndexi : procedure?  
glIndexiv : procedure?  
glIndexs : procedure?  
glIndexsv : procedure?  
glIndexub : procedure?  
glIndexubv : procedure?  
glInitNames : procedure?  
glIsBuffer : procedure?  
glIsEnabled : procedure?  
glIsList : procedure?  
glIsQuery : procedure?  
glIsTexture : procedure?  
glLightModelf : procedure?  
glLightModelfv : procedure?  
glLightModeli : procedure?  
glLightModeliv : procedure?  
glLightf : procedure?  
glLightfv : procedure?  
glLighti : procedure?  
glLightiv : procedure?  
glLineStipple : procedure?  
glLineWidth : procedure?  
glListBase : procedure?  
glLoadIdentity : procedure?  
glLoadMatrixd : procedure?  
glLoadMatrixf : procedure?  
glLoadName : procedure?  
glLoadTransposeMatrixd : procedure?  
glLoadTransposeMatrixf : procedure?
```

```
glLogicOp : procedure?
glMap1d : procedure?
glMap1f : procedure?
glMap2d : procedure?
glMap2f : procedure?
glMapGrid1d : procedure?
glMapGrid1f : procedure?
glMapGrid2d : procedure?
glMapGrid2f : procedure?
glMaterialf : procedure?
glMaterialfv : procedure?
glMateriali : procedure?
glMaterialiv : procedure?
glMatrixMode : procedure?
glMinmax : procedure?
glMultMatrixd : procedure?
glMultMatrixf : procedure?
glMultTransposeMatrixd : procedure?
glMultTransposeMatrixf : procedure?
glMultiTexCoord1d : procedure?
glMultiTexCoord1dv : procedure?
glMultiTexCoord1f : procedure?
glMultiTexCoord1fv : procedure?
glMultiTexCoord1i : procedure?
glMultiTexCoord1iv : procedure?
glMultiTexCoord1s : procedure?
glMultiTexCoord1sv : procedure?
glMultiTexCoord2d : procedure?
glMultiTexCoord2dv : procedure?
glMultiTexCoord2f : procedure?
glMultiTexCoord2fv : procedure?
glMultiTexCoord2i : procedure?
glMultiTexCoord2iv : procedure?
glMultiTexCoord2s : procedure?
glMultiTexCoord2sv : procedure?
glMultiTexCoord3d : procedure?
glMultiTexCoord3dv : procedure?
glMultiTexCoord3f : procedure?
glMultiTexCoord3fv : procedure?
glMultiTexCoord3i : procedure?
glMultiTexCoord3iv : procedure?
glMultiTexCoord3s : procedure?
glMultiTexCoord3sv : procedure?
glMultiTexCoord4d : procedure?
```

```
glMultiTexCoord4dv : procedure?
glMultiTexCoord4f : procedure?
glMultiTexCoord4fv : procedure?
glMultiTexCoord4i : procedure?
glMultiTexCoord4iv : procedure?
glMultiTexCoord4s : procedure?
glMultiTexCoord4sv : procedure?
glNewList : procedure?
glNormal3b : procedure?
glNormal3bv : procedure?
glNormal3d : procedure?
glNormal3dv : procedure?
glNormal3f : procedure?
glNormal3fv : procedure?
glNormal3i : procedure?
glNormal3iv : procedure?
glNormal3s : procedure?
glNormal3sv : procedure?
glOrtho : procedure?
glPassThrough : procedure?
glPixelStoref : procedure?
glPixelStorei : procedure?
glPixelTransferf : procedure?
glPixelTransferi : procedure?
glPixelZoom : procedure?
glPointParameterf : procedure?
glPointParameterfv : procedure?
glPointParameteri : procedure?
glPointParameteriv : procedure?
glPointSize : procedure?
glPolygonMode : procedure?
glPolygonOffset : procedure?
glPolygonStipple : procedure?
glPopAttrib : procedure?
glPopClientAttrib : procedure?
glPopMatrix : procedure?
glPopName : procedure?
glPushAttrib : procedure?
glPushClientAttrib : procedure?
glPushMatrix : procedure?
glPushName : procedure?
glRasterPos2d : procedure?
glRasterPos2dv : procedure?
glRasterPos2f : procedure?
```

```
glRasterPos2fv : procedure?
glRasterPos2i : procedure?
glRasterPos2iv : procedure?
glRasterPos2s : procedure?
glRasterPos2sv : procedure?
glRasterPos3d : procedure?
glRasterPos3dv : procedure?
glRasterPos3f : procedure?
glRasterPos3fv : procedure?
glRasterPos3i : procedure?
glRasterPos3iv : procedure?
glRasterPos3s : procedure?
glRasterPos3sv : procedure?
glRasterPos4d : procedure?
glRasterPos4dv : procedure?
glRasterPos4f : procedure?
glRasterPos4fv : procedure?
glRasterPos4i : procedure?
glRasterPos4iv : procedure?
glRasterPos4s : procedure?
glRasterPos4sv : procedure?
glReadBuffer : procedure?
glReadPixels : procedure?
glRectd : procedure?
glRectdv : procedure?
glRectf : procedure?
glRectfv : procedure?
glRecti : procedure?
glRectiv : procedure?
glRects : procedure?
glRectsv : procedure?
glRenderMode : procedure?
glResetHistogram : procedure?
glResetMinmax : procedure?
glRotated : procedure?
glRotatef : procedure?
glSampleCoverage : procedure?
glScaled : procedure?
glScalef : procedure?
glScissor : procedure?
glSecondaryColor3b : procedure?
glSecondaryColor3bv : procedure?
glSecondaryColor3d : procedure?
glSecondaryColor3dv : procedure?
```

```
glSecondaryColor3f : procedure?
glSecondaryColor3fv : procedure?
glSecondaryColor3i : procedure?
glSecondaryColor3iv : procedure?
glSecondaryColor3s : procedure?
glSecondaryColor3sv : procedure?
glSecondaryColor3ub : procedure?
glSecondaryColor3ubv : procedure?
glSecondaryColor3ui : procedure?
glSecondaryColor3uiv : procedure?
glSecondaryColor3us : procedure?
glSecondaryColor3usv : procedure?
glSeparableFilter2D : procedure?
glShadeModel : procedure?
glStencilFunc : procedure?
glStencilMask : procedure?
glStencilOp : procedure?
glTexCoord1d : procedure?
glTexCoord1dv : procedure?
glTexCoord1f : procedure?
glTexCoord1fv : procedure?
glTexCoord1i : procedure?
glTexCoord1iv : procedure?
glTexCoord1s : procedure?
glTexCoord1sv : procedure?
glTexCoord2d : procedure?
glTexCoord2dv : procedure?
glTexCoord2f : procedure?
glTexCoord2fv : procedure?
glTexCoord2i : procedure?
glTexCoord2iv : procedure?
glTexCoord2s : procedure?
glTexCoord2sv : procedure?
glTexCoord3d : procedure?
glTexCoord3dv : procedure?
glTexCoord3f : procedure?
glTexCoord3fv : procedure?
glTexCoord3i : procedure?
glTexCoord3iv : procedure?
glTexCoord3s : procedure?
glTexCoord3sv : procedure?
glTexCoord4d : procedure?
glTexCoord4dv : procedure?
glTexCoord4f : procedure?
```

```
glTexCoord4fv : procedure?
glTexCoord4i : procedure?
glTexCoord4iv : procedure?
glTexCoord4s : procedure?
glTexCoord4sv : procedure?
glTexEnvf : procedure?
glTexEnvfv : procedure?
glTexEnvi : procedure?
glTexEnviv : procedure?
glTexGend : procedure?
glTexGendv : procedure?
glTexGenf : procedure?
glTexGenfv : procedure?
glTexGeni : procedure?
glTexGeniv : procedure?
glTexImage1D : procedure?
glTexImage2D : procedure?
glTexImage3D : procedure?
glTexParameterf : procedure?
glTexParameterfv : procedure?
glTexParameteri : procedure?
glTexParameteriv : procedure?
glTexSubImage1D : procedure?
glTexSubImage2D : procedure?
glTexSubImage3D : procedure?
glTranslated : procedure?
glTranslatef : procedure?
glVertex2d : procedure?
glVertex2dv : procedure?
glVertex2f : procedure?
glVertex2fv : procedure?
glVertex2i : procedure?
glVertex2iv : procedure?
glVertex2s : procedure?
glVertex2sv : procedure?
glVertex3d : procedure?
glVertex3dv : procedure?
glVertex3f : procedure?
glVertex3fv : procedure?
glVertex3i : procedure?
glVertex3iv : procedure?
glVertex3s : procedure?
glVertex3sv : procedure?
glVertex4d : procedure?
```

```
glVertex4dv : procedure?
glVertex4f : procedure?
glVertex4fv : procedure?
glVertex4i : procedure?
glVertex4iv : procedure?
glVertex4s : procedure?
glVertex4sv : procedure?
glViewport : procedure?
glWindowPos2d : procedure?
glWindowPos2dv : procedure?
glWindowPos2f : procedure?
glWindowPos2fv : procedure?
glWindowPos2i : procedure?
glWindowPos2iv : procedure?
glWindowPos2s : procedure?
glWindowPos2sv : procedure?
glWindowPos3d : procedure?
glWindowPos3dv : procedure?
glWindowPos3f : procedure?
glWindowPos3fv : procedure?
glWindowPos3i : procedure?
glWindowPos3iv : procedure?
glWindowPos3s : procedure?
glWindowPos3sv : procedure?
gluBuild1DMipmapLevels : procedure?
gluBuild1DMipmaps : procedure?
gluBuild2DMipmapLevels : procedure?
gluBuild2DMipmaps : procedure?
gluBuild3DMipmapLevels : procedure?
gluBuild3DMipmaps : procedure?
gluCylinder : procedure?
gluDisk : procedure?
gluLookAt : procedure?
gluNewQuadric : procedure?
gluOrtho2D : procedure?
gluPartialDisk : procedure?
gluPerspective : procedure?
gluPickMatrix : procedure?
gluQuadricDrawStyle : procedure?
gluQuadricNormals : procedure?
gluQuadricOrientation : procedure?
gluQuadricTexture : procedure?
gluScaleImage : procedure?
gluSphere : procedure?
```

These functions are all direct translations of the C OpenGL API.

```
GL_FALSE : exact-integer?
GL_TRUE  : exact-integer?
GL_BYTE  : exact-integer?
GL_UNSIGNED_BYTE : exact-integer?
GL_SHORT : exact-integer?
GL_UNSIGNED_SHORT : exact-integer?
GL_INT    : exact-integer?
GL_UNSIGNED_INT : exact-integer?
GL_FLOAT  : exact-integer?
GL_DOUBLE : exact-integer?
GL_2_BYTES : exact-integer?
GL_3_BYTES : exact-integer?
GL_4_BYTES : exact-integer?
GL_POINTS  : exact-integer?
GL_LINES   : exact-integer?
GL_LINE_LOOP : exact-integer?
GL_LINE_STRIP : exact-integer?
GL_TRIANGLES : exact-integer?
GL_TRIANGLE_STRIP : exact-integer?
GL_TRIANGLE_FAN : exact-integer?
GL_QUADS    : exact-integer?
GL_QUAD_STRIP : exact-integer?
GL_POLYGON  : exact-integer?
GL_VERTEX_ARRAY : exact-integer?
GL_NORMAL_ARRAY : exact-integer?
GL_COLOR_ARRAY : exact-integer?
GL_INDEX_ARRAY : exact-integer?
GL_TEXTURE_COORD_ARRAY : exact-integer?
GL_EDGE_FLAG_ARRAY : exact-integer?
GL_VERTEX_ARRAY_SIZE : exact-integer?
GL_VERTEX_ARRAY_TYPE : exact-integer?
GL_VERTEX_ARRAY_STRIDE : exact-integer?
GL_NORMAL_ARRAY_TYPE : exact-integer?
GL_NORMAL_ARRAY_STRIDE : exact-integer?
GL_COLOR_ARRAY_SIZE : exact-integer?
GL_COLOR_ARRAY_TYPE : exact-integer?
GL_COLOR_ARRAY_STRIDE : exact-integer?
GL_INDEX_ARRAY_TYPE : exact-integer?
GL_INDEX_ARRAY_STRIDE : exact-integer?
GL_TEXTURE_COORD_ARRAY_SIZE : exact-integer?
GL_TEXTURE_COORD_ARRAY_TYPE : exact-integer?
GL_TEXTURE_COORD_ARRAY_STRIDE : exact-integer?
GL_EDGE_FLAG_ARRAY_STRIDE : exact-integer?
```

GL_VERTEX_ARRAY_POINTER : exact-integer?
GL_NORMAL_ARRAY_POINTER : exact-integer?
GL_COLOR_ARRAY_POINTER : exact-integer?
GL_INDEX_ARRAY_POINTER : exact-integer?
GL_TEXTURE_COORD_ARRAY_POINTER : exact-integer?
GL_EDGE_FLAG_ARRAY_POINTER : exact-integer?
GL_V2F : exact-integer?
GL_V3F : exact-integer?
GL_C4UB_V2F : exact-integer?
GL_C4UB_V3F : exact-integer?
GL_C3F_V3F : exact-integer?
GL_N3F_V3F : exact-integer?
GL_C4F_N3F_V3F : exact-integer?
GL_T2F_V3F : exact-integer?
GL_T4F_V4F : exact-integer?
GL_T2F_C4UB_V3F : exact-integer?
GL_T2F_C3F_V3F : exact-integer?
GL_T2F_N3F_V3F : exact-integer?
GL_T2F_C4F_N3F_V3F : exact-integer?
GL_T4F_C4F_N3F_V4F : exact-integer?
GL_MATRIX_MODE : exact-integer?
GL_MODELVIEW : exact-integer?
GL_PROJECTION : exact-integer?
GL_TEXTURE : exact-integer?
GL_POINT_SMOOTH : exact-integer?
GL_POINT_SIZE : exact-integer?
GL_POINT_SIZE_GRANULARITY : exact-integer?
GL_POINT_SIZE_RANGE : exact-integer?
GL_LINE_SMOOTH : exact-integer?
GL_LINE_STIPPLE : exact-integer?
GL_LINE_STIPPLE_PATTERN : exact-integer?
GL_LINE_STIPPLE_REPEAT : exact-integer?
GL_LINE_WIDTH : exact-integer?
GL_LINE_WIDTH_GRANULARITY : exact-integer?
GL_LINE_WIDTH_RANGE : exact-integer?
GL_POINT : exact-integer?
GL_LINE : exact-integer?
GL_FILL : exact-integer?
GL_CW : exact-integer?
GL_CCW : exact-integer?
GL_FRONT : exact-integer?
GL_BACK : exact-integer?
GL_POLYGON_MODE : exact-integer?
GL_POLYGON_SMOOTH : exact-integer?

GL_POLYGON_STIPPLE : exact-integer?
GL_EDGE_FLAG : exact-integer?
GL_CULL_FACE : exact-integer?
GL_CULL_FACE_MODE : exact-integer?
GL_FRONT_FACE : exact-integer?
GL_POLYGON_OFFSET_FACTOR : exact-integer?
GL_POLYGON_OFFSET_UNITS : exact-integer?
GL_POLYGON_OFFSET_POINT : exact-integer?
GL_POLYGON_OFFSET_LINE : exact-integer?
GL_POLYGON_OFFSET_FILL : exact-integer?
GL_COMPILE : exact-integer?
GL_COMPILE_AND_EXECUTE : exact-integer?
GL_LIST_BASE : exact-integer?
GL_LIST_INDEX : exact-integer?
GL_LIST_MODE : exact-integer?
GL_NEVER : exact-integer?
GL_LESS : exact-integer?
GL_EQUAL : exact-integer?
GL_LEQUAL : exact-integer?
GL_GREATER : exact-integer?
GL_NOTEQUAL : exact-integer?
GL_GEQUAL : exact-integer?
GL_ALWAYS : exact-integer?
GL_DEPTH_TEST : exact-integer?
GL_DEPTH_BITS : exact-integer?
GL_DEPTH_CLEAR_VALUE : exact-integer?
GL_DEPTH_FUNC : exact-integer?
GL_DEPTH_RANGE : exact-integer?
GL_DEPTH_WRITEMASK : exact-integer?
GL_DEPTH_COMPONENT : exact-integer?
GL_LIGHTING : exact-integer?
GL_LIGHT0 : exact-integer?
GL_LIGHT1 : exact-integer?
GL_LIGHT2 : exact-integer?
GL_LIGHT3 : exact-integer?
GL_LIGHT4 : exact-integer?
GL_LIGHT5 : exact-integer?
GL_LIGHT6 : exact-integer?
GL_LIGHT7 : exact-integer?
GL_SPOT_EXPONENT : exact-integer?
GL_SPOT_CUTOFF : exact-integer?
GL_CONSTANT_ATTENUATION : exact-integer?
GL_LINEAR_ATTENUATION : exact-integer?
GL_QUADRATIC_ATTENUATION : exact-integer?

GL_AMBIENT : exact-integer?
GL_DIFFUSE : exact-integer?
GL_SPECULAR : exact-integer?
GL_SHININESS : exact-integer?
GL_EMISSION : exact-integer?
GL_POSITION : exact-integer?
GL_SPOT_DIRECTION : exact-integer?
GL_AMBIENT_AND_DIFFUSE : exact-integer?
GL_COLOR_INDEXES : exact-integer?
GL_LIGHT_MODEL_TWO_SIDE : exact-integer?
GL_LIGHT_MODEL_LOCAL_VIEWER : exact-integer?
GL_LIGHT_MODEL_AMBIENT : exact-integer?
GL_FRONT_AND_BACK : exact-integer?
GL_SHADE_MODEL : exact-integer?
GL_FLAT : exact-integer?
GL_SMOOTH : exact-integer?
GL_COLOR_MATERIAL : exact-integer?
GL_COLOR_MATERIAL_FACE : exact-integer?
GL_COLOR_MATERIAL_PARAMETER : exact-integer?
GL_NORMALIZE : exact-integer?
GL_CLIP_PLANE0 : exact-integer?
GL_CLIP_PLANE1 : exact-integer?
GL_CLIP_PLANE2 : exact-integer?
GL_CLIP_PLANE3 : exact-integer?
GL_CLIP_PLANE4 : exact-integer?
GL_CLIP_PLANE5 : exact-integer?
GL_ACCUM_RED_BITS : exact-integer?
GL_ACCUM_GREEN_BITS : exact-integer?
GL_ACCUM_BLUE_BITS : exact-integer?
GL_ACCUM_ALPHA_BITS : exact-integer?
GL_ACCUM_CLEAR_VALUE : exact-integer?
GL_ACCUM : exact-integer?
GL_ADD : exact-integer?
GL_LOAD : exact-integer?
GL_MULT : exact-integer?
GL_RETURN : exact-integer?
GL_ALPHA_TEST : exact-integer?
GL_ALPHA_TEST_REF : exact-integer?
GL_ALPHA_TEST_FUNC : exact-integer?
GL_BLEND : exact-integer?
GL_BLEND_SRC : exact-integer?
GL_BLEND_DST : exact-integer?
GL_ZERO : exact-integer?
GL_ONE : exact-integer?

```
GL_SRC_COLOR : exact-integer?
GL_ONE_MINUS_SRC_COLOR : exact-integer?
GL_SRC_ALPHA : exact-integer?
GL_ONE_MINUS_SRC_ALPHA : exact-integer?
GL_DST_ALPHA : exact-integer?
GL_ONE_MINUS_DST_ALPHA : exact-integer?
GL_DST_COLOR : exact-integer?
GL_ONE_MINUS_DST_COLOR : exact-integer?
GL_SRC_ALPHA_SATURATE : exact-integer?
GL_FEEDBACK : exact-integer?
GL_RENDER : exact-integer?
GL_SELECT : exact-integer?
GL_2D : exact-integer?
GL_3D : exact-integer?
GL_3D_COLOR : exact-integer?
GL_3D_COLOR_TEXTURE : exact-integer?
GL_4D_COLOR_TEXTURE : exact-integer?
GL_POINT_TOKEN : exact-integer?
GL_LINE_TOKEN : exact-integer?
GL_LINE_RESET_TOKEN : exact-integer?
GL_POLYGON_TOKEN : exact-integer?
GL_BITMAP_TOKEN : exact-integer?
GL_DRAW_PIXEL_TOKEN : exact-integer?
GL_COPY_PIXEL_TOKEN : exact-integer?
GL_PASS_THROUGH_TOKEN : exact-integer?
GL_FEEDBACK_BUFFER_POINTER : exact-integer?
GL_FEEDBACK_BUFFER_SIZE : exact-integer?
GL_FEEDBACK_BUFFER_TYPE : exact-integer?
GL_SELECTION_BUFFER_POINTER : exact-integer?
GL_SELECTION_BUFFER_SIZE : exact-integer?
GL_FOG : exact-integer?
GL_FOG_MODE : exact-integer?
GL_FOG_DENSITY : exact-integer?
GL_FOG_COLOR : exact-integer?
GL_FOG_INDEX : exact-integer?
GL_FOG_START : exact-integer?
GL_FOG_END : exact-integer?
GL_LINEAR : exact-integer?
GL_EXP : exact-integer?
GL_EXP2 : exact-integer?
GL_LOGIC_OP : exact-integer?
GL_INDEX_LOGIC_OP : exact-integer?
GL_COLOR_LOGIC_OP : exact-integer?
GL_LOGIC_OP_MODE : exact-integer?
```

```
GL_CLEAR : exact-integer?
GL_SET : exact-integer?
GL_COPY : exact-integer?
GL_COPY_INVERTED : exact-integer?
GL_NOOP : exact-integer?
GL_INVERT : exact-integer?
GL_AND : exact-integer?
GL_NAND : exact-integer?
GL_OR : exact-integer?
GL_NOR : exact-integer?
GL_XOR : exact-integer?
GL_EQUIV : exact-integer?
GL_AND_REVERSE : exact-integer?
GL_AND_INVERTED : exact-integer?
GL_OR_REVERSE : exact-integer?
GL_OR_INVERTED : exact-integer?
GL_STENCIL_TEST : exact-integer?
GL_STENCIL_WRITEMASK : exact-integer?
GL_STENCIL_BITS : exact-integer?
GL_STENCIL_FUNC : exact-integer?
GL_STENCIL_VALUE_MASK : exact-integer?
GL_STENCIL_REF : exact-integer?
GL_STENCIL_FAIL : exact-integer?
GL_STENCIL_PASS_DEPTH_PASS : exact-integer?
GL_STENCIL_PASS_DEPTH_FAIL : exact-integer?
GL_STENCIL_CLEAR_VALUE : exact-integer?
GL_STENCIL_INDEX : exact-integer?
GL_KEEP : exact-integer?
GL_REPLACE : exact-integer?
GL_INCR : exact-integer?
GL_DECR : exact-integer?
GL_NONE : exact-integer?
GL_LEFT : exact-integer?
GL_RIGHT : exact-integer?
GL_FRONT_LEFT : exact-integer?
GL_FRONT_RIGHT : exact-integer?
GL_BACK_LEFT : exact-integer?
GL_BACK_RIGHT : exact-integer?
GL_AUX0 : exact-integer?
GL_AUX1 : exact-integer?
GL_AUX2 : exact-integer?
GL_AUX3 : exact-integer?
GL_COLOR_INDEX : exact-integer?
GL_RED : exact-integer?
```

```
GL_GREEN : exact-integer?
GL_BLUE : exact-integer?
GL_ALPHA : exact-integer?
GL_LUMINANCE : exact-integer?
GL_LUMINANCE_ALPHA : exact-integer?
GL_ALPHA_BITS : exact-integer?
GL_RED_BITS : exact-integer?
GL_GREEN_BITS : exact-integer?
GL_BLUE_BITS : exact-integer?
GL_INDEX_BITS : exact-integer?
GL_SUBPIXEL_BITS : exact-integer?
GL_AUX_BUFFERS : exact-integer?
GL_READ_BUFFER : exact-integer?
GL_DRAW_BUFFER : exact-integer?
GL_DOUBLEBUFFER : exact-integer?
GL_STEREO : exact-integer?
GL_BITMAP : exact-integer?
GL_COLOR : exact-integer?
GL_DEPTH : exact-integer?
GL_STENCIL : exact-integer?
GL_DITHER : exact-integer?
GL_RGB : exact-integer?
GL_RGBA : exact-integer?
GL_MAX_LIST_NESTING : exact-integer?
GL_MAX_ATTRIB_STACK_DEPTH : exact-integer?
GL_MAX_MODELVIEW_STACK_DEPTH : exact-integer?
GL_MAX_NAME_STACK_DEPTH : exact-integer?
GL_MAX_PROJECTION_STACK_DEPTH : exact-integer?
GL_MAX_TEXTURE_STACK_DEPTH : exact-integer?
GL_MAX_EVAL_ORDER : exact-integer?
GL_MAX_LIGHTS : exact-integer?
GL_MAX_CLIP_PLANES : exact-integer?
GL_MAX_TEXTURE_SIZE : exact-integer?
GL_MAX_PIXEL_MAP_TABLE : exact-integer?
GL_MAX_VIEWPORT_DIMS : exact-integer?
GL_MAX_CLIENT_ATTRIB_STACK_DEPTH : exact-integer?
GL_ATTRIB_STACK_DEPTH : exact-integer?
GL_CLIENT_ATTRIB_STACK_DEPTH : exact-integer?
GL_COLOR_CLEAR_VALUE : exact-integer?
GL_COLOR_WRITEMASK : exact-integer?
GL_CURRENT_INDEX : exact-integer?
GL_CURRENT_COLOR : exact-integer?
GL_CURRENT_NORMAL : exact-integer?
GL_CURRENT_RASTER_COLOR : exact-integer?
```

GL_CURRENT_RASTER_DISTANCE : exact-integer?
GL_CURRENT_RASTER_INDEX : exact-integer?
GL_CURRENT_RASTER_POSITION : exact-integer?
GL_CURRENT_RASTER_TEXTURE_COORDS : exact-integer?
GL_CURRENT_RASTER_POSITION_VALID : exact-integer?
GL_CURRENT_TEXTURE_COORDS : exact-integer?
GL_INDEX_CLEAR_VALUE : exact-integer?
GL_INDEX_MODE : exact-integer?
GL_INDEX_WRITEMASK : exact-integer?
GL_MODELVIEW_MATRIX : exact-integer?
GL_MODELVIEW_STACK_DEPTH : exact-integer?
GL_NAME_STACK_DEPTH : exact-integer?
GL_PROJECTION_MATRIX : exact-integer?
GL_PROJECTION_STACK_DEPTH : exact-integer?
GL_RENDER_MODE : exact-integer?
GL_RGBA_MODE : exact-integer?
GL_TEXTURE_MATRIX : exact-integer?
GL_TEXTURE_STACK_DEPTH : exact-integer?
GL_VIEWPORT : exact-integer?
GL_AUTO_NORMAL : exact-integer?
GL_MAP1_COLOR_4 : exact-integer?
GL_MAP1_GRID_DOMAIN : exact-integer?
GL_MAP1_GRID_SEGMENTS : exact-integer?
GL_MAP1_INDEX : exact-integer?
GL_MAP1_NORMAL : exact-integer?
GL_MAP1_TEXTURE_COORD_1 : exact-integer?
GL_MAP1_TEXTURE_COORD_2 : exact-integer?
GL_MAP1_TEXTURE_COORD_3 : exact-integer?
GL_MAP1_TEXTURE_COORD_4 : exact-integer?
GL_MAP1_VERTEX_3 : exact-integer?
GL_MAP1_VERTEX_4 : exact-integer?
GL_MAP2_COLOR_4 : exact-integer?
GL_MAP2_GRID_DOMAIN : exact-integer?
GL_MAP2_GRID_SEGMENTS : exact-integer?
GL_MAP2_INDEX : exact-integer?
GL_MAP2_NORMAL : exact-integer?
GL_MAP2_TEXTURE_COORD_1 : exact-integer?
GL_MAP2_TEXTURE_COORD_2 : exact-integer?
GL_MAP2_TEXTURE_COORD_3 : exact-integer?
GL_MAP2_TEXTURE_COORD_4 : exact-integer?
GL_MAP2_VERTEX_3 : exact-integer?
GL_MAP2_VERTEX_4 : exact-integer?
GL_COEFF : exact-integer?
GL_DOMAIN : exact-integer?

```
GL_ORDER : exact-integer?
GL_FOG_HINT : exact-integer?
GL_LINE_SMOOTH_HINT : exact-integer?
GL_PERSPECTIVE_CORRECTION_HINT : exact-integer?
GL_POINT_SMOOTH_HINT : exact-integer?
GL_POLYGON_SMOOTH_HINT : exact-integer?
GL_DONT_CARE : exact-integer?
GL_FASTEST : exact-integer?
GL_NICEST : exact-integer?
GL_SCISSOR_TEST : exact-integer?
GL_SCISSOR_BOX : exact-integer?
GL_MAP_COLOR : exact-integer?
GL_MAP_STENCIL : exact-integer?
GL_INDEX_SHIFT : exact-integer?
GL_INDEX_OFFSET : exact-integer?
GL_RED_SCALE : exact-integer?
GL_RED_BIAS : exact-integer?
GL_GREEN_SCALE : exact-integer?
GL_GREEN_BIAS : exact-integer?
GL_BLUE_SCALE : exact-integer?
GL_BLUE_BIAS : exact-integer?
GL_ALPHA_SCALE : exact-integer?
GL_ALPHA_BIAS : exact-integer?
GL_DEPTH_SCALE : exact-integer?
GL_DEPTH_BIAS : exact-integer?
GL_PIXEL_MAP_S_TO_S_SIZE : exact-integer?
GL_PIXEL_MAP_I_TO_I_SIZE : exact-integer?
GL_PIXEL_MAP_I_TO_R_SIZE : exact-integer?
GL_PIXEL_MAP_I_TO_G_SIZE : exact-integer?
GL_PIXEL_MAP_I_TO_B_SIZE : exact-integer?
GL_PIXEL_MAP_I_TO_A_SIZE : exact-integer?
GL_PIXEL_MAP_R_TO_R_SIZE : exact-integer?
GL_PIXEL_MAP_G_TO_G_SIZE : exact-integer?
GL_PIXEL_MAP_B_TO_B_SIZE : exact-integer?
GL_PIXEL_MAP_A_TO_A_SIZE : exact-integer?
GL_PIXEL_MAP_S_TO_S : exact-integer?
GL_PIXEL_MAP_I_TO_I : exact-integer?
GL_PIXEL_MAP_I_TO_R : exact-integer?
GL_PIXEL_MAP_I_TO_G : exact-integer?
GL_PIXEL_MAP_I_TO_B : exact-integer?
GL_PIXEL_MAP_I_TO_A : exact-integer?
GL_PIXEL_MAP_R_TO_R : exact-integer?
GL_PIXEL_MAP_G_TO_G : exact-integer?
GL_PIXEL_MAP_B_TO_B : exact-integer?
```

GL_PIXEL_MAP_A_TO_A : exact-integer?
GL_PACK_ALIGNMENT : exact-integer?
GL_PACK_LSB_FIRST : exact-integer?
GL_PACK_ROW_LENGTH : exact-integer?
GL_PACK_SKIP_PIXELS : exact-integer?
GL_PACK_SKIP_ROWS : exact-integer?
GL_PACK_SWAP_BYTES : exact-integer?
GL_UNPACK_ALIGNMENT : exact-integer?
GL_UNPACK_LSB_FIRST : exact-integer?
GL_UNPACK_ROW_LENGTH : exact-integer?
GL_UNPACK_SKIP_PIXELS : exact-integer?
GL_UNPACK_SKIP_ROWS : exact-integer?
GL_UNPACK_SWAP_BYTES : exact-integer?
GL_ZOOM_X : exact-integer?
GL_ZOOM_Y : exact-integer?
GL_TEXTURE_ENV : exact-integer?
GL_TEXTURE_ENV_MODE : exact-integer?
GL_TEXTURE_1D : exact-integer?
GL_TEXTURE_2D : exact-integer?
GL_TEXTURE_WRAP_S : exact-integer?
GL_TEXTURE_WRAP_T : exact-integer?
GL_TEXTURE_MAG_FILTER : exact-integer?
GL_TEXTURE_MIN_FILTER : exact-integer?
GL_TEXTURE_ENV_COLOR : exact-integer?
GL_TEXTURE_GEN_S : exact-integer?
GL_TEXTURE_GEN_T : exact-integer?
GL_TEXTURE_GEN_MODE : exact-integer?
GL_TEXTURE_BORDER_COLOR : exact-integer?
GL_TEXTURE_WIDTH : exact-integer?
GL_TEXTURE_HEIGHT : exact-integer?
GL_TEXTURE_BORDER : exact-integer?
GL_TEXTURE_COMPONENTS : exact-integer?
GL_TEXTURE_RED_SIZE : exact-integer?
GL_TEXTURE_GREEN_SIZE : exact-integer?
GL_TEXTURE_BLUE_SIZE : exact-integer?
GL_TEXTURE_ALPHA_SIZE : exact-integer?
GL_TEXTURE_LUMINANCE_SIZE : exact-integer?
GL_TEXTURE_INTENSITY_SIZE : exact-integer?
GL_NEAREST_MIPMAP_NEAREST : exact-integer?
GL_NEAREST_MIPMAP_LINEAR : exact-integer?
GL_LINEAR_MIPMAP_NEAREST : exact-integer?
GL_LINEAR_MIPMAP_LINEAR : exact-integer?
GL_OBJECT_LINEAR : exact-integer?
GL_OBJECT_PLANE : exact-integer?

```
GL_EYE_LINEAR : exact-integer?
GL_EYE_PLANE : exact-integer?
GL_SPHERE_MAP : exact-integer?
GL_DECAL : exact-integer?
GL_MODULATE : exact-integer?
GL_NEAREST : exact-integer?
GL_REPEAT : exact-integer?
GL_CLAMP : exact-integer?
GL_S : exact-integer?
GL_T : exact-integer?
GL_R : exact-integer?
GL_Q : exact-integer?
GL_TEXTURE_GEN_R : exact-integer?
GL_TEXTURE_GEN_Q : exact-integer?
GL_VENDOR : exact-integer?
GL_RENDERER : exact-integer?
GL_VERSION : exact-integer?
GL_EXTENSIONS : exact-integer?
GL_NO_ERROR : exact-integer?
GL_INVALID_VALUE : exact-integer?
GL_INVALID_ENUM : exact-integer?
GL_INVALID_OPERATION : exact-integer?
GL_STACK_OVERFLOW : exact-integer?
GL_STACK_UNDERFLOW : exact-integer?
GL_OUT_OF_MEMORY : exact-integer?
GL_CURRENT_BIT : exact-integer?
GL_POINT_BIT : exact-integer?
GL_LINE_BIT : exact-integer?
GL_POLYGON_BIT : exact-integer?
GL_POLYGON_STIPPLE_BIT : exact-integer?
GL_PIXEL_MODE_BIT : exact-integer?
GL_LIGHTING_BIT : exact-integer?
GL_FOG_BIT : exact-integer?
GL_DEPTH_BUFFER_BIT : exact-integer?
GL_ACCUM_BUFFER_BIT : exact-integer?
GL_STENCIL_BUFFER_BIT : exact-integer?
GL_VIEWPORT_BIT : exact-integer?
GL_TRANSFORM_BIT : exact-integer?
GL_ENABLE_BIT : exact-integer?
GL_COLOR_BUFFER_BIT : exact-integer?
GL_HINT_BIT : exact-integer?
GL_EVAL_BIT : exact-integer?
GL_LIST_BIT : exact-integer?
GL_TEXTURE_BIT : exact-integer?
```

GL_SCISSOR_BIT : exact-integer?
GL_ALL_ATTRIB_BITS : exact-integer?
GL_PROXY_TEXTURE_1D : exact-integer?
GL_PROXY_TEXTURE_2D : exact-integer?
GL_TEXTURE_PRIORITY : exact-integer?
GL_TEXTURE_RESIDENT : exact-integer?
GL_TEXTURE_BINDING_1D : exact-integer?
GL_TEXTURE_BINDING_2D : exact-integer?
GL_TEXTURE_INTERNAL_FORMAT : exact-integer?
GL_ALPHA4 : exact-integer?
GL_ALPHA8 : exact-integer?
GL_ALPHA12 : exact-integer?
GL_ALPHA16 : exact-integer?
GL_LUMINANCE4 : exact-integer?
GL_LUMINANCE8 : exact-integer?
GL_LUMINANCE12 : exact-integer?
GL_LUMINANCE16 : exact-integer?
GL_LUMINANCE4_ALPHA4 : exact-integer?
GL_LUMINANCE6_ALPHA2 : exact-integer?
GL_LUMINANCE8_ALPHA8 : exact-integer?
GL_LUMINANCE12_ALPHA4 : exact-integer?
GL_LUMINANCE12_ALPHA12 : exact-integer?
GL_LUMINANCE16_ALPHA16 : exact-integer?
GL_INTENSITY : exact-integer?
GL_INTENSITY4 : exact-integer?
GL_INTENSITY8 : exact-integer?
GL_INTENSITY12 : exact-integer?
GL_INTENSITY16 : exact-integer?
GL_R3_G3_B2 : exact-integer?
GL_RGB4 : exact-integer?
GL_RGB5 : exact-integer?
GL_RGB8 : exact-integer?
GL_RGB10 : exact-integer?
GL_RGB12 : exact-integer?
GL_RGB16 : exact-integer?
GL_RGBA2 : exact-integer?
GL_RGBA4 : exact-integer?
GL_RGB5_A1 : exact-integer?
GL_RGBA8 : exact-integer?
GL_RGB10_A2 : exact-integer?
GL_RGBA12 : exact-integer?
GL_RGBA16 : exact-integer?
GL_CLIENT_PIXEL_STORE_BIT : exact-integer?
GL_CLIENT_VERTEX_ARRAY_BIT : exact-integer?

GL_ALL_CLIENT_ATTRIB_BITS : exact-integer?
GL_CLIENT_ALL_ATTRIB_BITS : exact-integer?
GL_UNSIGNED_BYTE_3_3_2 : exact-integer?
GL_UNSIGNED_SHORT_4_4_4_4 : exact-integer?
GL_UNSIGNED_SHORT_5_5_5_1 : exact-integer?
GL_UNSIGNED_INT_8_8_8_8 : exact-integer?
GL_UNSIGNED_INT_10_10_10_2 : exact-integer?
GL_RESCALE_NORMAL : exact-integer?
GL_TEXTURE_BINDING_3D : exact-integer?
GL_PACK_SKIP_IMAGES : exact-integer?
GL_PACK_IMAGE_HEIGHT : exact-integer?
GL_UNPACK_SKIP_IMAGES : exact-integer?
GL_UNPACK_IMAGE_HEIGHT : exact-integer?
GL_TEXTURE_3D : exact-integer?
GL_PROXY_TEXTURE_3D : exact-integer?
GL_TEXTURE_DEPTH : exact-integer?
GL_TEXTURE_WRAP_R : exact-integer?
GL_MAX_3D_TEXTURE_SIZE : exact-integer?
GL_UNSIGNED_BYTE_2_3_3_REV : exact-integer?
GL_UNSIGNED_SHORT_5_6_5 : exact-integer?
GL_UNSIGNED_SHORT_5_6_5_REV : exact-integer?
GL_UNSIGNED_SHORT_4_4_4_4_REV : exact-integer?
GL_UNSIGNED_SHORT_1_5_5_5_REV : exact-integer?
GL_UNSIGNED_INT_8_8_8_8_REV : exact-integer?
GL_UNSIGNED_INT_2_10_10_10_REV : exact-integer?
GL_BGR : exact-integer?
GL_BGRA : exact-integer?
GL_MAX_ELEMENTS_VERTICES : exact-integer?
GL_MAX_ELEMENTS_INDICES : exact-integer?
GL_CLAMP_TO_EDGE : exact-integer?
GL_TEXTURE_MIN_LOD : exact-integer?
GL_TEXTURE_MAX_LOD : exact-integer?
GL_TEXTURE_BASE_LEVEL : exact-integer?
GL_TEXTURE_MAX_LEVEL : exact-integer?
GL_LIGHT_MODEL_COLOR_CONTROL : exact-integer?
GL_SINGLE_COLOR : exact-integer?
GL_SEPARATE_SPECULAR_COLOR : exact-integer?
GL_SMOOTH_POINT_SIZE_RANGE : exact-integer?
GL_SMOOTH_POINT_SIZE_GRANULARITY : exact-integer?
GL_SMOOTH_LINE_WIDTH_RANGE : exact-integer?
GL_SMOOTH_LINE_WIDTH_GRANULARITY : exact-integer?
GL_ALIASED_POINT_SIZE_RANGE : exact-integer?
GL_ALIASED_LINE_WIDTH_RANGE : exact-integer?
GL_CONSTANT_COLOR : exact-integer?

GL_ONE_MINUS_CONSTANT_COLOR : exact-integer?
GL_CONSTANT_ALPHA : exact-integer?
GL_ONE_MINUS_CONSTANT_ALPHA : exact-integer?
GL_BLEND_COLOR : exact-integer?
GL_FUNC_ADD : exact-integer?
GL_MIN : exact-integer?
GL_MAX : exact-integer?
GL_BLEND_EQUATION : exact-integer?
GL_FUNC_SUBTRACT : exact-integer?
GL_FUNC_REVERSE_SUBTRACT : exact-integer?
GL_CONVOLUTION_1D : exact-integer?
GL_CONVOLUTION_2D : exact-integer?
GL_SEPARABLE_2D : exact-integer?
GL_CONVOLUTION_BORDER_MODE : exact-integer?
GL_CONVOLUTION_FILTER_SCALE : exact-integer?
GL_CONVOLUTION_FILTER_BIAS : exact-integer?
GL_REDUCE : exact-integer?
GL_CONVOLUTION_FORMAT : exact-integer?
GL_CONVOLUTION_WIDTH : exact-integer?
GL_CONVOLUTION_HEIGHT : exact-integer?
GL_MAX_CONVOLUTION_WIDTH : exact-integer?
GL_MAX_CONVOLUTION_HEIGHT : exact-integer?
GL_POST_CONVOLUTION_RED_SCALE : exact-integer?
GL_POST_CONVOLUTION_GREEN_SCALE : exact-integer?
GL_POST_CONVOLUTION_BLUE_SCALE : exact-integer?
GL_POST_CONVOLUTION_ALPHA_SCALE : exact-integer?
GL_POST_CONVOLUTION_RED_BIAS : exact-integer?
GL_POST_CONVOLUTION_GREEN_BIAS : exact-integer?
GL_POST_CONVOLUTION_BLUE_BIAS : exact-integer?
GL_POST_CONVOLUTION_ALPHA_BIAS : exact-integer?
GL_HISTOGRAM : exact-integer?
GL_PROXY_HISTOGRAM : exact-integer?
GL_HISTOGRAM_WIDTH : exact-integer?
GL_HISTOGRAM_FORMAT : exact-integer?
GL_HISTOGRAM_RED_SIZE : exact-integer?
GL_HISTOGRAM_GREEN_SIZE : exact-integer?
GL_HISTOGRAM_BLUE_SIZE : exact-integer?
GL_HISTOGRAM_ALPHA_SIZE : exact-integer?
GL_HISTOGRAM_LUMINANCE_SIZE : exact-integer?
GL_HISTOGRAM_SINK : exact-integer?
GL_MINMAX : exact-integer?
GL_MINMAX_FORMAT : exact-integer?
GL_MINMAX_SINK : exact-integer?
GL_TABLE_TOO_LARGE : exact-integer?

GL_COLOR_MATRIX : exact-integer?
GL_COLOR_MATRIX_STACK_DEPTH : exact-integer?
GL_MAX_COLOR_MATRIX_STACK_DEPTH : exact-integer?
GL_POST_COLOR_MATRIX_RED_SCALE : exact-integer?
GL_POST_COLOR_MATRIX_GREEN_SCALE : exact-integer?
GL_POST_COLOR_MATRIX_BLUE_SCALE : exact-integer?
GL_POST_COLOR_MATRIX_ALPHA_SCALE : exact-integer?
GL_POST_COLOR_MATRIX_RED_BIAS : exact-integer?
GL_POST_COLOR_MATRIX_GREEN_BIAS : exact-integer?
GL_POST_COLOR_MATRIX_BLUE_BIAS : exact-integer?
GL_POST_COLOR_MATRIX_ALPHA_BIAS : exact-integer?
GL_COLOR_TABLE : exact-integer?
GL_POST_CONVOLUTION_COLOR_TABLE : exact-integer?
GL_POST_COLOR_MATRIX_COLOR_TABLE : exact-integer?
GL_PROXY_COLOR_TABLE : exact-integer?
GL_PROXY_POST_CONVOLUTION_COLOR_TABLE : exact-integer?
GL_PROXY_POST_COLOR_MATRIX_COLOR_TABLE : exact-integer?
GL_COLOR_TABLE_SCALE : exact-integer?
GL_COLOR_TABLE_BIAS : exact-integer?
GL_COLOR_TABLE_FORMAT : exact-integer?
GL_COLOR_TABLE_WIDTH : exact-integer?
GL_COLOR_TABLE_RED_SIZE : exact-integer?
GL_COLOR_TABLE_GREEN_SIZE : exact-integer?
GL_COLOR_TABLE_BLUE_SIZE : exact-integer?
GL_COLOR_TABLE_ALPHA_SIZE : exact-integer?
GL_COLOR_TABLE_LUMINANCE_SIZE : exact-integer?
GL_COLOR_TABLE_INTENSITY_SIZE : exact-integer?
GL_CONSTANT_BORDER : exact-integer?
GL_REPLICATE_BORDER : exact-integer?
GL_CONVOLUTION_BORDER_COLOR : exact-integer?
GL_TEXTURE0 : exact-integer?
GL_TEXTURE1 : exact-integer?
GL_TEXTURE2 : exact-integer?
GL_TEXTURE3 : exact-integer?
GL_TEXTURE4 : exact-integer?
GL_TEXTURE5 : exact-integer?
GL_TEXTURE6 : exact-integer?
GL_TEXTURE7 : exact-integer?
GL_TEXTURE8 : exact-integer?
GL_TEXTURE9 : exact-integer?
GL_TEXTURE10 : exact-integer?
GL_TEXTURE11 : exact-integer?
GL_TEXTURE12 : exact-integer?
GL_TEXTURE13 : exact-integer?

```
GL_TEXTURE14 : exact-integer?
GL_TEXTURE15 : exact-integer?
GL_TEXTURE16 : exact-integer?
GL_TEXTURE17 : exact-integer?
GL_TEXTURE18 : exact-integer?
GL_TEXTURE19 : exact-integer?
GL_TEXTURE20 : exact-integer?
GL_TEXTURE21 : exact-integer?
GL_TEXTURE22 : exact-integer?
GL_TEXTURE23 : exact-integer?
GL_TEXTURE24 : exact-integer?
GL_TEXTURE25 : exact-integer?
GL_TEXTURE26 : exact-integer?
GL_TEXTURE27 : exact-integer?
GL_TEXTURE28 : exact-integer?
GL_TEXTURE29 : exact-integer?
GL_TEXTURE30 : exact-integer?
GL_TEXTURE31 : exact-integer?
GL_ACTIVE_TEXTURE : exact-integer?
GL_CLIENT_ACTIVE_TEXTURE : exact-integer?
GL_MAX_TEXTURE_UNITS : exact-integer?
GL_TRANSPOSE_MODELVIEW_MATRIX : exact-integer?
GL_TRANSPOSE_PROJECTION_MATRIX : exact-integer?
GL_TRANSPOSE_TEXTURE_MATRIX : exact-integer?
GL_TRANSPOSE_COLOR_MATRIX : exact-integer?
GL_MULTISAMPLE : exact-integer?
GL_SAMPLE_ALPHA_TO_COVERAGE : exact-integer?
GL_SAMPLE_ALPHA_TO_ONE : exact-integer?
GL_SAMPLE_COVERAGE : exact-integer?
GL_SAMPLE_BUFFERS : exact-integer?
GL_SAMPLES : exact-integer?
GL_SAMPLE_COVERAGE_VALUE : exact-integer?
GL_SAMPLE_COVERAGE_INVERT : exact-integer?
GL_MULTISAMPLE_BIT : exact-integer?
GL_NORMAL_MAP : exact-integer?
GL_REFLECTION_MAP : exact-integer?
GL_TEXTURE_CUBE_MAP : exact-integer?
GL_TEXTURE_BINDING_CUBE_MAP : exact-integer?
GL_TEXTURE_CUBE_MAP_POSITIVE_X : exact-integer?
GL_TEXTURE_CUBE_MAP_NEGATIVE_X : exact-integer?
GL_TEXTURE_CUBE_MAP_POSITIVE_Y : exact-integer?
GL_TEXTURE_CUBE_MAP_NEGATIVE_Y : exact-integer?
GL_TEXTURE_CUBE_MAP_POSITIVE_Z : exact-integer?
GL_TEXTURE_CUBE_MAP_NEGATIVE_Z : exact-integer?
```

GL_PROXY_TEXTURE_CUBE_MAP : exact-integer?
GL_MAX_CUBE_MAP_TEXTURE_SIZE : exact-integer?
GL_COMPRESSED_ALPHA : exact-integer?
GL_COMPRESSED_LUMINANCE : exact-integer?
GL_COMPRESSED_LUMINANCE_ALPHA : exact-integer?
GL_COMPRESSED_INTENSITY : exact-integer?
GL_COMPRESSED_RGB : exact-integer?
GL_COMPRESSED_RGBA : exact-integer?
GL_TEXTURE_COMPRESSION_HINT : exact-integer?
GL_TEXTURE_COMPRESSED_IMAGE_SIZE : exact-integer?
GL_TEXTURE_COMPRESSED : exact-integer?
GL_NUM_COMPRESSED_TEXTURE_FORMATS : exact-integer?
GL_COMPRESSED_TEXTURE_FORMATS : exact-integer?
GL_CLAMP_TO_BORDER : exact-integer?
GL_COMBINE : exact-integer?
GL_COMBINE_RGB : exact-integer?
GL_COMBINE_ALPHA : exact-integer?
GL_SOURCE0_RGB : exact-integer?
GL_SOURCE1_RGB : exact-integer?
GL_SOURCE2_RGB : exact-integer?
GL_SOURCE0_ALPHA : exact-integer?
GL_SOURCE1_ALPHA : exact-integer?
GL_SOURCE2_ALPHA : exact-integer?
GL_OPERAND0_RGB : exact-integer?
GL_OPERAND1_RGB : exact-integer?
GL_OPERAND2_RGB : exact-integer?
GL_OPERAND0_ALPHA : exact-integer?
GL_OPERAND1_ALPHA : exact-integer?
GL_OPERAND2_ALPHA : exact-integer?
GL_RGB_SCALE : exact-integer?
GL_ADD_SIGNED : exact-integer?
GL_INTERPOLATE : exact-integer?
GL_SUBTRACT : exact-integer?
GL_CONSTANT : exact-integer?
GL_PRIMARY_COLOR : exact-integer?
GL_PREVIOUS : exact-integer?
GL_DOT3_RGB : exact-integer?
GL_DOT3_RGBA : exact-integer?
GL_BLEND_DST_RGB : exact-integer?
GL_BLEND_SRC_RGB : exact-integer?
GL_BLEND_DST_ALPHA : exact-integer?
GL_BLEND_SRC_ALPHA : exact-integer?
GL_POINT_SIZE_MIN : exact-integer?
GL_POINT_SIZE_MAX : exact-integer?

GL_POINT_FADE_THRESHOLD_SIZE : exact-integer?
GL_POINT_DISTANCE_ATTENUATION : exact-integer?
GL_GENERATE_MIPMAP : exact-integer?
GL_GENERATE_MIPMAP_HINT : exact-integer?
GL_DEPTH_COMPONENT16 : exact-integer?
GL_DEPTH_COMPONENT24 : exact-integer?
GL_DEPTH_COMPONENT32 : exact-integer?
GL_MIRRORED_REPEAT : exact-integer?
GL_FOG_COORDINATE_SOURCE : exact-integer?
GL_FOG_COORDINATE : exact-integer?
GL_FRAGMENT_DEPTH : exact-integer?
GL_CURRENT_FOG_COORDINATE : exact-integer?
GL_FOG_COORDINATE_ARRAY_TYPE : exact-integer?
GL_FOG_COORDINATE_ARRAY_STRIDE : exact-integer?
GL_FOG_COORDINATE_ARRAY_POINTER : exact-integer?
GL_FOG_COORDINATE_ARRAY : exact-integer?
GL_COLOR_SUM : exact-integer?
GL_CURRENT_SECONDARY_COLOR : exact-integer?
GL_SECONDARY_COLOR_ARRAY_SIZE : exact-integer?
GL_SECONDARY_COLOR_ARRAY_TYPE : exact-integer?
GL_SECONDARY_COLOR_ARRAY_STRIDE : exact-integer?
GL_SECONDARY_COLOR_ARRAY_POINTER : exact-integer?
GL_SECONDARY_COLOR_ARRAY : exact-integer?
GL_MAX_TEXTURE_LOD_BIAS : exact-integer?
GL_TEXTURE_FILTER_CONTROL : exact-integer?
GL_TEXTURE_LOD_BIAS : exact-integer?
GL_INCR_WRAP : exact-integer?
GL DECR_WRAP : exact-integer?
GL_TEXTURE_DEPTH_SIZE : exact-integer?
GL_DEPTH_TEXTURE_MODE : exact-integer?
GL_TEXTURE_COMPARE_MODE : exact-integer?
GL_TEXTURE_COMPARE_FUNC : exact-integer?
GL_COMPARE_R_TO_TEXTURE : exact-integer?
GL_BUFFER_SIZE : exact-integer?
GL_BUFFER_USAGE : exact-integer?
GL_QUERY_COUNTER_BITS : exact-integer?
GL_CURRENT_QUERY : exact-integer?
GL_QUERY_RESULT : exact-integer?
GL_QUERY_RESULT_AVAILABLE : exact-integer?
GL_ARRAY_BUFFER : exact-integer?
GL_ELEMENT_ARRAY_BUFFER : exact-integer?
GL_ARRAY_BUFFER_BINDING : exact-integer?
GL_ELEMENT_ARRAY_BUFFER_BINDING : exact-integer?
GL_VERTEX_ARRAY_BUFFER_BINDING : exact-integer?

GL_NORMAL_ARRAY_BUFFER_BINDING : exact-integer?
GL_COLOR_ARRAY_BUFFER_BINDING : exact-integer?
GL_INDEX_ARRAY_BUFFER_BINDING : exact-integer?
GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING : exact-integer?
GL_EDGE_FLAG_ARRAY_BUFFER_BINDING : exact-integer?
GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING : exact-integer?
GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING : exact-integer?
GL_WEIGHT_ARRAY_BUFFER_BINDING : exact-integer?
GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING : exact-integer?
GL_READ_ONLY : exact-integer?
GL_WRITE_ONLY : exact-integer?
GL_READ_WRITE : exact-integer?
GL_BUFFER_ACCESS : exact-integer?
GL_BUFFER_MAPPED : exact-integer?
GL_BUFFER_MAP_POINTER : exact-integer?
GL_STREAM_DRAW : exact-integer?
GL_STREAM_READ : exact-integer?
GL_STREAM_COPY : exact-integer?
GL_STATIC_DRAW : exact-integer?
GL_STATIC_READ : exact-integer?
GL_STATIC_COPY : exact-integer?
GL_DYNAMIC_DRAW : exact-integer?
GL_DYNAMIC_READ : exact-integer?
GL_DYNAMIC_COPY : exact-integer?
GL_SAMPLES_PASSED : exact-integer?
GL_FOG_COORD_SRC : exact-integer?
GL_FOG_COORD : exact-integer?
GL_CURRENT_FOG_COORD : exact-integer?
GL_FOG_COORD_ARRAY_TYPE : exact-integer?
GL_FOG_COORD_ARRAY_STRIDE : exact-integer?
GL_FOG_COORD_ARRAY_POINTER : exact-integer?
GL_FOG_COORD_ARRAY : exact-integer?
GL_FOG_COORD_ARRAY_BUFFER_BINDING : exact-integer?
GL_SRC0_RGB : exact-integer?
GL_SRC1_RGB : exact-integer?
GL_SRC2_RGB : exact-integer?
GL_SRC0_ALPHA : exact-integer?
GL_SRC1_ALPHA : exact-integer?
GL_SRC2_ALPHA : exact-integer?
GLU_FALSE : exact-integer?
GLU_TRUE : exact-integer?
GLU_VERSION : exact-integer?
GLU_EXTENSIONS : exact-integer?
GLU_INVALID_ENUM : exact-integer?

GLU_INVALID_VALUE : exact-integer?
GLU_OUT_OF_MEMORY : exact-integer?
GLU_INVALID_OPERATION : exact-integer?
GLU_OUTLINE_POLYGON : exact-integer?
GLU_OUTLINE_PATCH : exact-integer?
GLU_NURBS_ERROR : exact-integer?
GLU_ERROR : exact-integer?
GLU_NURBS_BEGIN : exact-integer?
GLU_NURBS_BEGIN_EXT : exact-integer?
GLU_NURBS_VERTEX : exact-integer?
GLU_NURBS_VERTEX_EXT : exact-integer?
GLU_NURBS_NORMAL : exact-integer?
GLU_NURBS_NORMAL_EXT : exact-integer?
GLU_NURBS_COLOR : exact-integer?
GLU_NURBS_COLOR_EXT : exact-integer?
GLU_NURBS_TEXTURE_COORD : exact-integer?
GLU_NURBS_TEX_COORD_EXT : exact-integer?
GLU_NURBS_END : exact-integer?
GLU_NURBS_END_EXT : exact-integer?
GLU_NURBS_BEGIN_DATA : exact-integer?
GLU_NURBS_BEGIN_DATA_EXT : exact-integer?
GLU_NURBS_VERTEX_DATA : exact-integer?
GLU_NURBS_VERTEX_DATA_EXT : exact-integer?
GLU_NURBS_NORMAL_DATA : exact-integer?
GLU_NURBS_NORMAL_DATA_EXT : exact-integer?
GLU_NURBS_COLOR_DATA : exact-integer?
GLU_NURBS_COLOR_DATA_EXT : exact-integer?
GLU_NURBS_TEXTURE_COORD_DATA : exact-integer?
GLU_NURBS_TEX_COORD_DATA_EXT : exact-integer?
GLU_NURBS_END_DATA : exact-integer?
GLU_NURBS_END_DATA_EXT : exact-integer?
GLU_NURBS_ERROR1 : exact-integer?
GLU_NURBS_ERROR2 : exact-integer?
GLU_NURBS_ERROR3 : exact-integer?
GLU_NURBS_ERROR4 : exact-integer?
GLU_NURBS_ERROR5 : exact-integer?
GLU_NURBS_ERROR6 : exact-integer?
GLU_NURBS_ERROR7 : exact-integer?
GLU_NURBS_ERROR8 : exact-integer?
GLU_NURBS_ERROR9 : exact-integer?
GLU_NURBS_ERROR10 : exact-integer?
GLU_NURBS_ERROR11 : exact-integer?
GLU_NURBS_ERROR12 : exact-integer?
GLU_NURBS_ERROR13 : exact-integer?

GLU_NURBS_ERROR14 : exact-integer?
GLU_NURBS_ERROR15 : exact-integer?
GLU_NURBS_ERROR16 : exact-integer?
GLU_NURBS_ERROR17 : exact-integer?
GLU_NURBS_ERROR18 : exact-integer?
GLU_NURBS_ERROR19 : exact-integer?
GLU_NURBS_ERROR20 : exact-integer?
GLU_NURBS_ERROR21 : exact-integer?
GLU_NURBS_ERROR22 : exact-integer?
GLU_NURBS_ERROR23 : exact-integer?
GLU_NURBS_ERROR24 : exact-integer?
GLU_NURBS_ERROR25 : exact-integer?
GLU_NURBS_ERROR26 : exact-integer?
GLU_NURBS_ERROR27 : exact-integer?
GLU_NURBS_ERROR28 : exact-integer?
GLU_NURBS_ERROR29 : exact-integer?
GLU_NURBS_ERROR30 : exact-integer?
GLU_NURBS_ERROR31 : exact-integer?
GLU_NURBS_ERROR32 : exact-integer?
GLU_NURBS_ERROR33 : exact-integer?
GLU_NURBS_ERROR34 : exact-integer?
GLU_NURBS_ERROR35 : exact-integer?
GLU_NURBS_ERROR36 : exact-integer?
GLU_NURBS_ERROR37 : exact-integer?
GLU_AUTO_LOAD_MATRIX : exact-integer?
GLU_CULLING : exact-integer?
GLU_SAMPLING_TOLERANCE : exact-integer?
GLU_DISPLAY_MODE : exact-integer?
GLU_PARAMETRIC_TOLERANCE : exact-integer?
GLU_SAMPLING_METHOD : exact-integer?
GLU_U_STEP : exact-integer?
GLU_V_STEP : exact-integer?
GLU_NURBS_MODE : exact-integer?
GLU_NURBS_MODE_EXT : exact-integer?
GLU_NURBS_TESSELLATOR : exact-integer?
GLU_NURBS_TESSELLATOR_EXT : exact-integer?
GLU_NURBS_RENDERER : exact-integer?
GLU_NURBS_RENDERER_EXT : exact-integer?
GLU_OBJECT_PARAMETRIC_ERROR : exact-integer?
GLU_OBJECT_PARAMETRIC_ERROR_EXT : exact-integer?
GLU_OBJECT_PATH_LENGTH : exact-integer?
GLU_OBJECT_PATH_LENGTH_EXT : exact-integer?
GLU_PATH_LENGTH : exact-integer?
GLU_PARAMETRIC_ERROR : exact-integer?

GLU_DOMAIN_DISTANCE : exact-integer?
GLU_MAP1_TRIM_2 : exact-integer?
GLU_MAP1_TRIM_3 : exact-integer?
GLU_POINT : exact-integer?
GLU_LINE : exact-integer?
GLU_FILL : exact-integer?
GLU_SILHOUETTE : exact-integer?
GLU_SMOOTH : exact-integer?
GLU_FLAT : exact-integer?
GLU_NONE : exact-integer?
GLU_OUTSIDE : exact-integer?
GLU_INSIDE : exact-integer?
GLU_TESS_BEGIN : exact-integer?
GLU_BEGIN : exact-integer?
GLU_TESS_VERTEX : exact-integer?
GLU_VERTEX : exact-integer?
GLU_TESS_END : exact-integer?
GLU_END : exact-integer?
GLU_TESS_ERROR : exact-integer?
GLU_TESS_EDGE_FLAG : exact-integer?
GLU_EDGE_FLAG : exact-integer?
GLU_TESS_COMBINE : exact-integer?
GLU_TESS_BEGIN_DATA : exact-integer?
GLU_TESS_VERTEX_DATA : exact-integer?
GLU_TESS_END_DATA : exact-integer?
GLU_TESS_ERROR_DATA : exact-integer?
GLU_TESS_EDGE_FLAG_DATA : exact-integer?
GLU_TESS_COMBINE_DATA : exact-integer?
GLU_CW : exact-integer?
GLU_CCW : exact-integer?
GLU_INTERIOR : exact-integer?
GLU_EXTERIOR : exact-integer?
GLU_UNKNOWN : exact-integer?
GLU_TESS_WINDING_RULE : exact-integer?
GLU_TESS_BOUNDARY_ONLY : exact-integer?
GLU_TESS_TOLERANCE : exact-integer?
GLU_TESS_ERROR1 : exact-integer?
GLU_TESS_ERROR2 : exact-integer?
GLU_TESS_ERROR3 : exact-integer?
GLU_TESS_ERROR4 : exact-integer?
GLU_TESS_ERROR5 : exact-integer?
GLU_TESS_ERROR6 : exact-integer?
GLU_TESS_ERROR7 : exact-integer?
GLU_TESS_ERROR8 : exact-integer?

```
GLU_TESS_MISSING_BEGIN_POLYGON : exact-integer?
GLU_TESS_MISSING_BEGIN_CONTOUR : exact-integer?
GLU_TESS_MISSING_END_POLYGON : exact-integer?
GLU_TESS_MISSING_END_CONTOUR : exact-integer?
GLU_TESS_COORD_TOO_LARGE : exact-integer?
GLU_TESS_NEED_COMBINE_CALLBACK : exact-integer?
GLU_TESS_WINDING_ODD : exact-integer?
GLU_TESS_WINDING_NONZERO : exact-integer?
GLU_TESS_WINDING_POSITIVE : exact-integer?
GLU_TESS_WINDING_NEGATIVE : exact-integer?
GLU_TESS_WINDING_ABS_GEQ_TWO : exact-integer?
GLU_TESS_MAX_COORD : real?
```

All OpenGL-defined constants.

```
(feedback-buffer->gl-float-vector buf) → gl-float-vector?
  buf : feedback-buffer-object?
```

Converts a result from `glFeedbackBuffer` to a vector.

```
(select-buffer->gl-uint-vector buf) → gl-uint-vector?
  buf : select-buffer-object?
```

Converts a result from `glSelectBuffer` to a vector.

3 Racket-Style OpenGL

```
(require sgl)      package: sgl
```

The functions in `sgl` use Racket style names instead of C style names. To convert a C OpenGL name to a Racket OpenGL name, change the `gl` prefix to `gl-`, separate adjacent words with hyphens, and convert to all lower case. Functions that have several variants to accommodate different numbers and types of arguments are collapsed into one or two functions in `sgl`. For example, `sgl` provides two vertex functions: `vertex` and `vertex-v`. The `vertex` function accepts 2, 3 or 4 numerical arguments, and the `vertex-v` function accepts `gl-vectors` of length 2, 3 or 4. The C language OpenGL interface, in contrast, has 24 vertex functions: `glVertex3i`, `glVertex4fv`, etc.

Functions in `sgl` take symbols instead of integers for `GLenum` arguments. Each function checks that the given symbol is an acceptable argument and raises an exception if it is not. Given the name of a C-language `#define` constant, determine the corresponding symbolic argument by removing the leading `GL_`, converting the letters to lower-case and replacing each `_` with `-`. For example, `GL_TRIANGLES` becomes `'triangles`, and `GL_TRIANGLE_STRIP` becomes `'triangle-strip`. Additionally, the functions check the length of any array arguments to ensure that OpenGL does not attempt to write or read after the array.

The `sgl` module is not as complete as the `sgl/gl` module.

Examples:

```
(require sgl sgl/gl-vectors)
(gl-begin 'triangles)
(gl-vertex 1 2 3)
(gl-vertex-v (gl-float-vector 1 2 3 4))
(gl-end)
```

```
(struct gl-selection-record (min-z max-z stack)
  #:extra-constructor-name make-gl-selection-record)
min-z : real?
max-z : real?
stack : ....
```

Represents a selection.

```
gl-accum : procedure?
gl-active-texture : procedure?
gl-alpha-func : procedure?
gl-begin : procedure?
gl-begin-query : procedure?
```

gl-blend-color : procedure?
gl-blend-equation : procedure?
gl-blend-func : procedure?
gl-blend-func-separate : procedure?
gl-call-list : procedure?
gl-check-extension : procedure?
gl-clear : procedure?
gl-clear-accum : procedure?
gl-clear-color : procedure?
gl-clear-depth : procedure?
gl-clear-index : procedure?
gl-clear-stencil : procedure?
gl-clip-plane : procedure?
gl-color : procedure?
gl-color-mask : procedure?
gl-color-material : procedure?
gl-color-v : procedure?
gl-copy-pixels : procedure?
gl-cull-face : procedure?
gl-cylinder : procedure?
gl-delete-lists : procedure?
gl-delete-queries : procedure?
gl-depth-func : procedure?
gl-depth-mask : procedure?
gl-depth-range : procedure?
gl-disable : procedure?
gl-disk : procedure?
gl-edge-flag : procedure?
gl-enable : procedure?
gl-end : procedure?
gl-end-list : procedure?
gl-end-query : procedure?
gl-eval-coord : procedure?
gl-eval-coord-v : procedure?
gl-eval-mesh : procedure?
gl-eval-point : procedure?
gl-feedback-buffer->gl-float-vector : procedure?
gl-finish : procedure?
gl-flush : procedure?
gl-front-face : procedure?
gl-frustum : procedure?
gl-gen-lists : procedure?
gl-gen-queries : procedure?
gl-get-error : procedure?

```
gl-get-string : procedure?  
gl-hint : procedure?  
gl-index : procedure?  
gl-index-mask : procedure?  
gl-index-v : procedure?  
gl-init-names : procedure?  
gl-is-buffer : procedure?  
gl-is-enabled : procedure?  
gl-is-list : procedure?  
gl-is-query : procedure?  
gl-light : procedure?  
gl-light-model : procedure?  
gl-light-model-v : procedure?  
gl-light-v : procedure?  
gl-line-stipple : procedure?  
gl-line-width : procedure?  
gl-list-base : procedure?  
gl-load-identity : procedure?  
gl-load-matrix : procedure?  
gl-load-name : procedure?  
gl-load-transpose-matrix : procedure?  
gl-look-at : procedure?  
gl-map-grid : procedure?  
gl-material : procedure?  
gl-material-v : procedure?  
gl-matrix-mode : procedure?  
gl-mult-matrix : procedure?  
gl-mult-transpose-matrix : procedure?  
gl-multi-tex-coord : procedure?  
gl-multi-tex-coord-v : procedure?  
gl-new-list : procedure?  
gl-new-quadric : procedure?  
gl-normal : procedure?  
gl-normal-v : procedure?  
gl-ortho : procedure?  
gl-ortho-2d : procedure?  
gl-partial-disk : procedure?  
gl-pass-through : procedure?  
gl-perspective : procedure?  
gl-pick-matrix : procedure?  
gl-pixel-store : procedure?  
gl-point-parameter : procedure?  
gl-point-parameter-v : procedure?  
gl-point-size : procedure?
```

```
gl-polygon-mode : procedure?  
gl-polygon-offset : procedure?  
gl-pop-attrib : procedure?  
gl-pop-client-attrib : procedure?  
gl-pop-matrix : procedure?  
gl-pop-name : procedure?  
gl-project : procedure?  
gl-push-matrix : procedure?  
gl-push-name : procedure?  
gl-quadric-draw-style : procedure?  
gl-quadric-normals : procedure?  
gl-quadric-orientation : procedure?  
gl-quadric-texture : procedure?  
gl-raster-pos : procedure?  
gl-raster-pos-v : procedure?  
gl-rect : procedure?  
gl-rect-v : procedure?  
gl-render-mode : procedure?  
gl-rotate : procedure?  
gl-sample-coverage : procedure?  
gl-scale : procedure?  
gl-scissor : procedure?  
gl-secondary-color : procedure?  
gl-secondary-color-v : procedure?  
gl-select-buffer->gl-uint-vector : procedure?  
gl-shade-model : procedure?  
gl-sphere : procedure?  
gl-stencil-func : procedure?  
gl-stencil-mask : procedure?  
gl-stencil-op : procedure?  
gl-tex-coord : procedure?  
gl-tex-coord-v : procedure?  
gl-tex-gen : procedure?  
gl-tex-gen-v : procedure?  
gl-translate : procedure?  
gl-u-get-string : procedure?  
gl-un-project : procedure?  
gl-un-project4 : procedure?  
gl-vertex : procedure?  
gl-vertex-v : procedure?  
gl-viewport : procedure?  
gl-window-pos : procedure?  
gl-window-pos-v : procedure?
```

Racket-style variants of the OpenGL functions.

```
(gl-process-selection vec hits) → (listof gl-selection-record?)  
  vec : gl-uint-vector?  
  hits : exact-nonnegative-integer?
```

Parses the contents of `vec` from the format used by `glSelectBuffer`. The second argument should be the number of hits as returned by `glRenderMode`.

```
(gl-get-gl-version-number) → exact-nonnegative-integer?
```

Returns the run-time OpenGL version number as an integer: 10, 11, 12, 13, 14, 15, or 20.

```
(gl-get-glu-version-number) → exact-nonnegative-integer?
```

Returns the run-time GLU version number as an integer: 10, 11, 12, or 13.

4 OpenGL Vectors

```
(require sgl/gl-vectors)      package: sgl
```

The `sgl/gl-vectors` module supports OpenGL programming with `cvector`s. In this document and in the error messages, a “gl-vector” is just a `cvector`, while a “gl-*<type>*-vector” is a `cvector` with an appropriate type. Use the `sgl/gl-vectors` module vectors instead of a Racket `cvector` directly, because they are specialized to handle the OpenGL types correctly.

```
(gl-vector? v) → boolean?
  v : any/c
(gl-vector->vector vec) → vector?
  vec : cvector?
(gl-vector->list vec) → list?
  vec : cvector?
(gl-vector-length vec) → exact-nonnegative-integer?
  vec : cvector?
(gl-vector-ref vec pos) → any/v
  vec : cvector?
  pos : exact-nonnegative-integer?
(gl-vector-set! vec pos v) → void?
  vec : cvector?
  pos : exact-nonnegative-integer?
  v : any/v
```

Synonyms for `cvector?`, `cvector->vector`, `cvector-length`, etc.

```
(gl-byte-vector? v) → boolean?
  v : any/c
(make-gl-byte-vector pos) → gl-byte-vector?
  pos : exact-nonnegative-integer?
(gl-byte-vector v ...) → gl-byte-vector?
  v : byte?
(vector->gl-byte-vector v ...) → gl-byte-vector?
  v : (vectorof byte?)
(list->gl-byte-vector v ...) → gl-byte-vector?
  v : (listof byte?)
(gl-byte-vector+ vec ...+) → gl-byte-vector?
  vec : gl-byte-vector?
(gl-byte-vector- vec ...+) → gl-byte-vector?
  vec : gl-byte-vector?
(gl-byte-vector* x vec) → gl-byte-vector?
  x : real?
  vec : gl-byte-vector?
```

Operations on vectors of `byte` elements. The `gl-byte-vector+` and `gl-byte-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-byte-vector*` function multiplies each element of `vec` by `x`.

```
(gl-byte-vector? v) → boolean?
  v : any/c
(make-gl-byte-vector pos) → gl-byte-vector?
  pos : exact-nonnegative-integer?
(gl-byte-vector v ...) → gl-byte-vector?
  v : ubyte?
(vector->gl-byte-vector v ...) → gl-byte-vector?
  v : (vectorof ubyte?)
(list->gl-byte-vector v ...) → gl-byte-vector?
  v : (listof ubyte?)
(gl-byte-vector+ vec ...+) → gl-byte-vector?
  vec : gl-byte-vector?
(gl-byte-vector- vec ...+) → gl-byte-vector?
  vec : gl-byte-vector?
(gl-byte-vector* x vec) → gl-byte-vector?
  x : real?
  vec : gl-byte-vector?
```

Operations on vectors of `ubyte` elements. The `gl-ubyte-vector+` and `gl-ubyte-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-ubyte-vector*` function multiplies each element of `vec` by `x`.

```
(gl-short-vector? v) → boolean?
  v : any/c
(make-gl-short-vector pos) → gl-short-vector?
  pos : exact-nonnegative-integer?
(gl-short-vector v ...) → gl-short-vector?
  v : short?
(vector->gl-short-vector v ...) → gl-short-vector?
  v : (vectorof short?)
(list->gl-short-vector v ...) → gl-short-vector?
  v : (listof short?)
(gl-short-vector+ vec ...+) → gl-short-vector?
  vec : gl-short-vector?
(gl-short-vector- vec ...+) → gl-short-vector?
  vec : gl-short-vector?
(gl-short-vector* x vec) → gl-short-vector?
  x : real?
  vec : gl-short-vector?
```

Operations on vectors of `short` elements. The `gl-short-vector+` and `gl-short-`

`vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-short-vector*` function multiplies each element of `vec` by `x`.

```
(gl-ushort-vector? v) → boolean?
  v : any/c
(make-gl-ushort-vector pos) → gl-ushort-vector?
  pos : exact-nonnegative-integer?
(gl-ushort-vector v ...) → gl-ushort-vector?
  v : ushort?
(vector->gl-ushort-vector v ...) → gl-ushort-vector?
  v : (vectorof ushort?)
(list->gl-ushort-vector v ...) → gl-ushort-vector?
  v : (listof ushort?)
(gl-ushort-vector+ vec ...+) → gl-ushort-vector?
  vec : gl-ushort-vector?
(gl-ushort-vector- vec ...+) → gl-ushort-vector?
  vec : gl-ushort-vector?
(gl-ushort-vector* x vec) → gl-ushort-vector?
  x : real?
  vec : gl-ushort-vector?
```

Operations on vectors of `ushort` elements. The `gl-ushort-vector+` and `gl-ushort-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-ushort-vector*` function multiplies each element of `vec` by `x`.

```
(gl-int-vector? v) → boolean?
  v : any/c
(make-gl-int-vector pos) → gl-int-vector?
  pos : exact-nonnegative-integer?
(gl-int-vector v ...) → gl-int-vector?
  v : int?
(vector->gl-int-vector v ...) → gl-int-vector?
  v : (vectorof int?)
(list->gl-int-vector v ...) → gl-int-vector?
  v : (listof int?)
(gl-int-vector+ vec ...+) → gl-int-vector?
  vec : gl-int-vector?
(gl-int-vector- vec ...+) → gl-int-vector?
  vec : gl-int-vector?
(gl-int-vector* x vec) → gl-int-vector?
  x : real?
  vec : gl-int-vector?
```

Operations on vectors of `int` elements. The `gl-int-vector+` and `gl-int-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively.

The `gl-int-vector*` function multiplies each element of `vec` by `x`.

```
(gl-uint-vector? v) → boolean?
  v : any/c
(make-gl-uint-vector pos) → gl-uint-vector?
  pos : exact-nonnegative-integer?
(gl-uint-vector v ...) → gl-uint-vector?
  v : uint?
(vector->gl-uint-vector v ...) → gl-uint-vector?
  v : (vectorof uint?)
(list->gl-uint-vector v ...) → gl-uint-vector?
  v : (listof uint?)
(gl-uint-vector+ vec ...+) → gl-uint-vector?
  vec : gl-uint-vector?
(gl-uint-vector- vec ...+) → gl-uint-vector?
  vec : gl-uint-vector?
(gl-uint-vector* x vec) → gl-uint-vector?
  x : real?
  vec : gl-uint-vector?
```

Operations on vectors of `uint` elements. The `gl-uint-vector+` and `gl-uint-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-uint-vector*` function multiplies each element of `vec` by `x`.

```
(gl-float-vector? v) → boolean?
  v : any/c
(make-gl-float-vector pos) → gl-float-vector?
  pos : exact-nonnegative-integer?
(gl-float-vector v ...) → gl-float-vector?
  v : float?
(vector->gl-float-vector v ...) → gl-float-vector?
  v : (vectorof float?)
(list->gl-float-vector v ...) → gl-float-vector?
  v : (listof float?)
(gl-float-vector+ vec ...+) → gl-float-vector?
  vec : gl-float-vector?
(gl-float-vector- vec ...+) → gl-float-vector?
  vec : gl-float-vector?
(gl-float-vector* x vec) → gl-float-vector?
  x : real?
  vec : gl-float-vector?
```

Operations on vectors of `float` elements. The `gl-float-vector+` and `gl-float-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-float-vector*` function multiplies each element of `vec` by `x`.

```

(gl-double-vector? v) → boolean?
  v : any/c
(make-gl-double-vector pos) → gl-double-vector?
  pos : exact-nonnegative-integer?
(gl-double-vector v ...) → gl-double-vector?
  v : double?
(vector->gl-double-vector v ...) → gl-double-vector?
  v : (vectorof double?)
(list->gl-double-vector v ...) → gl-double-vector?
  v : (listof double?)
(gl-double-vector+ vec ...+) → gl-double-vector?
  vec : gl-double-vector?
(gl-double-vector- vec ...+) → gl-double-vector?
  vec : gl-double-vector?
(gl-double-vector* x vec) → gl-double-vector?
  x : real?
  vec : gl-double-vector?

```

Operations on vectors of double elements. The `gl-double-vector+` and `gl-double-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-double-vector*` function multiplies each element of `vec` by `x`.

```

(gl-boolean-vector? v) → boolean?
  v : any/c
(make-gl-boolean-vector pos) → gl-boolean-vector?
  pos : exact-nonnegative-integer?
(gl-boolean-vector v ...) → gl-boolean-vector?
  v : boolean?
(vector->gl-boolean-vector v ...) → gl-boolean-vector?
  v : (vectorof boolean?)
(list->gl-boolean-vector v ...) → gl-boolean-vector?
  v : (listof boolean?)
(gl-boolean-vector+ vec ...+) → gl-boolean-vector?
  vec : gl-boolean-vector?
(gl-boolean-vector- vec ...+) → gl-boolean-vector?
  vec : gl-boolean-vector?
(gl-boolean-vector* x vec) → gl-boolean-vector?
  x : real?
  vec : gl-boolean-vector?

```

Operations on vectors of boolean elements. The `gl-boolean-vector+` and `gl-boolean-vector-` functions compute the element-by-element sum and difference of the given vectors, respectively. The `gl-boolean-vector*` function multiplies each element of `vec` by `x`.

```
(gl-vector-norm vec) → real?  
vec : gl-vector?
```

Returns the square root of the sum of the squares of the elements of *vec*.

5 Bitmaps

```
(require sgl/bitmap)      package: sgl

(bitmap->gl-list bitmap
  [#:with-gl with-gl-proc
   #:mask mask])          → exact-integer?

bitmap : (is-a?/c bitmap%)
with-gl-proc : ((-> any) . -> . any) = (lambda (f) (f))
mask : (or/c (is-a?/c bitmap%) false/c)
      = (send bitmap get-loaded-mask)
```

Converts the given bitmap into an OpenGL list that can be rendered with `gl-call-list` or `glCallList`. The rendered object is a square on the $z=0$ plane with corners at (0,0) and (1,1).

The `with-gl-proc` must accept a thunk and call it while the relevant OpenGL context is selected. Otherwise, the relevant OpenGL context must be selected already.

If `mask` is not `#f`, it is used as the mask bitmap for extracting alpha values.

6 Initialization

```
(require sgl/init)      package: sgl
```

Requiring the `sgl/init` library initializes platform-specific OpenGL state to help avoid crashes when OpenGL commands are incorrectly used without a current context. This library is required by `sgl` and `sgl/gl`, so it normally does not need to be required explicitly.

On Mac OS X, `sgl/init` checks whether any GL context is current, and if not, it creates a dummy context and sets it as the current context.

Index

bitmap->gl-list, 51
Bitmaps, 51
C-Style OpenGL, 4
feedback-buffer->gl-float-vector, 39
GFclampf, 4
GFfloat, 4
gl-accum, 40
gl-active-texture, 40
gl-alpha-func, 40
gl-begin, 40
gl-begin-query, 40
gl-blend-color, 41
gl-blend-equation, 41
gl-blend-func, 41
gl-blend-func-separate, 41
gl-boolean-vector, 49
gl-boolean-vector*, 49
gl-boolean-vector+, 49
gl-boolean-vector-, 49
gl-boolean-vector?, 49
gl-byte-vector, 45
gl-byte-vector*, 45
gl-byte-vector+, 45
gl-byte-vector-, 45
gl-byte-vector?, 45
gl-call-list, 41
gl-check-extension, 41
gl-clear, 41
gl-clear-accum, 41
gl-clear-color, 41
gl-clear-depth, 41
gl-clear-index, 41
gl-clear-stencil, 41
gl-clip-plane, 41
gl-color, 41
gl-color-mask, 41
gl-color-material, 41
gl-color-v, 41
gl-copy-pixels, 41
gl-cull-face, 41
gl-cylinder, 41
gl-delete-lists, 41
gl-delete-queries, 41
gl-depth-func, 41
gl-depth-mask, 41
gl-depth-range, 41
gl-disable, 41
gl-disk, 41
gl-double-vector, 49
gl-double-vector*, 49
gl-double-vector+, 49
gl-double-vector-, 49
gl-double-vector?, 49
gl-edge-flag, 41
gl-enable, 41
gl-end, 41
gl-end-list, 41
gl-end-query, 41
gl-eval-coord, 41
gl-eval-coord-v, 41
gl-eval-mesh, 41
gl-eval-point, 41
gl-feedback-buffer->gl-float-vector, 41
gl-finish, 41
gl-float-vector, 48
gl-float-vector*, 48
gl-float-vector+, 48
gl-float-vector-, 48
gl-float-vector?, 48
gl-flush, 41
gl-front-face, 41
gl-frustum, 41
gl-gen-lists, 41
gl-gen-queries, 41
gl-get-error, 41
gl-get-gl-version-number, 44
gl-get-glu-version-number, 44
gl-get-string, 42
gl-hint, 42
gl-index, 42

- `gl-index-mask`, 42
- `gl-index-v`, 42
- `gl-init-names`, 42
- `gl-int-vector`, 47
- `gl-int-vector*`, 47
- `gl-int-vector+`, 47
- `gl-int-vector-`, 47
- `gl-int-vector?`, 47
- `gl-is-buffer`, 42
- `gl-is-enabled`, 42
- `gl-is-list`, 42
- `gl-is-query`, 42
- `gl-light`, 42
- `gl-light-model`, 42
- `gl-light-model-v`, 42
- `gl-light-v`, 42
- `gl-line-stipple`, 42
- `gl-line-width`, 42
- `gl-list-base`, 42
- `gl-load-identity`, 42
- `gl-load-matrix`, 42
- `gl-load-name`, 42
- `gl-load-transpose-matrix`, 42
- `gl-look-at`, 42
- `gl-map-grid`, 42
- `gl-material`, 42
- `gl-material-v`, 42
- `gl-matrix-mode`, 42
- `gl-mult-matrix`, 42
- `gl-mult-transpose-matrix`, 42
- `gl-multi-tex-coord`, 42
- `gl-multi-tex-coord-v`, 42
- `gl-new-list`, 42
- `gl-new-quadric`, 42
- `gl-normal`, 42
- `gl-normal-v`, 42
- `gl-ortho`, 42
- `gl-ortho-2d`, 42
- `gl-partial-disk`, 42
- `gl-pass-through`, 42
- `gl-perspective`, 42
- `gl-pick-matrix`, 42
- `gl-pixel-store`, 42
- `gl-point-parameter`, 42
- `gl-point-parameter-v`, 42
- `gl-point-size`, 42
- `gl-polygon-mode`, 43
- `gl-polygon-offset`, 43
- `gl-pop-attrib`, 43
- `gl-pop-client-attrib`, 43
- `gl-pop-matrix`, 43
- `gl-pop-name`, 43
- `gl-process-selection`, 44
- `gl-project`, 43
- `gl-push-matrix`, 43
- `gl-push-name`, 43
- `gl-quadric-draw-style`, 43
- `gl-quadric-normals`, 43
- `gl-quadric-orientation`, 43
- `gl-quadric-texture`, 43
- `gl-raster-pos`, 43
- `gl-raster-pos-v`, 43
- `gl-rect`, 43
- `gl-rect-v`, 43
- `gl-render-mode`, 43
- `gl-rotate`, 43
- `gl-sample-coverage`, 43
- `gl-scale`, 43
- `gl-scissor`, 43
- `gl-secondary-color`, 43
- `gl-secondary-color-v`, 43
- `gl-select-buffer->gl-uint-vector`, 43
- `gl-selection-record` (struct), 40
- `gl-selection-record-max-z`, 40
- `gl-selection-record-min-z`, 40
- `gl-selection-record-stack`, 40
- `gl-selection-record?`, 40
- `gl-shade-model`, 43
- `gl-short-vector`, 46
- `gl-short-vector*`, 46
- `gl-short-vector+`, 46
- `gl-short-vector-`, 46
- `gl-short-vector?`, 46

- gl-sphere, 43
- gl-stencil-func, 43
- gl-stencil-mask, 43
- gl-stencil-op, 43
- gl-tex-coord, 43
- gl-tex-coord-v, 43
- gl-tex-gen, 43
- gl-tex-gen-v, 43
- gl-translate, 43
- gl-u-get-string, 43
- gl-ubyte-vector, 46
- gl-ubyte-vector*, 46
- gl-ubyte-vector+, 46
- gl-ubyte-vector-, 46
- gl-ubyte-vector?, 46
- gl-uint-vector, 48
- gl-uint-vector*, 48
- gl-uint-vector+, 48
- gl-uint-vector-, 48
- gl-uint-vector?, 48
- gl-un-project, 43
- gl-un-project4, 43
- gl-ushort-vector, 47
- gl-ushort-vector*, 47
- gl-ushort-vector+, 47
- gl-ushort-vector-, 47
- gl-ushort-vector?, 47
- gl-vector->list, 45
- gl-vector->vector, 45
- gl-vector-length, 45
- gl-vector-norm, 50
- gl-vector-ref, 45
- gl-vector-set!, 45
- gl-vector?, 45
- gl-vertex, 43
- gl-vertex-v, 43
- gl-viewport, 43
- gl-window-pos, 43
- gl-window-pos-v, 43
- GL: 3-D Graphics, 1
- GL_2_BYTES, 17
- GL_2D, 21
- GL_3_BYTES, 17
- GL_3D, 21
- GL_3D_COLOR, 21
- GL_3D_COLOR_TEXTURE, 21
- GL_4_BYTES, 17
- GL_4D_COLOR_TEXTURE, 21
- GL_ACCUM, 20
- GL_ACCUM_ALPHA_BITS, 20
- GL_ACCUM_BLUE_BITS, 20
- GL_ACCUM_BUFFER_BIT, 27
- GL_ACCUM_CLEAR_VALUE, 20
- GL_ACCUM_GREEN_BITS, 20
- GL_ACCUM_RED_BITS, 20
- GL_ACTIVE_TEXTURE, 32
- GL_ADD, 20
- GL_ADD_SIGNED, 33
- GL_ALIASED_LINE_WIDTH_RANGE, 29
- GL_ALIASED_POINT_SIZE_RANGE, 29
- GL_ALL_ATTRIB_BITS, 28
- GL_ALL_CLIENT_ATTRIB_BITS, 29
- GL_ALPHA, 23
- GL_ALPHA12, 28
- GL_ALPHA16, 28
- GL_ALPHA4, 28
- GL_ALPHA8, 28
- GL_ALPHA_BIAS, 25
- GL_ALPHA_BITS, 23
- GL_ALPHA_SCALE, 25
- GL_ALPHA_TEST, 20
- GL_ALPHA_TEST_FUNC, 20
- GL_ALPHA_TEST_REF, 20
- GL_ALWAYS, 19
- GL_AMBIENT, 20
- GL_AMBIENT_AND_DIFFUSE, 20
- GL_AND, 22
- GL_AND_INVERTED, 22
- GL_AND_REVERSE, 22
- GL_ARRAY_BUFFER, 34
- GL_ARRAY_BUFFER_BINDING, 34
- GL_ATTRIB_STACK_DEPTH, 23
- GL_AUTO_NORMAL, 24
- GL_AUX0, 22

GL_AUX1, 22	GL_CLIENT_PIXEL_STORE_BIT, 28
GL_AUX2, 22	GL_CLIENT_VERTEX_ARRAY_BIT, 28
GL_AUX3, 22	GL_CLIP_PLANE0, 20
GL_AUX_BUFFERS, 23	GL_CLIP_PLANE1, 20
GL_BACK, 18	GL_CLIP_PLANE2, 20
GL_BACK_LEFT, 22	GL_CLIP_PLANE3, 20
GL_BACK_RIGHT, 22	GL_CLIP_PLANE4, 20
GL_BGR, 29	GL_CLIP_PLANE5, 20
GL_BGRA, 29	GL_COEFF, 24
GL_BITMAP, 23	GL_COLOR, 23
GL_BITMAP_TOKEN, 21	GL_COLOR_ARRAY, 17
GL_BLEND, 20	GL_COLOR_ARRAY_BUFFER_BINDING, 35
GL_BLEND_COLOR, 30	GL_COLOR_ARRAY_POINTER, 18
GL_BLEND_DST, 20	GL_COLOR_ARRAY_SIZE, 17
GL_BLEND_DST_ALPHA, 33	GL_COLOR_ARRAY_STRIDE, 17
GL_BLEND_DST_RGB, 33	GL_COLOR_ARRAY_TYPE, 17
GL_BLEND_EQUATION, 30	GL_COLOR_BUFFER_BIT, 27
GL_BLEND_SRC, 20	GL_COLOR_CLEAR_VALUE, 23
GL_BLEND_SRC_ALPHA, 33	GL_COLOR_INDEX, 22
GL_BLEND_SRC_RGB, 33	GL_COLOR_INDEXES, 20
GL_BLUE, 23	GL_COLOR_LOGIC_OP, 21
GL_BLUE_BIAS, 25	GL_COLOR_MATERIAL, 20
GL_BLUE_BITS, 23	GL_COLOR_MATERIAL_FACE, 20
GL_BLUE_SCALE, 25	GL_COLOR_MATERIAL_PARAMETER, 20
GL_BUFFER_ACCESS, 35	GL_COLOR_MATRIX, 31
GL_BUFFER_MAP_POINTER, 35	GL_COLOR_MATRIX_STACK_DEPTH, 31
GL_BUFFER_MAPPED, 35	GL_COLOR_SUM, 34
GL_BUFFER_SIZE, 34	GL_COLOR_TABLE, 31
GL_BUFFER_USAGE, 34	GL_COLOR_TABLE_ALPHA_SIZE, 31
GL_BYTE, 17	GL_COLOR_TABLE_BIAS, 31
GL_C3F_V3F, 18	GL_COLOR_TABLE_BLUE_SIZE, 31
GL_C4F_N3F_V3F, 18	GL_COLOR_TABLE_FORMAT, 31
GL_C4UB_V2F, 18	GL_COLOR_TABLE_GREEN_SIZE, 31
GL_C4UB_V3F, 18	GL_COLOR_TABLE_INTENSITY_SIZE, 31
GL_CCW, 18	GL_COLOR_TABLE_LUMINANCE_SIZE, 31
GL_CLAMP, 27	GL_COLOR_TABLE_RED_SIZE, 31
GL_CLAMP_TO_BORDER, 33	GL_COLOR_TABLE_SCALE, 31
GL_CLAMP_TO_EDGE, 29	GL_COLOR_TABLE_WIDTH, 31
GL_CLEAR, 22	GL_COLOR_WRITEMASK, 23
GL_CLIENT_ACTIVE_TEXTURE, 32	GL_COMBINE, 33
GL_CLIENT_ALL_ATTRIB_BITS, 29	GL_COMBINE_ALPHA, 33
GL_CLIENT_ATTRIB_STACK_DEPTH, 23	GL_COMBINE_RGB, 33

GL_COMPARE_R_TO_TEXTURE, 34	GL_CURRENT_RASTER_TEXTURE_COORDS,
GL_COMPILE, 19	24
GL_COMPILE_AND_EXECUTE, 19	GL_CURRENT_SECONDARY_COLOR, 34
GL_COMPRESSED_ALPHA, 33	GL_CURRENT_TEXTURE_COORDS, 24
GL_COMPRESSED_INTENSITY, 33	GL_CW, 18
GL_COMPRESSED_LUMINANCE, 33	GL_DECAL, 27
GL_COMPRESSED_LUMINANCE_ALPHA, 33	GL_DECR, 22
GL_COMPRESSED_RGB, 33	GL_DECR_WRAP, 34
GL_COMPRESSED_RGBA, 33	GL_DEPTH, 23
GL_COMPRESSED_TEXTURE_FORMATS, 33	GL_DEPTH_BIAS, 25
GL_CONSTANT, 33	GL_DEPTH_BITS, 19
GL_CONSTANT_ALPHA, 30	GL_DEPTH_BUFFER_BIT, 27
GL_CONSTANT_ATTENUATION, 19	GL_DEPTH_CLEAR_VALUE, 19
GL_CONSTANT_BORDER, 31	GL_DEPTH_COMPONENT, 19
GL_CONSTANT_COLOR, 29	GL_DEPTH_COMPONENT16, 34
GL_CONVOLUTION_1D, 30	GL_DEPTH_COMPONENT24, 34
GL_CONVOLUTION_2D, 30	GL_DEPTH_COMPONENT32, 34
GL_CONVOLUTION_BORDER_COLOR, 31	GL_DEPTH_FUNC, 19
GL_CONVOLUTION_BORDER_MODE, 30	GL_DEPTH_RANGE, 19
GL_CONVOLUTION_FILTER_BIAS, 30	GL_DEPTH_SCALE, 25
GL_CONVOLUTION_FILTER_SCALE, 30	GL_DEPTH_TEST, 19
GL_CONVOLUTION_FORMAT, 30	GL_DEPTH_TEXTURE_MODE, 34
GL_CONVOLUTION_HEIGHT, 30	GL_DEPTH_WRITEMASK, 19
GL_CONVOLUTION_WIDTH, 30	GL_DIFFUSE, 20
GL_COPY, 22	GL_DITHER, 23
GL_COPY_INVERTED, 22	GL_DOMAIN, 24
GL_COPY_PIXEL_TOKEN, 21	GL_DONT_CARE, 25
GL_CULL_FACE, 19	GL_DOT3_RGB, 33
GL_CULL_FACE_MODE, 19	GL_DOT3_RGBA, 33
GL_CURRENT_BIT, 27	GL_DOUBLE, 17
GL_CURRENT_COLOR, 23	GL_DOUBLEBUFFER, 23
GL_CURRENT_FOG_COORD, 35	GL_DRAW_BUFFER, 23
GL_CURRENT_FOG_COORDINATE, 34	GL_DRAW_PIXEL_TOKEN, 21
GL_CURRENT_INDEX, 23	GL_DST_ALPHA, 21
GL_CURRENT_NORMAL, 23	GL_DST_COLOR, 21
GL_CURRENT_QUERY, 34	GL_DYNAMIC_COPY, 35
GL_CURRENT_RASTER_COLOR, 23	GL_DYNAMIC_DRAW, 35
GL_CURRENT_RASTER_DISTANCE, 24	GL_DYNAMIC_READ, 35
GL_CURRENT_RASTER_INDEX, 24	GL_EDGE_FLAG, 19
GL_CURRENT_RASTER_POSITION, 24	GL_EDGE_FLAG_ARRAY, 17
GL_CURRENT_RASTER_POSITION_VALID,	GL_EDGE_FLAG_ARRAY_BUFFER_BINDING,
24	35

GL_EDGE_FLAG_ARRAY_POINTER, 18	GL_FOG_COORDINATE_SOURCE, 34
GL_EDGE_FLAG_ARRAY_STRIDE, 17	GL_FOG_DENSITY, 21
GL_ELEMENT_ARRAY_BUFFER, 34	GL_FOG_END, 21
GL_ELEMENT_ARRAY_BUFFER_BINDING, 34	GL_FOG_HINT, 25
GL_EMISSION, 20	GL_FOG_INDEX, 21
GL_ENABLE_BIT, 27	GL_FOG_MODE, 21
GL_EQUAL, 19	GL_FOG_START, 21
GL_EQUIV, 22	GL_FRAGMENT_DEPTH, 34
GL_EVAL_BIT, 27	GL_FRONT, 18
GL_EXP, 21	GL_FRONT_AND_BACK, 20
GL_EXP2, 21	GL_FRONT_FACE, 19
GL_EXTENSIONS, 27	GL_FRONT_LEFT, 22
GL_EYE_LINEAR, 27	GL_FRONT_RIGHT, 22
GL_EYE_PLANE, 27	GL_FUNC_ADD, 30
GL_FALSE, 17	GL_FUNC_REVERSE_SUBTRACT, 30
GL_FALSE, 4	GL_FUNC_SUBTRACT, 30
GL_FASTEST, 25	GL_GENERATE_MIPMAP, 34
GL_FEEDBACK, 21	GL_GENERATE_MIPMAP_HINT, 34
GL_FEEDBACK_BUFFER_POINTER, 21	GL_GEQUAL, 19
GL_FEEDBACK_BUFFER_SIZE, 21	GL_GREATER, 19
GL_FEEDBACK_BUFFER_TYPE, 21	GL_GREEN, 23
GL_FILL, 18	GL_GREEN_BIAS, 25
GL_FLAT, 20	GL_GREEN_BITS, 23
GL_FLOAT, 17	GL_GREEN_SCALE, 25
GL_FOG, 21	GL_HINT_BIT, 27
GL_FOG_BIT, 27	GL_HISTOGRAM, 30
GL_FOG_COLOR, 21	GL_HISTOGRAM_ALPHA_SIZE, 30
GL_FOG_COORD, 35	GL_HISTOGRAM_BLUE_SIZE, 30
GL_FOG_COORD_ARRAY, 35	GL_HISTOGRAM_FORMAT, 30
GL_FOG_COORD_ARRAY_BUFFER_BINDING, 35	GL_HISTOGRAM_GREEN_SIZE, 30
GL_FOG_COORD_ARRAY_POINTER, 35	GL_HISTOGRAM_LUMINANCE_SIZE, 30
GL_FOG_COORD_ARRAY_STRIDE, 35	GL_HISTOGRAM_RED_SIZE, 30
GL_FOG_COORD_ARRAY_TYPE, 35	GL_HISTOGRAM_SINK, 30
GL_FOG_COORD_SRC, 35	GL_HISTOGRAM_WIDTH, 30
GL_FOG_COORDINATE, 34	GL_INCR, 22
GL_FOG_COORDINATE_ARRAY, 34	GL_INCR_WRAP, 34
GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING, 35	GL_INDEX_ARRAY, 17
GL_FOG_COORDINATE_ARRAY_POINTER, 34	GL_INDEX_ARRAY_BUFFER_BINDING, 35
GL_FOG_COORDINATE_ARRAY_STRIDE, 34	GL_INDEX_ARRAY_POINTER, 18
GL_FOG_COORDINATE_ARRAY_TYPE, 34	GL_INDEX_ARRAY_STRIDE, 17
	GL_INDEX_ARRAY_TYPE, 17
	GL_INDEX_BITS, 23

GL_INDEX_CLEAR_VALUE, 24	GL_LINE_STIPPLE_PATTERN, 18
GL_INDEX_LOGIC_OP, 21	GL_LINE_STIPPLE_REPEAT, 18
GL_INDEX_MODE, 24	GL_LINE_STRIP, 17
GL_INDEX_OFFSET, 25	GL_LINE_TOKEN, 21
GL_INDEX_SHIFT, 25	GL_LINE_WIDTH, 18
GL_INDEX_WRITEMASK, 24	GL_LINE_WIDTH_GRANULARITY, 18
GL_INT, 17	GL_LINE_WIDTH_RANGE, 18
GL_INTENSITY, 28	GL_LINEAR, 21
GL_INTENSITY12, 28	GL_LINEAR_ATTENUATION, 19
GL_INTENSITY16, 28	GL_LINEAR_MIPMAP_LINEAR, 26
GL_INTENSITY4, 28	GL_LINEAR_MIPMAP_NEAREST, 26
GL_INTENSITY8, 28	GL_LINES, 17
GL_INTERPOLATE, 33	GL_LIST_BASE, 19
GL_INVALID_ENUM, 27	GL_LIST_BIT, 27
GL_INVALID_OPERATION, 27	GL_LIST_INDEX, 19
GL_INVALID_VALUE, 27	GL_LIST_MODE, 19
GL_INVERT, 22	GL_LOAD, 20
GL_KEEP, 22	GL_LOGIC_OP, 21
GL_LEFT, 22	GL_LOGIC_OP_MODE, 21
GL_LEQUAL, 19	GL_LUMINANCE, 23
GL_LESS, 19	GL_LUMINANCE12, 28
GL_LIGHT0, 19	GL_LUMINANCE12_ALPHA12, 28
GL_LIGHT1, 19	GL_LUMINANCE12_ALPHA4, 28
GL_LIGHT2, 19	GL_LUMINANCE16, 28
GL_LIGHT3, 19	GL_LUMINANCE16_ALPHA16, 28
GL_LIGHT4, 19	GL_LUMINANCE4, 28
GL_LIGHT5, 19	GL_LUMINANCE4_ALPHA4, 28
GL_LIGHT6, 19	GL_LUMINANCE6_ALPHA2, 28
GL_LIGHT7, 19	GL_LUMINANCE8, 28
GL_LIGHT_MODEL_AMBIENT, 20	GL_LUMINANCE8_ALPHA8, 28
GL_LIGHT_MODEL_COLOR_CONTROL, 29	GL_LUMINANCE_ALPHA, 23
GL_LIGHT_MODEL_LOCAL_VIEWER, 20	GL_MAP1_COLOR_4, 24
GL_LIGHT_MODEL_TWO_SIDE, 20	GL_MAP1_GRID_DOMAIN, 24
GL_LIGHTING, 19	GL_MAP1_GRID_SEGMENTS, 24
GL_LIGHTING_BIT, 27	GL_MAP1_INDEX, 24
GL_LINE, 18	GL_MAP1_NORMAL, 24
GL_LINE_BIT, 27	GL_MAP1_TEXTURE_COORD_1, 24
GL_LINE_LOOP, 17	GL_MAP1_TEXTURE_COORD_2, 24
GL_LINE_RESET_TOKEN, 21	GL_MAP1_TEXTURE_COORD_3, 24
GL_LINE_SMOOTH, 18	GL_MAP1_TEXTURE_COORD_4, 24
GL_LINE_SMOOTH_HINT, 25	GL_MAP1_VERTEX_3, 24
GL_LINE_STIPPLE, 18	GL_MAP1_VERTEX_4, 24

[GL_MAP2_COLOR_4](#), 24
[GL_MAP2_GRID_DOMAIN](#), 24
[GL_MAP2_GRID_SEGMENTS](#), 24
[GL_MAP2_INDEX](#), 24
[GL_MAP2_NORMAL](#), 24
[GL_MAP2_TEXTURE_COORD_1](#), 24
[GL_MAP2_TEXTURE_COORD_2](#), 24
[GL_MAP2_TEXTURE_COORD_3](#), 24
[GL_MAP2_TEXTURE_COORD_4](#), 24
[GL_MAP2_VERTEX_3](#), 24
[GL_MAP2_VERTEX_4](#), 24
[GL_MAP_COLOR](#), 25
[GL_MAP_STENCIL](#), 25
[GL_MATRIX_MODE](#), 18
[GL_MAX](#), 30
[GL_MAX_3D_TEXTURE_SIZE](#), 29
[GL_MAX_ATTRIB_STACK_DEPTH](#), 23
[GL_MAX_CLIENT_ATTRIB_STACK_DEPTH](#), 23
[GL_MAX_CLIP_PLANES](#), 23
[GL_MAX_COLOR_MATRIX_STACK_DEPTH](#), 31
[GL_MAX_CONVOLUTION_HEIGHT](#), 30
[GL_MAX_CONVOLUTION_WIDTH](#), 30
[GL_MAX_CUBE_MAP_TEXTURE_SIZE](#), 33
[GL_MAX_ELEMENTS_INDICES](#), 29
[GL_MAX_ELEMENTS_VERTICES](#), 29
[GL_MAX_EVAL_ORDER](#), 23
[GL_MAX_LIGHTS](#), 23
[GL_MAX_LIST_NESTING](#), 23
[GL_MAX_MODELVIEW_STACK_DEPTH](#), 23
[GL_MAX_NAME_STACK_DEPTH](#), 23
[GL_MAX_PIXEL_MAP_TABLE](#), 23
[GL_MAX_PROJECTION_STACK_DEPTH](#), 23
[GL_MAX_TEXTURE_LOD_BIAS](#), 34
[GL_MAX_TEXTURE_SIZE](#), 23
[GL_MAX_TEXTURE_STACK_DEPTH](#), 23
[GL_MAX_TEXTURE_UNITS](#), 32
[GL_MAX_VIEWPORT_DIMS](#), 23
[GL_MIN](#), 30
[GL_MINMAX](#), 30
[GL_MINMAX_FORMAT](#), 30
[GL_MINMAX_SINK](#), 30
[GL_MIRRORED_REPEAT](#), 34
[GL_MODELVIEW](#), 18
[GL_MODELVIEW_MATRIX](#), 24
[GL_MODELVIEW_STACK_DEPTH](#), 24
[GL_MODULATE](#), 27
[GL_MULT](#), 20
[GL_MULTISAMPLE](#), 32
[GL_MULTISAMPLE_BIT](#), 32
[GL_N3F_V3F](#), 18
[GL_NAME_STACK_DEPTH](#), 24
[GL_NAND](#), 22
[GL_NEAREST](#), 27
[GL_NEAREST_MIPMAP_LINEAR](#), 26
[GL_NEAREST_MIPMAP_NEAREST](#), 26
[GL_NEVER](#), 19
[GL_NICEST](#), 25
[GL_NO_ERROR](#), 27
[GL_NONE](#), 22
[GL_NOOP](#), 22
[GL_NOR](#), 22
[GL_NORMAL_ARRAY](#), 17
[GL_NORMAL_ARRAY_BUFFER_BINDING](#), 35
[GL_NORMAL_ARRAY_POINTER](#), 18
[GL_NORMAL_ARRAY_STRIDE](#), 17
[GL_NORMAL_ARRAY_TYPE](#), 17
[GL_NORMAL_MAP](#), 32
[GL_NORMALIZE](#), 20
[GL_NOTEQUAL](#), 19
[GL_NUM_COMPRESSED_TEXTURE_FORMATS](#), 33
[GL_OBJECT_LINEAR](#), 26
[GL_OBJECT_PLANE](#), 26
[GL_ONE](#), 20
[GL_ONE_MINUS_CONSTANT_ALPHA](#), 30
[GL_ONE_MINUS_CONSTANT_COLOR](#), 30
[GL_ONE_MINUS_DST_ALPHA](#), 21
[GL_ONE_MINUS_DST_COLOR](#), 21
[GL_ONE_MINUS_SRC_ALPHA](#), 21
[GL_ONE_MINUS_SRC_COLOR](#), 21
[GL_OPERANDO_ALPHA](#), 33
[GL_OPERANDO_RGB](#), 33
[GL_OPERAND1_ALPHA](#), 33

GL_OPERAND1_RGB, 33	GL_POINT_FADE_THRESHOLD_SIZE, 34
GL_OPERAND2_ALPHA, 33	GL_POINT_SIZE, 18
GL_OPERAND2_RGB, 33	GL_POINT_SIZE_GRANULARITY, 18
GL_OR, 22	GL_POINT_SIZE_MAX, 33
GL_OR_INVERTED, 22	GL_POINT_SIZE_MIN, 33
GL_OR_REVERSE, 22	GL_POINT_SIZE_RANGE, 18
GL_ORDER, 25	GL_POINT_SMOOTH, 18
GL_OUT_OF_MEMORY, 27	GL_POINT_SMOOTH_HINT, 25
GL_PACK_ALIGNMENT, 26	GL_POINT_TOKEN, 21
GL_PACK_IMAGE_HEIGHT, 29	GL_POINTS, 17
GL_PACK_LSB_FIRST, 26	GL_POLYGON, 17
GL_PACK_ROW_LENGTH, 26	GL_POLYGON_BIT, 27
GL_PACK_SKIP_IMAGES, 29	GL_POLYGON_MODE, 18
GL_PACK_SKIP_PIXELS, 26	GL_POLYGON_OFFSET_FACTOR, 19
GL_PACK_SKIP_ROWS, 26	GL_POLYGON_OFFSET_FILL, 19
GL_PACK_SWAP_BYTES, 26	GL_POLYGON_OFFSET_LINE, 19
GL_PASS_THROUGH_TOKEN, 21	GL_POLYGON_OFFSET_POINT, 19
GL_PERSPECTIVE_CORRECTION_HINT, 25	GL_POLYGON_OFFSET_UNITS, 19
GL_PIXEL_MAP_A_TO_A, 26	GL_POLYGON_SMOOTH, 18
GL_PIXEL_MAP_A_TO_A_SIZE, 25	GL_POLYGON_SMOOTH_HINT, 25
GL_PIXEL_MAP_B_TO_B, 25	GL_POLYGON_STIPPLE, 19
GL_PIXEL_MAP_B_TO_B_SIZE, 25	GL_POLYGON_STIPPLE_BIT, 27
GL_PIXEL_MAP_G_TO_G, 25	GL_POLYGON_TOKEN, 21
GL_PIXEL_MAP_G_TO_G_SIZE, 25	GL_POSITION, 20
GL_PIXEL_MAP_I_TO_A, 25	GL_POST_COLOR_MATRIX_ALPHA_BIAS, 31
GL_PIXEL_MAP_I_TO_A_SIZE, 25	GL_POST_COLOR_MATRIX_ALPHA_SCALE, 31
GL_PIXEL_MAP_I_TO_B, 25	GL_POST_COLOR_MATRIX_BLUE_BIAS, 31
GL_PIXEL_MAP_I_TO_B_SIZE, 25	GL_POST_COLOR_MATRIX_BLUE_SCALE, 31
GL_PIXEL_MAP_I_TO_G, 25	GL_POST_COLOR_MATRIX_COLOR_TABLE, 31
GL_PIXEL_MAP_I_TO_G_SIZE, 25	GL_POST_COLOR_MATRIX_GREEN_BIAS, 31
GL_PIXEL_MAP_I_TO_I, 25	GL_POST_COLOR_MATRIX_GREEN_SCALE, 31
GL_PIXEL_MAP_I_TO_I_SIZE, 25	GL_POST_COLOR_MATRIX_RED_BIAS, 31
GL_PIXEL_MAP_I_TO_R, 25	GL_POST_COLOR_MATRIX_RED_SCALE, 31
GL_PIXEL_MAP_I_TO_R_SIZE, 25	GL_POST_CONVOLUTION_ALPHA_BIAS, 30
GL_PIXEL_MAP_R_TO_R, 25	GL_POST_CONVOLUTION_ALPHA_SCALE, 30
GL_PIXEL_MAP_R_TO_R_SIZE, 25	GL_POST_CONVOLUTION_BLUE_BIAS, 30
GL_PIXEL_MAP_S_TO_S, 25	GL_POST_CONVOLUTION_BLUE_SCALE, 30
GL_PIXEL_MAP_S_TO_S_SIZE, 25	GL_POST_CONVOLUTION_COLOR_TABLE, 31
GL_PIXEL_MODE_BIT, 27	GL_POST_CONVOLUTION_GREEN_BIAS, 30
GL_POINT, 18	
GL_POINT_BIT, 27	
GL_POINT_DISTANCE_ATTENUATION, 34	

GL_POST_CONVOLUTION_GREEN_SCALE, 30	GL_RESCALE_NORMAL, 29
GL_POST_CONVOLUTION_RED_BIAS, 30	GL_RETURN, 20
GL_POST_CONVOLUTION_RED_SCALE, 30	GL_RGB, 23
GL_PREVIOUS, 33	GL_RGB10, 28
GL_PRIMARY_COLOR, 33	GL_RGB10_A2, 28
GL_PROJECTION, 18	GL_RGB12, 28
GL_PROJECTION_MATRIX, 24	GL_RGB16, 28
GL_PROJECTION_STACK_DEPTH, 24	GL_RGB4, 28
GL_PROXY_COLOR_TABLE, 31	GL_RGB5, 28
GL_PROXY_HISTOGRAM, 30	GL_RGB5_A1, 28
GL_PROXY_POST_COLOR_MATRIX_COLOR_TABLE, 31	GL_RGB8, 28
GL_PROXY_POST_CONVOLUTION_COLOR_TABLE, 31	GL_RGB_SCALE, 33
GL_PROXY_TEXTURE_1D, 28	GL_RGBA, 23
GL_PROXY_TEXTURE_2D, 28	GL_RGBA12, 28
GL_PROXY_TEXTURE_3D, 29	GL_RGBA16, 28
GL_PROXY_TEXTURE_CUBE_MAP, 33	GL_RGBA2, 28
GL_Q, 27	GL_RGBA4, 28
GL_QUAD_STRIP, 17	GL_RGBA8, 28
GL_QUADRATIC_ATTENUATION, 19	GL_RGBA_MODE, 24
GL_QUADS, 17	GL_RIGHT, 22
GL_QUERY_COUNTER_BITS, 34	GL_S, 27
GL_QUERY_RESULT, 34	GL_SAMPLE_ALPHA_TO_COVERAGE, 32
GL_QUERY_RESULT_AVAILABLE, 34	GL_SAMPLE_ALPHA_TO_ONE, 32
GL_R, 27	GL_SAMPLE_BUFFERS, 32
GL_R3_G3_B2, 28	GL_SAMPLE_COVERAGE, 32
GL_READ_BUFFER, 23	GL_SAMPLE_COVERAGE_INVERT, 32
GL_READ_ONLY, 35	GL_SAMPLE_COVERAGE_VALUE, 32
GL_READ_WRITE, 35	GL_SAMPLES, 32
GL_RED, 22	GL_SAMPLES_PASSED, 35
GL_RED_BIAS, 25	GL_SCISSOR_BIT, 28
GL_RED_BITS, 23	GL_SCISSOR_BOX, 25
GL_RED_SCALE, 25	GL_SCISSOR_TEST, 25
GL_REDUCE, 30	GL_SECONDARY_COLOR_ARRAY, 34
GL_REFLECTION_MAP, 32	GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING, 35
GL_RENDER, 21	GL_SECONDARY_COLOR_ARRAY_POINTER, 34
GL_RENDER_MODE, 24	GL_SECONDARY_COLOR_ARRAY_SIZE, 34
GL_RENDERER, 27	GL_SECONDARY_COLOR_ARRAY_STRIDE, 34
GL_REPEAT, 27	GL_SECONDARY_COLOR_ARRAY_TYPE, 34
GL_REPLACE, 22	GL_SELECT, 21
GL_REPLICATE_BORDER, 31	GL_SELECTION_BUFFER_POINTER, 21

GL_SELECTION_BUFFER_SIZE, 21	GL_STENCIL_BUFFER_BIT, 27
GL_SEPARABLE_2D, 30	GL_STENCIL_CLEAR_VALUE, 22
GL_SEPARATE_SPECULAR_COLOR, 29	GL_STENCIL_FAIL, 22
GL_SET, 22	GL_STENCIL_FUNC, 22
GL_SHADE_MODEL, 20	GL_STENCIL_INDEX, 22
GL_SHININESS, 20	GL_STENCIL_PASS_DEPTH_FAIL, 22
GL_SHORT, 17	GL_STENCIL_PASS_DEPTH_PASS, 22
GL_SINGLE_COLOR, 29	GL_STENCIL_REF, 22
GL_SMOOTH, 20	GL_STENCIL_TEST, 22
GL_SMOOTH_LINE_WIDTH_GRANULARITY, 29	GL_STENCIL_VALUE_MASK, 22
GL_SMOOTH_LINE_WIDTH_RANGE, 29	GL_STENCIL_WRITEMASK, 22
GL_SMOOTH_POINT_SIZE_GRANULARITY, 29	GL_STEREO, 23
GL_SMOOTH_POINT_SIZE_RANGE, 29	GL_STREAM_COPY, 35
GL_SOURCE0_ALPHA, 33	GL_STREAM_DRAW, 35
GL_SOURCE0_RGB, 33	GL_STREAM_READ, 35
GL_SOURCE1_ALPHA, 33	GL_SUBPIXEL_BITS, 23
GL_SOURCE1_RGB, 33	GL_SUBTRACT, 33
GL_SOURCE2_ALPHA, 33	GL_T, 27
GL_SOURCE2_RGB, 33	GL_T2F_C3F_V3F, 18
GL_SPECULAR, 20	GL_T2F_C4F_N3F_V3F, 18
GL_SPHERE_MAP, 27	GL_T2F_C4UB_V3F, 18
GL_SPOT_CUTOFF, 19	GL_T2F_N3F_V3F, 18
GL_SPOT_DIRECTION, 20	GL_T2F_V3F, 18
GL_SPOT_EXPONENT, 19	GL_T4F_C4F_N3F_V4F, 18
GL_SRC0_ALPHA, 35	GL_T4F_V4F, 18
GL_SRC0_RGB, 35	GL_TABLE_TOO_LARGE, 30
GL_SRC1_ALPHA, 35	GL_TEXTURE, 18
GL_SRC1_RGB, 35	GL_TEXTURE0, 31
GL_SRC2_ALPHA, 35	GL_TEXTURE1, 31
GL_SRC2_RGB, 35	GL_TEXTURE10, 31
GL_SRC_ALPHA, 21	GL_TEXTURE11, 31
GL_SRC_ALPHA_SATURATE, 21	GL_TEXTURE12, 31
GL_SRC_COLOR, 21	GL_TEXTURE13, 31
GL_STACK_OVERFLOW, 27	GL_TEXTURE14, 32
GL_STACK_UNDERFLOW, 27	GL_TEXTURE15, 32
GL_STATIC_COPY, 35	GL_TEXTURE16, 32
GL_STATIC_DRAW, 35	GL_TEXTURE17, 32
GL_STATIC_READ, 35	GL_TEXTURE18, 32
GL_STENCIL, 23	GL_TEXTURE19, 32
GL_STENCIL_BITS, 22	GL_TEXTURE2, 31
	GL_TEXTURE20, 32
	GL_TEXTURE21, 32

GL_TEXTURE22, 32	GL_TEXTURE_COORD_ARRAY_STRIDE, 17
GL_TEXTURE23, 32	GL_TEXTURE_COORD_ARRAY_TYPE, 17
GL_TEXTURE24, 32	GL_TEXTURE_CUBE_MAP, 32
GL_TEXTURE25, 32	GL_TEXTURE_CUBE_MAP_NEGATIVE_X, 32
GL_TEXTURE26, 32	GL_TEXTURE_CUBE_MAP_NEGATIVE_Y, 32
GL_TEXTURE27, 32	GL_TEXTURE_CUBE_MAP_NEGATIVE_Z, 32
GL_TEXTURE28, 32	GL_TEXTURE_CUBE_MAP_POSITIVE_X, 32
GL_TEXTURE29, 32	GL_TEXTURE_CUBE_MAP_POSITIVE_Y, 32
GL_TEXTURE3, 31	GL_TEXTURE_CUBE_MAP_POSITIVE_Z, 32
GL_TEXTURE30, 32	GL_TEXTURE_DEPTH, 29
GL_TEXTURE31, 32	GL_TEXTURE_DEPTH_SIZE, 34
GL_TEXTURE4, 31	GL_TEXTURE_ENV, 26
GL_TEXTURE5, 31	GL_TEXTURE_ENV_COLOR, 26
GL_TEXTURE6, 31	GL_TEXTURE_ENV_MODE, 26
GL_TEXTURE7, 31	GL_TEXTURE_FILTER_CONTROL, 34
GL_TEXTURE8, 31	GL_TEXTURE_GEN_MODE, 26
GL_TEXTURE9, 31	GL_TEXTURE_GEN_Q, 27
GL_TEXTURE_1D, 26	GL_TEXTURE_GEN_R, 27
GL_TEXTURE_2D, 26	GL_TEXTURE_GEN_S, 26
GL_TEXTURE_3D, 29	GL_TEXTURE_GEN_T, 26
GL_TEXTURE_ALPHA_SIZE, 26	GL_TEXTURE_GREEN_SIZE, 26
GL_TEXTURE_BASE_LEVEL, 29	GL_TEXTURE_HEIGHT, 26
GL_TEXTURE_BINDING_1D, 28	GL_TEXTURE_INTENSITY_SIZE, 26
GL_TEXTURE_BINDING_2D, 28	GL_TEXTURE_INTERNAL_FORMAT, 28
GL_TEXTURE_BINDING_3D, 29	GL_TEXTURE_LOD_BIAS, 34
GL_TEXTURE_BINDING_CUBE_MAP, 32	GL_TEXTURE_LUMINANCE_SIZE, 26
GL_TEXTURE_BIT, 27	GL_TEXTURE_MAG_FILTER, 26
GL_TEXTURE_BLUE_SIZE, 26	GL_TEXTURE_MATRIX, 24
GL_TEXTURE_BORDER, 26	GL_TEXTURE_MAX_LEVEL, 29
GL_TEXTURE_BORDER_COLOR, 26	GL_TEXTURE_MAX_LOD, 29
GL_TEXTURE_COMPARE_FUNC, 34	GL_TEXTURE_MIN_FILTER, 26
GL_TEXTURE_COMPARE_MODE, 34	GL_TEXTURE_MIN_LOD, 29
GL_TEXTURE_COMPONENTS, 26	GL_TEXTURE_PRIORITY, 28
GL_TEXTURE_COMPRESSED, 33	GL_TEXTURE_RED_SIZE, 26
GL_TEXTURE_COMPRESSED_IMAGE_SIZE, 33	GL_TEXTURE_RESIDENT, 28
GL_TEXTURE_COMPRESSION_HINT, 33	GL_TEXTURE_STACK_DEPTH, 24
GL_TEXTURE_COORD_ARRAY, 17	GL_TEXTURE_WIDTH, 26
GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING, 35	GL_TEXTURE_WRAP_R, 29
GL_TEXTURE_COORD_ARRAY_POINTER, 18	GL_TEXTURE_WRAP_S, 26
GL_TEXTURE_COORD_ARRAY_SIZE, 17	GL_TEXTURE_WRAP_T, 26
	GL_TRANSFORM_BIT, 27
	GL_TRANSPOSE_COLOR_MATRIX, 32

GL_TRANSPOSE_MODELVIEW_MATRIX, 32	35
GL_TRANSPOSE_PROJECTION_MATRIX, 32	GL_VIEWPORT, 24
GL_TRANSPOSE_TEXTURE_MATRIX, 32	GL_VIEWPORT_BIT, 27
GL_TRIANGLE_FAN, 17	GL_WEIGHT_ARRAY_BUFFER_BINDING, 35
GL_TRIANGLE_STRIP, 17	GL_WRITE_ONLY, 35
GL_TRIANGLES, 17	GL_XOR, 22
GL_TRUE, 17	GL_ZERO, 20
GL_TRUE, 4	GL_ZOOM_X, 26
GL_UNPACK_ALIGNMENT, 26	GL_ZOOM_Y, 26
GL_UNPACK_IMAGE_HEIGHT, 29	glAccum, 7
GL_UNPACK_LSB_FIRST, 26	glActiveTexture, 7
GL_UNPACK_ROW_LENGTH, 26	glAlphaFunc, 7
GL_UNPACK_SKIP_IMAGES, 29	glAreTexturesResident, 5
GL_UNPACK_SKIP_PIXELS, 26	glBegin, 7
GL_UNPACK_SKIP_ROWS, 26	glBeginQuery, 7
GL_UNPACK_SWAP_BYTES, 26	glBindTexture, 7
GL_UNSIGNED_BYTE, 17	GLbitfield, 4
GL_UNSIGNED_BYTE_2_3_3_REV, 29	glBitmap, 7
GL_UNSIGNED_BYTE_3_3_2, 29	glBlendColor, 7
GL_UNSIGNED_INT, 17	glBlendEquation, 7
GL_UNSIGNED_INT_10_10_10_2, 29	glBlendFunc, 7
GL_UNSIGNED_INT_2_10_10_10_REV, 29	glBlendFuncSeparate, 7
GL_UNSIGNED_INT_8_8_8_8, 29	GLboolean, 4
GL_UNSIGNED_INT_8_8_8_8_REV, 29	GLbyte, 4
GL_UNSIGNED_SHORT, 17	glCallList, 7
GL_UNSIGNED_SHORT_1_5_5_5_REV, 29	glCallLists, 7
GL_UNSIGNED_SHORT_4_4_4_4, 29	GLclampd, 4
GL_UNSIGNED_SHORT_4_4_4_4_REV, 29	glClear, 7
GL_UNSIGNED_SHORT_5_5_5_1, 29	glClearAccum, 7
GL_UNSIGNED_SHORT_5_6_5, 29	glClearColor, 7
GL_UNSIGNED_SHORT_5_6_5_REV, 29	glClearDepth, 7
GL_V2F, 18	glClearIndex, 7
GL_V3F, 18	glClearStencil, 7
GL_VENDOR, 27	glClipPlane, 7
GL_VERSION, 27	glColor3b, 7
GL_VERTEX_ARRAY, 17	glColor3bv, 7
GL_VERTEX_ARRAY_BUFFER_BINDING, 34	glColor3d, 7
GL_VERTEX_ARRAY_POINTER, 18	glColor3dv, 7
GL_VERTEX_ARRAY_SIZE, 17	glColor3f, 7
GL_VERTEX_ARRAY_STRIDE, 17	glColor3fv, 7
GL_VERTEX_ARRAY_TYPE, 17	glColor3i, 7
GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING, 35	glColor3iv, 7

<code>glColor3s</code> , 7	<code>glCopyColorSubTable</code> , 8
<code>glColor3sv</code> , 7	<code>glCopyColorTable</code> , 8
<code>glColor3ub</code> , 7	<code>glCopyConvolutionFilter1D</code> , 8
<code>glColor3ubv</code> , 7	<code>glCopyConvolutionFilter2D</code> , 8
<code>glColor3ui</code> , 8	<code>glCopyPixels</code> , 8
<code>glColor3uiv</code> , 8	<code>glCopyTexImage1D</code> , 8
<code>glColor3us</code> , 8	<code>glCopyTexImage2D</code> , 9
<code>glColor3usv</code> , 8	<code>glCopyTexSubImage1D</code> , 9
<code>glColor4b</code> , 8	<code>glCopyTexSubImage2D</code> , 9
<code>glColor4bv</code> , 8	<code>glCopyTexSubImage3D</code> , 9
<code>glColor4d</code> , 8	<code>glCullFace</code> , 9
<code>glColor4dv</code> , 8	<code>glDeleteLists</code> , 9
<code>glColor4f</code> , 8	<code>glDeleteQueries</code> , 5
<code>glColor4fv</code> , 8	<code>glDeleteTextures</code> , 5
<code>glColor4i</code> , 8	<code>glDepthFunc</code> , 9
<code>glColor4iv</code> , 8	<code>glDepthMask</code> , 9
<code>glColor4s</code> , 8	<code>glDepthRange</code> , 9
<code>glColor4sv</code> , 8	<code>glDisable</code> , 9
<code>glColor4ub</code> , 8	<code>GLdouble</code> , 4
<code>glColor4ubv</code> , 8	<code>glDrawBuffer</code> , 9
<code>glColor4ui</code> , 8	<code>glDrawPixels</code> , 9
<code>glColor4uiv</code> , 8	<code>glEdgeFlag</code> , 9
<code>glColor4us</code> , 8	<code>glEdgeFlagv</code> , 9
<code>glColor4usv</code> , 8	<code>glEnable</code> , 9
<code>glColorMask</code> , 8	<code>glEnd</code> , 9
<code>glColorMaterial</code> , 8	<code>glEndList</code> , 9
<code>glColorSubTable</code> , 8	<code>glEndQuery</code> , 9
<code>glColorTable</code> , 8	<code>GLenum</code> , 4
<code>glColorTableParameterfv</code> , 8	<code>glEvalCoord1d</code> , 9
<code>glColorTableParameteriv</code> , 8	<code>glEvalCoord1dv</code> , 9
<code>glCompressedTexImage1D</code> , 8	<code>glEvalCoord1f</code> , 9
<code>glCompressedTexImage2D</code> , 8	<code>glEvalCoord1fv</code> , 9
<code>glCompressedTexImage3D</code> , 8	<code>glEvalCoord2d</code> , 9
<code>glCompressedTexSubImage1D</code> , 8	<code>glEvalCoord2dv</code> , 9
<code>glCompressedTexSubImage2D</code> , 8	<code>glEvalCoord2f</code> , 9
<code>glCompressedTexSubImage3D</code> , 8	<code>glEvalCoord2fv</code> , 9
<code>glConvolutionFilter1D</code> , 8	<code>glEvalMesh1</code> , 9
<code>glConvolutionFilter2D</code> , 8	<code>glEvalMesh2</code> , 9
<code>glConvolutionParameterf</code> , 8	<code>glEvalPoint1</code> , 9
<code>glConvolutionParameterfv</code> , 8	<code>glEvalPoint2</code> , 9
<code>glConvolutionParameteri</code> , 8	<code>glFeedbackBuffer</code> , 7
<code>glConvolutionParameteriv</code> , 8	<code>glFinish</code> , 9

glFlush, 9	glGetPolygonStipple, 10
glFogCoorddd, 9	glGetQueryiv, 6
glFogCoorddv, 9	glGetQueryObjectiv, 6
glFogCoordf, 9	glGetQueryObjectuiv, 6
glFogCoordfv, 9	glGetSeparableFilter, 10
glFogf, 9	glGetString, 6
glFogfv, 9	glGetTexEnvfv, 5
glFogi, 9	glGetTexEnviv, 5
glFogiv, 9	glGetTexGendv, 6
glFrontFace, 9	glGetTexGenfv, 6
glFrustum, 9	glGetTexGeniv, 6
glGenLists, 9	glGetTexImage, 10
glGenQueries, 5	glGetTexLevelParameterfv, 6
glGenTextures, 5	glGetTexLevelParameteriv, 6
glGetBooleanv, 5	glGetTexParameterfv, 6
glGetBufferParameteriv, 6	glGetTexParameteriv, 6
glGetClipPlane, 6	glHint, 10
glGetColorTable, 9	glHistogram, 10
glGetCompressedTexImage, 10	glIndexd, 10
glGetConvolutionFilter, 10	glIndexdv, 10
glGetConvolutionParameterfv, 6	glIndexf, 10
glGetConvolutionParameteriv, 6	glIndexfv, 10
glGetDoublev, 5	glIndexi, 10
glGetError, 10	glIndexiv, 10
glGetFloatv, 5	glIndexMask, 10
glGetHistogram, 10	glIndexsv, 10
glGetHistogramParameterfv, 6	glIndexub, 10
glGetHistogramParameteriv, 6	glIndexubv, 10
glGetIntegerv, 5	glInitNames, 10
glGetLightfv, 5	GLint, 4
glGetLightiv, 5	glIsBuffer, 10
glGetMapdv, 6	glIsEnabled, 10
glGetMapfv, 6	glIsList, 10
glGetMapiv, 6	glIsQuery, 10
glGetMaterialfv, 5	glIsTexture, 10
glGetMaterialiv, 5	glLightf, 10
glGetMinmax, 10	glLightfv, 10
glGetMinmaxParameterfv, 6	glLighti, 10
glGetMinmaxParameteriv, 6	glLightiv, 10
glGetPixelMapfv, 6	glLightModelf, 10
glGetPixelMapuiv, 6	glLightModelfv, 10
glGetPixelMapusv, 6	

<code>glLightModeli</code> , 10	<code>glMultiTexCoord3d</code> , 11
<code>glLightModeliv</code> , 10	<code>glMultiTexCoord3dv</code> , 11
<code>glLineStipple</code> , 10	<code>glMultiTexCoord3f</code> , 11
<code>glLineWidth</code> , 10	<code>glMultiTexCoord3fv</code> , 11
<code>glListBase</code> , 10	<code>glMultiTexCoord3i</code> , 11
<code>glLoadIdentity</code> , 10	<code>glMultiTexCoord3iv</code> , 11
<code>glLoadMatrixd</code> , 10	<code>glMultiTexCoord3s</code> , 11
<code>glLoadMatrixf</code> , 10	<code>glMultiTexCoord3sv</code> , 11
<code>glLoadName</code> , 10	<code>glMultiTexCoord4d</code> , 11
<code>glLoadTransposeMatrixd</code> , 10	<code>glMultiTexCoord4dv</code> , 12
<code>glLoadTransposeMatrixf</code> , 10	<code>glMultiTexCoord4f</code> , 12
<code>glLogicOp</code> , 11	<code>glMultiTexCoord4fv</code> , 12
<code>glMap1d</code> , 11	<code>glMultiTexCoord4i</code> , 12
<code>glMap1f</code> , 11	<code>glMultiTexCoord4iv</code> , 12
<code>glMap2d</code> , 11	<code>glMultiTexCoord4s</code> , 12
<code>glMap2f</code> , 11	<code>glMultiTexCoord4sv</code> , 12
<code>glMapGrid1d</code> , 11	<code>glMultMatrixd</code> , 11
<code>glMapGrid1f</code> , 11	<code>glMultMatrixf</code> , 11
<code>glMapGrid2d</code> , 11	<code>glMultTransposeMatrixd</code> , 11
<code>glMapGrid2f</code> , 11	<code>glMultTransposeMatrixf</code> , 11
<code>glMaterialf</code> , 11	<code>glNewList</code> , 12
<code>glMaterialfv</code> , 11	<code>glNormal3b</code> , 12
<code>glMateriali</code> , 11	<code>glNormal3bv</code> , 12
<code>glMaterialiv</code> , 11	<code>glNormal3d</code> , 12
<code>glMatrixMode</code> , 11	<code>glNormal3dv</code> , 12
<code>glMinmax</code> , 11	<code>glNormal3f</code> , 12
<code>glMultiTexCoord1d</code> , 11	<code>glNormal3fv</code> , 12
<code>glMultiTexCoord1dv</code> , 11	<code>glNormal3i</code> , 12
<code>glMultiTexCoord1f</code> , 11	<code>glNormal3iv</code> , 12
<code>glMultiTexCoord1fv</code> , 11	<code>glNormal3s</code> , 12
<code>glMultiTexCoord1i</code> , 11	<code>glNormal3sv</code> , 12
<code>glMultiTexCoord1iv</code> , 11	<code>glOrtho</code> , 12
<code>glMultiTexCoord1s</code> , 11	<code>glPassThrough</code> , 12
<code>glMultiTexCoord1sv</code> , 11	<code>glPixelMapfv</code> , 5
<code>glMultiTexCoord2d</code> , 11	<code>glPixelMapuiv</code> , 5
<code>glMultiTexCoord2dv</code> , 11	<code>glPixelMapusv</code> , 5
<code>glMultiTexCoord2f</code> , 11	<code>glPixelStoref</code> , 12
<code>glMultiTexCoord2fv</code> , 11	<code>glPixelStorei</code> , 12
<code>glMultiTexCoord2i</code> , 11	<code>glPixelTransferf</code> , 12
<code>glMultiTexCoord2iv</code> , 11	<code>glPixelTransferi</code> , 12
<code>glMultiTexCoord2s</code> , 11	<code>glPixelZoom</code> , 12
<code>glMultiTexCoord2sv</code> , 11	<code>glPointParameterf</code> , 12

<code>glPointParameterfv</code> , 12	<code>glRectdv</code> , 13
<code>glPointParameteri</code> , 12	<code>glRectf</code> , 13
<code>glPointParameteriv</code> , 12	<code>glRectfv</code> , 13
<code>glPointSize</code> , 12	<code>glRecti</code> , 13
<code>glPolygonMode</code> , 12	<code>glRectiv</code> , 13
<code>glPolygonOffset</code> , 12	<code>glRects</code> , 13
<code>glPolygonStipple</code> , 12	<code>glRectsv</code> , 13
<code>glPopAttrib</code> , 12	<code>glRenderMode</code> , 13
<code>glPopClientAttrib</code> , 12	<code>glResetHistogram</code> , 13
<code>glPopMatrix</code> , 12	<code>glResetMinmax</code> , 13
<code>glPopName</code> , 12	<code>glRotated</code> , 13
<code>glPushAttrib</code> , 12	<code>glRotatef</code> , 13
<code>glPushClientAttrib</code> , 12	<code>glSampleCoverage</code> , 13
<code>glPushMatrix</code> , 12	<code>glScaled</code> , 13
<code>glPushName</code> , 12	<code>glScalef</code> , 13
<code>glRasterPos2d</code> , 12	<code>glScissor</code> , 13
<code>glRasterPos2dv</code> , 12	<code>glSecondaryColor3b</code> , 13
<code>glRasterPos2f</code> , 12	<code>glSecondaryColor3bv</code> , 13
<code>glRasterPos2fv</code> , 13	<code>glSecondaryColor3d</code> , 13
<code>glRasterPos2i</code> , 13	<code>glSecondaryColor3dv</code> , 13
<code>glRasterPos2iv</code> , 13	<code>glSecondaryColor3f</code> , 14
<code>glRasterPos2s</code> , 13	<code>glSecondaryColor3fv</code> , 14
<code>glRasterPos2sv</code> , 13	<code>glSecondaryColor3i</code> , 14
<code>glRasterPos3d</code> , 13	<code>glSecondaryColor3iv</code> , 14
<code>glRasterPos3dv</code> , 13	<code>glSecondaryColor3s</code> , 14
<code>glRasterPos3f</code> , 13	<code>glSecondaryColor3sv</code> , 14
<code>glRasterPos3fv</code> , 13	<code>glSecondaryColor3sub</code> , 14
<code>glRasterPos3i</code> , 13	<code>glSecondaryColor3ubv</code> , 14
<code>glRasterPos3iv</code> , 13	<code>glSecondaryColor3ui</code> , 14
<code>glRasterPos3s</code> , 13	<code>glSecondaryColor3uiv</code> , 14
<code>glRasterPos3sv</code> , 13	<code>glSecondaryColor3us</code> , 14
<code>glRasterPos4d</code> , 13	<code>glSecondaryColor3usv</code> , 14
<code>glRasterPos4dv</code> , 13	<code>glSelectBuffer</code> , 7
<code>glRasterPos4f</code> , 13	<code>glSeparableFilter2D</code> , 14
<code>glRasterPos4fv</code> , 13	<code>glShadeModel</code> , 14
<code>glRasterPos4i</code> , 13	<code>GLshort</code> , 4
<code>glRasterPos4iv</code> , 13	<code>GLsizei</code> , 4
<code>glRasterPos4s</code> , 13	<code>glStencilFunc</code> , 14
<code>glRasterPos4sv</code> , 13	<code>glStencilMask</code> , 14
<code>glReadBuffer</code> , 13	<code>glStencilOp</code> , 14
<code>glReadPixels</code> , 13	<code>glTexCoord1d</code> , 14
<code>glRectd</code> , 13	<code>glTexCoord1dv</code> , 14

- [glTexCoord1f](#), 14
- [glTexCoord1fv](#), 14
- [glTexCoord1i](#), 14
- [glTexCoord1iv](#), 14
- [glTexCoord1s](#), 14
- [glTexCoord1sv](#), 14
- [glTexCoord2d](#), 14
- [glTexCoord2dv](#), 14
- [glTexCoord2f](#), 14
- [glTexCoord2fv](#), 14
- [glTexCoord2i](#), 14
- [glTexCoord2iv](#), 14
- [glTexCoord2s](#), 14
- [glTexCoord2sv](#), 14
- [glTexCoord3d](#), 14
- [glTexCoord3dv](#), 14
- [glTexCoord3f](#), 14
- [glTexCoord3fv](#), 14
- [glTexCoord3i](#), 14
- [glTexCoord3iv](#), 14
- [glTexCoord3s](#), 14
- [glTexCoord3sv](#), 14
- [glTexCoord4d](#), 14
- [glTexCoord4dv](#), 14
- [glTexCoord4f](#), 14
- [glTexCoord4fv](#), 15
- [glTexCoord4i](#), 15
- [glTexCoord4iv](#), 15
- [glTexCoord4s](#), 15
- [glTexCoord4sv](#), 15
- [glTexEnvf](#), 15
- [glTexEnvfv](#), 15
- [glTexEnvi](#), 15
- [glTexEnviv](#), 15
- [glTexGend](#), 15
- [glTexGendv](#), 15
- [glTexGenf](#), 15
- [glTexGenfv](#), 15
- [glTexGeni](#), 15
- [glTexGeniv](#), 15
- [glTexImage1D](#), 15
- [glTexImage2D](#), 15

- [glTexImage3D](#), 15
- [glTexParameterf](#), 15
- [glTexParameterfv](#), 15
- [glTexParameteri](#), 15
- [glTexParameteriv](#), 15
- [glTexSubImage1D](#), 15
- [glTexSubImage2D](#), 15
- [glTexSubImage3D](#), 15
- [glTranslated](#), 15
- [glTranslatef](#), 15
- [GLU](#), 1
- [GLU_AUTO_LOAD_MATRIX](#), 37
- [GLU_BEGIN](#), 38
- [GLU_CCW](#), 38
- [GLU_CULLING](#), 37
- [GLU_CW](#), 38
- [GLU_DISPLAY_MODE](#), 37
- [GLU_DOMAIN_DISTANCE](#), 38
- [GLU_EDGE_FLAG](#), 38
- [GLU_END](#), 38
- [GLU_ERROR](#), 36
- [GLU_EXTENSIONS](#), 35
- [GLU_EXTERIOR](#), 38
- [GLU_FALSE](#), 35
- [GLU_FILL](#), 38
- [GLU_FLAT](#), 38
- [GLU_INSIDE](#), 38
- [GLU_INTERIOR](#), 38
- [GLU_INVALID_ENUM](#), 35
- [GLU_INVALID_OPERATION](#), 36
- [GLU_INVALID_VALUE](#), 36
- [GLU_LINE](#), 38
- [GLU_MAP1_TRIM_2](#), 38
- [GLU_MAP1_TRIM_3](#), 38
- [GLU_NONE](#), 38
- [GLU_NURBS_BEGIN](#), 36
- [GLU_NURBS_BEGIN_DATA](#), 36
- [GLU_NURBS_BEGIN_DATA_EXT](#), 36
- [GLU_NURBS_BEGIN_EXT](#), 36
- [GLU_NURBS_COLOR](#), 36
- [GLU_NURBS_COLOR_DATA](#), 36
- [GLU_NURBS_COLOR_DATA_EXT](#), 36

GLU_NURBS_COLOR_EXT, 36
 GLU_NURBS_END, 36
 GLU_NURBS_END_DATA, 36
 GLU_NURBS_END_DATA_EXT, 36
 GLU_NURBS_END_EXT, 36
 GLU_NURBS_ERROR, 36
 GLU_NURBS_ERROR1, 36
 GLU_NURBS_ERROR10, 36
 GLU_NURBS_ERROR11, 36
 GLU_NURBS_ERROR12, 36
 GLU_NURBS_ERROR13, 36
 GLU_NURBS_ERROR14, 37
 GLU_NURBS_ERROR15, 37
 GLU_NURBS_ERROR16, 37
 GLU_NURBS_ERROR17, 37
 GLU_NURBS_ERROR18, 37
 GLU_NURBS_ERROR19, 37
 GLU_NURBS_ERROR2, 36
 GLU_NURBS_ERROR20, 37
 GLU_NURBS_ERROR21, 37
 GLU_NURBS_ERROR22, 37
 GLU_NURBS_ERROR23, 37
 GLU_NURBS_ERROR24, 37
 GLU_NURBS_ERROR25, 37
 GLU_NURBS_ERROR26, 37
 GLU_NURBS_ERROR27, 37
 GLU_NURBS_ERROR28, 37
 GLU_NURBS_ERROR29, 37
 GLU_NURBS_ERROR3, 36
 GLU_NURBS_ERROR30, 37
 GLU_NURBS_ERROR31, 37
 GLU_NURBS_ERROR32, 37
 GLU_NURBS_ERROR33, 37
 GLU_NURBS_ERROR34, 37
 GLU_NURBS_ERROR35, 37
 GLU_NURBS_ERROR36, 37
 GLU_NURBS_ERROR37, 37
 GLU_NURBS_ERROR4, 36
 GLU_NURBS_ERROR5, 36
 GLU_NURBS_ERROR6, 36
 GLU_NURBS_ERROR7, 36
 GLU_NURBS_ERROR8, 36
 GLU_NURBS_ERROR9, 36
 GLU_NURBS_MODE, 37
 GLU_NURBS_MODE_EXT, 37
 GLU_NURBS_NORMAL, 36
 GLU_NURBS_NORMAL_DATA, 36
 GLU_NURBS_NORMAL_DATA_EXT, 36
 GLU_NURBS_NORMAL_EXT, 36
 GLU_NURBS_RENDERER, 37
 GLU_NURBS_RENDERER_EXT, 37
 GLU_NURBS_TESSELLATOR, 37
 GLU_NURBS_TESSELLATOR_EXT, 37
 GLU_NURBS_TEX_COORD_DATA_EXT, 36
 GLU_NURBS_TEX_COORD_EXT, 36
 GLU_NURBS_TEXTURE_COORD, 36
 GLU_NURBS_TEXTURE_COORD_DATA, 36
 GLU_NURBS_VERTEX, 36
 GLU_NURBS_VERTEX_DATA, 36
 GLU_NURBS_VERTEX_DATA_EXT, 36
 GLU_NURBS_VERTEX_EXT, 36
 GLU_OBJECT_PARAMETRIC_ERROR, 37
 GLU_OBJECT_PARAMETRIC_ERROR_EXT, 37
 GLU_OBJECT_PATH_LENGTH, 37
 GLU_OBJECT_PATH_LENGTH_EXT, 37
 GLU_OUT_OF_MEMORY, 36
 GLU_OUTLINE_PATCH, 36
 GLU_OUTLINE_POLYGON, 36
 GLU_OUTSIDE, 38
 GLU_PARAMETRIC_ERROR, 37
 GLU_PARAMETRIC_TOLERANCE, 37
 GLU_PATH_LENGTH, 37
 GLU_POINT, 38
 GLU_SAMPLING_METHOD, 37
 GLU_SAMPLING_TOLERANCE, 37
 GLU_SILHOUETTE, 38
 GLU_SMOOTH, 38
 GLU_TESS_BEGIN, 38
 GLU_TESS_BEGIN_DATA, 38
 GLU_TESS_BOUNDARY_ONLY, 38
 GLU_TESS_COMBINE, 38
 GLU_TESS_COMBINE_DATA, 38
 GLU_TESS_COORD_TOO_LARGE, 39
 GLU_TESS_EDGE_FLAG, 38

GLU_TESS_EDGE_FLAG_DATA , 38	gluCylinder , 16
GLU_TESS_END , 38	gluDisk , 16
GLU_TESS_END_DATA , 38	gluErrorString , 6
GLU_TESS_ERROR , 38	gluGetString , 6
GLU_TESS_ERROR1 , 38	GLuint , 4
GLU_TESS_ERROR2 , 38	gluLookAt , 16
GLU_TESS_ERROR3 , 38	gluNewQuadric , 16
GLU_TESS_ERROR4 , 38	gluOrtho2D , 16
GLU_TESS_ERROR5 , 38	gluPartialDisk , 16
GLU_TESS_ERROR6 , 38	gluPerspective , 16
GLU_TESS_ERROR7 , 38	gluPickMatrix , 16
GLU_TESS_ERROR8 , 38	gluProject , 6
GLU_TESS_ERROR_DATA , 38	gluQuadricDrawStyle , 16
GLU_TESS_MAX_COORD , 39	gluQuadricNormals , 16
GLU_TESS_MISSING_BEGIN_CONTOUR , 39	gluQuadricOrientation , 16
GLU_TESS_MISSING_BEGIN_POLYGON , 39	gluQuadricTexture , 16
GLU_TESS_MISSING_END_CONTOUR , 39	gluScaleImage , 16
GLU_TESS_MISSING_END_POLYGON , 39	GLushort , 4
GLU_TESS_NEED_COMBINE_CALLBACK , 39	gluSphere , 16
GLU_TESS_TOLERANCE , 38	gluUnProject , 6
GLU_TESS_VERTEX , 38	gluUnProject4 , 6
GLU_TESS_VERTEX_DATA , 38	glVertex2d , 15
GLU_TESS_WINDING_ABS_GEQ_TWO , 39	glVertex2dv , 15
GLU_TESS_WINDING_NEGATIVE , 39	glVertex2f , 15
GLU_TESS_WINDING_NONZERO , 39	glVertex2fv , 15
GLU_TESS_WINDING_ODD , 39	glVertex2i , 15
GLU_TESS_WINDING_POSITIVE , 39	glVertex2iv , 15
GLU_TESS_WINDING_RULE , 38	glVertex2s , 15
GLU_TRUE , 35	glVertex2sv , 15
GLU_U_STEP , 37	glVertex3d , 15
GLU_UNKNOWN , 38	glVertex3dv , 15
GLU_V_STEP , 37	glVertex3f , 15
GLU_VERSION , 35	glVertex3fv , 15
GLU_VERTEX , 38	glVertex3i , 15
gluBuild1DMipmapLevels , 16	glVertex3iv , 15
gluBuild1DMipmaps , 16	glVertex3s , 15
gluBuild2DMipmapLevels , 16	glVertex3sv , 15
gluBuild2DMipmaps , 16	glVertex4d , 15
gluBuild3DMipmapLevels , 16	glVertex4dv , 16
gluBuild3DMipmaps , 16	glVertex4f , 16
GLubyte , 4	glVertex4fv , 16
gluCheckExtension , 6	glVertex4i , 16

- [glVertex4iv](#), 16
- [glVertex4s](#), 16
- [glVertex4sv](#), 16
- [glViewport](#), 16
- [glWindowPos2d](#), 16
- [glWindowPos2dv](#), 16
- [glWindowPos2f](#), 16
- [glWindowPos2fv](#), 16
- [glWindowPos2i](#), 16
- [glWindowPos2iv](#), 16
- [glWindowPos2s](#), 16
- [glWindowPos2sv](#), 16
- [glWindowPos3d](#), 16
- [glWindowPos3dv](#), 16
- [glWindowPos3f](#), 16
- [glWindowPos3fv](#), 16
- [glWindowPos3i](#), 16
- [glWindowPos3iv](#), 16
- [glWindowPos3s](#), 16
- [glWindowPos3sv](#), 16
- Initialization, 52
- [list->gl-boolean-vector](#), 49
- [list->gl-byte-vector](#), 45
- [list->gl-double-vector](#), 49
- [list->gl-float-vector](#), 48
- [list->gl-int-vector](#), 47
- [list->gl-short-vector](#), 46
- [list->gl-ubyte-vector](#), 46
- [list->gl-uint-vector](#), 48
- [list->gl-ushort-vector](#), 47
- [make-gl-boolean-vector](#), 49
- [make-gl-byte-vector](#), 45
- [make-gl-double-vector](#), 49
- [make-gl-float-vector](#), 48
- [make-gl-int-vector](#), 47
- [make-gl-selection-record](#), 40
- [make-gl-short-vector](#), 46
- [make-gl-ubyte-vector](#), 46
- [make-gl-uint-vector](#), 48
- [make-gl-ushort-vector](#), 47
- OpenGL, 1
- OpenGL Vectors, 45
- Racket-Style OpenGL, 40
- [select-buffer->gl-uint-vector](#), 39
- [sgl](#), 40
- [sgl/bitmap](#), 51
- [sgl/gl](#), 4
- [sgl/gl-vectors](#), 45
- [sgl/init](#), 52
- [struct:gl-selection-record](#), 40
- Using OpenGL, 3
- [vector->gl-boolean-vector](#), 49
- [vector->gl-byte-vector](#), 45
- [vector->gl-double-vector](#), 49
- [vector->gl-float-vector](#), 48
- [vector->gl-int-vector](#), 47
- [vector->gl-short-vector](#), 46
- [vector->gl-ubyte-vector](#), 46
- [vector->gl-uint-vector](#), 48
- [vector->gl-ushort-vector](#), 47