```
#ifndef __gl_h_
#define __gl_h_

#if defined(USE_MGL_NAMESPACE)
#include "gl_mangle.h"
#endif


/**********************************************************************
 * Begin system-specific stuff.
 */

#if defined(_WIN32) && !defined(__WIN32__) && !defined(__CYGWIN__)
#define __WIN32__
#endif

#if defined(__WIN32__) && !defined(__CYGWIN__)
#  if (defined(_MSC_VER) || defined(__MINGW32__)) && defined(BUILD_GL32) /* tag specify we're building mesa as a DLL */
#    define GLAPI __declspec(dllexport)
#  elif (defined(_MSC_VER) || defined(__MINGW32__)) && defined(_DLL) /* tag specifying we're building for DLL runtime support */
#    define GLAPI __declspec(dllimport)
#  else /* for use with static link lib build of Win32 edition only */
#    define GLAPI extern
#  endif /* _STATIC_MESA support */
#  if defined(__MINGW32__) && defined(GL_NO_STDCALL) || defined(UNDER_CE)  /* The generated DLLs by MingW with STDCALL are not compatible with the ones done by Microsoft's compilers */
#    define GLAPIENTRY
#  else
#    define GLAPIENTRY __stdcall
#  endif
#elif defined(__CYGWIN__) && defined(USE_OPENGL32) /* use native windows opengl32 */
#  define GLAPI extern
#  define GLAPIENTRY __stdcall
#elif (defined(__GNUC__) && __GNUC__ >= 4) || (defined(__SUNPRO_C) && (__SUNPRO_C >= 0x590))
#  define GLAPI __attribute__((visibility("default")))
#  define GLAPIENTRY
#endif /* WIN32 && !CYGWIN */

/*
 * WINDOWS: Include windows.h here to define APIENTRY.
 * It is also useful when applications include this file by
 * including only glut.h, since glut.h depends on windows.h.
 * Applications needing to include windows.h with parms other
 * than "WIN32_LEAN_AND_MEAN" may include windows.h before
 * glut.h or gl.h.
 */
#if defined(_WIN32) && !defined(APIENTRY) && !defined(__CYGWIN__)
#ifndef WIN32_LEAN_AND_MEAN
#define WIN32_LEAN_AND_MEAN 1
#endif
#include <windows.h>
#endif

#ifndef GLAPI
#define GLAPI extern
#endif

#ifndef GLAPIENTRY
#define GLAPIENTRY
#endif

#ifndef APIENTRY
#define APIENTRY GLAPIENTRY
#endif

/* "P" suffix to be used for a pointer to a function */
#ifndef APIENTRYP
#define APIENTRYP APIENTRY *
#endif

#ifndef GLAPIENTRYP
```

```
#define GLAPIENTRYP GLAPIENTRY *
#endif

/*
 * End system-specific stuff.
 **********************************************************************/



#ifdef __cplusplus
extern "C" {
#endif



#define GL_VERSION_1_1   1
#define GL_VERSION_1_2   1
#define GL_VERSION_1_3   1
#define GL_ARB_imaging   1


/*
 * Datatypes
 */
typedef unsigned int     GLenum;
typedef unsigned char    GLboolean;
typedef unsigned int     GLbitfield;
typedef void             GLvoid;
typedef signed char      GLbyte;         /* 1-byte signed */
typedef short            GLshort;  /* 2-byte signed */
typedef int              GLint;          /* 4-byte signed */
typedef unsigned char    GLubyte; /* 1-byte unsigned */
typedef unsigned short GLushort;         /* 2-byte unsigned */
typedef unsigned int     GLuint;         /* 4-byte unsigned */
typedef int              GLsizei;  /* 4-byte signed */
typedef float            GLfloat;  /* single precision float */
typedef float            GLclampf;      /* single precision float in [0,1] */
typedef double           GLdouble;       /* double precision float */
typedef double           GLclampd;       /* double precision float in [0,1] */



/*
 * Constants
 */

/* Boolean values */
#define GL_FALSE                          0
#define GL_TRUE                           1

/* Data types */
#define GL_BYTE                           0x1400
#define GL_UNSIGNED_BYTE                         0x1401
#define GL_SHORT                          0x1402
```

```
#define GL_UNSIGNED_SHORT        0x1403
#define GL_INT               0x1404
#define GL_UNSIGNED_INT             0x1405
#define GL_FLOAT             0x1406
#define GL_2_BYTES           0x1407
#define GL_3_BYTES           0x1408
#define GL_4_BYTES           0x1409
#define GL_DOUBLE            0x140A

/* Primitives */
#define GL_POINTS            0x0000
#define GL_LINES          0x0001
#define GL_LINE_LOOP             0x0002
#define GL_LINE_STRIP            0x0003
#define GL_TRIANGLES             0x0004
#define GL_TRIANGLE_STRIP        0x0005
#define GL_TRIANGLE_FAN             0x0006
#define GL_QUADS             0x0007
#define GL_QUAD_STRIP            0x0008
#define GL_POLYGON           0x0009

/* Vertex Arrays */
#define GL_VERTEX_ARRAY             0x8074
#define GL_NORMAL_ARRAY             0x8075
#define GL_COLOR_ARRAY              0x8076
#define GL_INDEX_ARRAY              0x8077
#define GL_TEXTURE_COORD_ARRAY          0x8078
#define GL_EDGE_FLAG_ARRAY          0x8079
#define GL_VERTEX_ARRAY_SIZE        0x807A
#define GL_VERTEX_ARRAY_TYPE        0x807B
#define GL_VERTEX_ARRAY_STRIDE          0x807C
#define GL_NORMAL_ARRAY_TYPE            0x807E
#define GL_NORMAL_ARRAY_STRIDE          0x807F
#define GL_COLOR_ARRAY_SIZE         0x8081
#define GL_COLOR_ARRAY_TYPE         0x8082
#define GL_COLOR_ARRAY_STRIDE           0x8083
#define GL_INDEX_ARRAY_TYPE         0x8085
#define GL_INDEX_ARRAY_STRIDE           0x8086
#define GL_TEXTURE_COORD_ARRAY_SIZE         0x8088
#define GL_TEXTURE_COORD_ARRAY_TYPE         0x8089
#define GL_TEXTURE_COORD_ARRAY_STRIDE       0x808A
#define GL_EDGE_FLAG_ARRAY_STRIDE       0x808C
#define GL_VERTEX_ARRAY_POINTER         0x808E
#define GL_NORMAL_ARRAY_POINTER             0x808F
#define GL_COLOR_ARRAY_POINTER          0x8090
#define GL_INDEX_ARRAY_POINTER          0x8091
#define GL_TEXTURE_COORD_ARRAY_POINTER         0x8092
#define GL_EDGE_FLAG_ARRAY_POINTER      0x8093
#define GL_V2F               0x2A20
#define GL_V3F               0x2A21
#define GL_C4UB_V2F          0x2A22
#define GL_C4UB_V3F          0x2A23
#define GL_C3F_V3F           0x2A24
#define GL_N3F_V3F           0x2A25
```

```
#define GL_C4F_N3F_V3F             0x2A26
#define GL_T2F_V3F             0x2A27
#define GL_T4F_V4F             0x2A28
#define GL_T2F_C4UB_V3F                0x2A29
#define GL_T2F_C3F_V3F             0x2A2A
#define GL_T2F_N3F_V3F             0x2A2B
#define GL_T2F_C4F_N3F_V3F         0x2A2C
#define GL_T4F_C4F_N3F_V4F         0x2A2D

/* Matrix Mode */
#define GL_MATRIX_MODE                 0x0BA0
#define GL_MODELVIEW               0x1700
#define GL_PROJECTION              0x1701
#define GL_TEXTURE             0x1702

/* Points */
#define GL_POINT_SMOOTH                0x0B10
#define GL_POINT_SIZE              0x0B11
#define GL_POINT_SIZE_GRANULARITY          0x0B13
#define GL_POINT_SIZE_RANGE            0x0B12

/* Lines */
#define GL_LINE_SMOOTH                 0x0B20
#define GL_LINE_STIPPLE            0x0B24
#define GL_LINE_STIPPLE_PATTERN            0x0B25
#define GL_LINE_STIPPLE_REPEAT         0x0B26
#define GL_LINE_WIDTH              0x0B21
#define GL_LINE_WIDTH_GRANULARITY          0x0B23
#define GL_LINE_WIDTH_RANGE            0x0B22

/* Polygons */
#define GL_POINT               0x1B00
#define GL_LINE                0x1B01
#define GL_FILL                0x1B02
#define GL_CW                  0x0900
#define GL_CCW                 0x0901
#define GL_FRONT               0x0404
#define GL_BACK                0x0405
#define GL_POLYGON_MODE                0x0B40
#define GL_POLYGON_SMOOTH              0x0B41
#define GL_POLYGON_STIPPLE             0x0B42
#define GL_EDGE_FLAG               0x0B43
#define GL_CULL_FACE               0x0B44
#define GL_CULL_FACE_MODE              0x0B45
#define GL_FRONT_FACE              0x0B46
#define GL_POLYGON_OFFSET_FACTOR           0x8038
#define GL_POLYGON_OFFSET_UNITS            0x2A00
#define GL_POLYGON_OFFSET_POINT            0x2A01
#define GL_POLYGON_OFFSET_LINE             0x2A02
#define GL_POLYGON_OFFSET_FILL             0x8037

/* Display Lists */
#define GL_COMPILE             0x1300
#define GL_COMPILE_AND_EXECUTE             0x1301
```

```
#define GL_LIST_BASE                         0x0B32
#define GL_LIST_INDEX                        0x0B33
#define GL_LIST_MODE                         0x0B30

/* Depth buffer */
#define GL_NEVER                        0x0200
#define GL_LESS                         0x0201
#define GL_EQUAL                        0x0202
#define GL_LEQUAL                       0x0203
#define GL_GREATER                      0x0204
#define GL_NOTEQUAL                          0x0205
#define GL_GEQUAL                       0x0206
#define GL_ALWAYS                       0x0207
#define GL_DEPTH_TEST                        0x0B71
#define GL_DEPTH_BITS                        0x0D56
#define GL_DEPTH_CLEAR_VALUE                      0x0B73
#define GL_DEPTH_FUNC                        0x0B74
#define GL_DEPTH_RANGE                            0x0B70
#define GL_DEPTH_WRITEMASK                        0x0B72
#define GL_DEPTH_COMPONENT                        0x1902

/* Lighting */
#define GL_LIGHTING                     0x0B50
#define GL_LIGHT0                       0x4000
#define GL_LIGHT1                       0x4001
#define GL_LIGHT2                       0x4002
#define GL_LIGHT3                       0x4003
#define GL_LIGHT4                       0x4004
#define GL_LIGHT5                       0x4005
#define GL_LIGHT6                       0x4006
#define GL_LIGHT7                       0x4007
#define GL_SPOT_EXPONENT                     0x1205
#define GL_SPOT_CUTOFF                       0x1206
#define GL_CONSTANT_ATTENUATION                        0x1207
#define GL_LINEAR_ATTENUATION                          0x1208
#define GL_QUADRATIC_ATTENUATION                       0x1209
#define GL_AMBIENT                      0x1200
#define GL_DIFFUSE                      0x1201
#define GL_SPECULAR                          0x1202
#define GL_SHININESS                         0x1601
#define GL_EMISSION                     0x1600
#define GL_POSITION                     0x1203
#define GL_SPOT_DIRECTION                    0x1204
#define GL_AMBIENT_AND_DIFFUSE                         0x1602
#define GL_COLOR_INDEXES                     0x1603
#define GL_LIGHT_MODEL_TWO_SIDE                        0x0B52
#define GL_LIGHT_MODEL_LOCAL_VIEWER                         0x0B51
#define GL_LIGHT_MODEL_AMBIENT                         0x0B53
#define GL_FRONT_AND_BACK                         0x0408
#define GL_SHADE_MODEL                            0x0B54
#define GL_FLAT                         0x1D00
#define GL_SMOOTH                       0x1D01
#define GL_COLOR_MATERIAL                         0x0B57
#define GL_COLOR_MATERIAL_FACE                         0x0B55
```

```
#define GL_COLOR_MATERIAL_PARAMETER                                     0x0B56
#define GL_NORMALIZE                            0x0BA1

/* User clipping planes */
#define GL_CLIP_PLANE0                          0x3000
#define GL_CLIP_PLANE1                          0x3001
#define GL_CLIP_PLANE2                          0x3002
#define GL_CLIP_PLANE3                          0x3003
#define GL_CLIP_PLANE4                          0x3004
#define GL_CLIP_PLANE5                          0x3005

/* Accumulation buffer */
#define GL_ACCUM_RED_BITS                       0x0D58
#define GL_ACCUM_GREEN_BITS                             0x0D59
#define GL_ACCUM_BLUE_BITS                              0x0D5A
#define GL_ACCUM_ALPHA_BITS                             0x0D5B
#define GL_ACCUM_CLEAR_VALUE                                    0x0B80
#define GL_ACCUM                        0x0100
#define GL_ADD                          0x0104
#define GL_LOAD                         0x0101
#define GL_MULT                         0x0103
#define GL_RETURN                       0x0102

/* Alpha testing */
#define GL_ALPHA_TEST                           0x0BC0
#define GL_ALPHA_TEST_REF                       0x0BC2
#define GL_ALPHA_TEST_FUNC                              0x0BC1

/* Blending */
#define GL_BLEND                        0x0BE2
#define GL_BLEND_SRC                            0x0BE1
#define GL_BLEND_DST                            0x0BE0
#define GL_ZERO                         0
#define GL_ONE                          1
#define GL_SRC_COLOR                            0x0300
#define GL_ONE_MINUS_SRC_COLOR                                  0x0301
#define GL_SRC_ALPHA                            0x0302
#define GL_ONE_MINUS_SRC_ALPHA                                  0x0303
#define GL_DST_ALPHA                            0x0304
#define GL_ONE_MINUS_DST_ALPHA                                  0x0305
#define GL_DST_COLOR                            0x0306
#define GL_ONE_MINUS_DST_COLOR                                  0x0307
#define GL_SRC_ALPHA_SATURATE                                   0x0308

/* Render Mode */
#define GL_FEEDBACK                             0x1C01
#define GL_RENDER                       0x1C00
#define GL_SELECT                       0x1C02

/* Feedback */
#define GL_2D                           0x0600
#define GL_3D                           0x0601
#define GL_3D_COLOR                             0x0602
#define GL_3D_COLOR_TEXTURE                             0x0603
```

```
#define GL_4D_COLOR_TEXTURE         0x0604
#define GL_POINT_TOKEN          0x0701
#define GL_LINE_TOKEN           0x0702
#define GL_LINE_RESET_TOKEN         0x0707
#define GL_POLYGON_TOKEN        0x0703
#define GL_BITMAP_TOKEN             0x0704
#define GL_DRAW_PIXEL_TOKEN         0x0705
#define GL_COPY_PIXEL_TOKEN         0x0706
#define GL_PASS_THROUGH_TOKEN          0x0700
#define GL_FEEDBACK_BUFFER_POINTER     0x0DF0
#define GL_FEEDBACK_BUFFER_SIZE        0x0DF1
#define GL_FEEDBACK_BUFFER_TYPE        0x0DF2

/* Selection */
#define GL_SELECTION_BUFFER_POINTER    0x0DF3
#define GL_SELECTION_BUFFER_SIZE    0x0DF4

/* Fog */
#define GL_FOG               0x0B60
#define GL_FOG_MODE             0x0B65
#define GL_FOG_DENSITY          0x0B62
#define GL_FOG_COLOR            0x0B66
#define GL_FOG_INDEX            0x0B61
#define GL_FOG_START            0x0B63
#define GL_FOG_END           0x0B64
#define GL_LINEAR            0x2601
#define GL_EXP               0x0800
#define GL_EXP2              0x0801

/* Logic Ops */
#define GL_LOGIC_OP          0x0BF1
#define GL_INDEX_LOGIC_OP       0x0BF1
#define GL_COLOR_LOGIC_OP       0x0BF2
#define GL_LOGIC_OP_MODE        0x0BF0
#define GL_CLEAR             0x1500
#define GL_SET               0x150F
#define GL_COPY              0x1503
#define GL_COPY_INVERTED        0x150C
#define GL_NOOP              0x1505
#define GL_INVERT            0x150A
#define GL_AND               0x1501
#define GL_NAND              0x150E
#define GL_OR                0x1507
#define GL_NOR               0x1508
#define GL_XOR               0x1506
#define GL_EQUIV             0x1509
#define GL_AND_REVERSE              0x1502
#define GL_AND_INVERTED             0x1504
#define GL_OR_REVERSE           0x150B
#define GL_OR_INVERTED          0x150D

/* Stencil */
#define GL_STENCIL_BITS         0x0D57
#define GL_STENCIL_TEST             0x0B90
```

```
#define GL_STENCIL_CLEAR_VALUE                          0x0B91
#define GL_STENCIL_FUNC                             0x0B92
#define GL_STENCIL_VALUE_MASK                           0x0B93
#define GL_STENCIL_FAIL                         0x0B94
#define GL_STENCIL_PASS_DEPTH_FAIL                      0x0B95
#define GL_STENCIL_PASS_DEPTH_PASS                      0x0B96
#define GL_STENCIL_REF                          0x0B97
#define GL_STENCIL_WRITEMASK                        0x0B98
#define GL_STENCIL_INDEX                        0x1901
#define GL_KEEP                             0x1E00
#define GL_REPLACE                          0x1E01
#define GL_INCR                             0x1E02
#define GL_DECR                             0x1E03

/* Buffers, Pixel Drawing/Reading */
#define GL_NONE                             0
#define GL_LEFT                             0x0406
#define GL_RIGHT                            0x0407
/*GL_FRONT                          0x0404 */
/*GL_BACK                           0x0405 */
/*GL_FRONT_AND_BACK                         0x0408 */
#define GL_FRONT_LEFT                           0x0400
#define GL_FRONT_RIGHT                          0x0401
#define GL_BACK_LEFT                            0x0402
#define GL_BACK_RIGHT                           0x0403
#define GL_AUX0                             0x0409
#define GL_AUX1                             0x040A
#define GL_AUX2                             0x040B
#define GL_AUX3                             0x040C
#define GL_COLOR_INDEX                              0x1900
#define GL_RED                              0x1903
#define GL_GREEN                            0x1904
#define GL_BLUE                             0x1905
#define GL_ALPHA                            0x1906
#define GL_LUMINANCE                            0x1909
#define GL_LUMINANCE_ALPHA                          0x190A
#define GL_ALPHA_BITS                           0x0D55
#define GL_RED_BITS                         0x0D52
#define GL_GREEN_BITS                           0x0D53
#define GL_BLUE_BITS                            0x0D54
#define GL_INDEX_BITS                           0x0D51
#define GL_SUBPIXEL_BITS                        0x0D50
#define GL_AUX_BUFFERS                              0x0C00
#define GL_READ_BUFFER                              0x0C02
#define GL_DRAW_BUFFER                              0x0C01
#define GL_DOUBLEBUFFER                             0x0C32
#define GL_STEREO                           0x0C33
#define GL_BITMAP                           0x1A00
#define GL_COLOR                            0x1800
#define GL_DEPTH                            0x1801
#define GL_STENCIL                          0x1802
#define GL_DITHER                           0x0BD0
#define GL_RGB                              0x1907
#define GL_RGBA                             0x1908
```

```
/* Implementation limits */
#define GL_MAX_LIST_NESTING       0x0B31
#define GL_MAX_EVAL_ORDER         0x0D30
#define GL_MAX_LIGHTS         0x0D31
#define GL_MAX_CLIP_PLANES        0x0D32
#define GL_MAX_TEXTURE_SIZE       0x0D33
#define GL_MAX_PIXEL_MAP_TABLE        0x0D34
#define GL_MAX_ATTRIB_STACK_DEPTH     0x0D35
#define GL_MAX_MODELVIEW_STACK_DEPTH      0x0D36
#define GL_MAX_NAME_STACK_DEPTH           0x0D37
#define GL_MAX_PROJECTION_STACK_DEPTH     0x0D38
#define GL_MAX_TEXTURE_STACK_DEPTH    0x0D39
#define GL_MAX_VIEWPORT_DIMS          0x0D3A
#define GL_MAX_CLIENT_ATTRIB_STACK_DEPTH  0x0D3B

/* Gets */
#define GL_ATTRIB_STACK_DEPTH         0x0BB0
#define GL_CLIENT_ATTRIB_STACK_DEPTH      0x0BB1
#define GL_COLOR_CLEAR_VALUE          0x0C22
#define GL_COLOR_WRITEMASK        0x0C23
#define GL_CURRENT_INDEX      0x0B01
#define GL_CURRENT_COLOR      0x0B00
#define GL_CURRENT_NORMAL         0x0B02
#define GL_CURRENT_RASTER_COLOR       0x0B04
#define GL_CURRENT_RASTER_DISTANCE    0x0B09
#define GL_CURRENT_RASTER_INDEX       0x0B05
#define GL_CURRENT_RASTER_POSITION    0x0B07
#define GL_CURRENT_RASTER_TEXTURE_COORDS 0x0B06
#define GL_CURRENT_RASTER_POSITION_VALID  0x0B08
#define GL_CURRENT_TEXTURE_COORDS     0x0B03
#define GL_INDEX_CLEAR_VALUE      0x0C20
#define GL_INDEX_MODE         0x0C30
#define GL_INDEX_WRITEMASK        0x0C21
#define GL_MODELVIEW_MATRIX       0x0BA6
#define GL_MODELVIEW_STACK_DEPTH      0x0BA3
#define GL_NAME_STACK_DEPTH       0x0D70
#define GL_PROJECTION_MATRIX      0x0BA7
#define GL_PROJECTION_STACK_DEPTH     0x0BA4
#define GL_RENDER_MODE            0x0C40
#define GL_RGBA_MODE          0x0C31
#define GL_TEXTURE_MATRIX     0x0BA8
#define GL_TEXTURE_STACK_DEPTH        0x0BA5
#define GL_VIEWPORT           0x0BA2

/* Evaluators */
#define GL_AUTO_NORMAL            0x0D80
#define GL_MAP1_COLOR_4           0x0D90
#define GL_MAP1_INDEX         0x0D91
#define GL_MAP1_NORMAL            0x0D92
#define GL_MAP1_TEXTURE_COORD_1       0x0D93
#define GL_MAP1_TEXTURE_COORD_2       0x0D94
#define GL_MAP1_TEXTURE_COORD_3       0x0D95
#define GL_MAP1_TEXTURE_COORD_4       0x0D96
```

```
#define GL_MAP1_VERTEX_3            0x0D97
#define GL_MAP1_VERTEX_4            0x0D98
#define GL_MAP2_COLOR_4                 0x0DB0
#define GL_MAP2_INDEX               0x0DB1
#define GL_MAP2_NORMAL                  0x0DB2
#define GL_MAP2_TEXTURE_COORD_1             0x0DB3
#define GL_MAP2_TEXTURE_COORD_2             0x0DB4
#define GL_MAP2_TEXTURE_COORD_3             0x0DB5
#define GL_MAP2_TEXTURE_COORD_4             0x0DB6
#define GL_MAP2_VERTEX_3            0x0DB7
#define GL_MAP2_VERTEX_4            0x0DB8
#define GL_MAP1_GRID_DOMAIN             0x0DD0
#define GL_MAP1_GRID_SEGMENTS               0x0DD1
#define GL_MAP2_GRID_DOMAIN             0x0DD2
#define GL_MAP2_GRID_SEGMENTS               0x0DD3
#define GL_COEFF                0x0A00
#define GL_ORDER                0x0A01
#define GL_DOMAIN               0x0A02

/* Hints */
#define GL_PERSPECTIVE_CORRECTION_HINT          0x0C50
#define GL_POINT_SMOOTH_HINT            0x0C51
#define GL_LINE_SMOOTH_HINT             0x0C52
#define GL_POLYGON_SMOOTH_HINT              0x0C53
#define GL_FOG_HINT             0x0C54
#define GL_DONT_CARE                0x1100
#define GL_FASTEST              0x1101
#define GL_NICEST               0x1102

/* Scissor box */
#define GL_SCISSOR_BOX              0x0C10
#define GL_SCISSOR_TEST                 0x0C11

/* Pixel Mode / Transfer */
#define GL_MAP_COLOR                0x0D10
#define GL_MAP_STENCIL              0x0D11
#define GL_INDEX_SHIFT              0x0D12
#define GL_INDEX_OFFSET                 0x0D13
#define GL_RED_SCALE                0x0D14
#define GL_RED_BIAS             0x0D15
#define GL_GREEN_SCALE                  0x0D18
#define GL_GREEN_BIAS               0x0D19
#define GL_BLUE_SCALE               0x0D1A
#define GL_BLUE_BIAS                0x0D1B
#define GL_ALPHA_SCALE                  0x0D1C
#define GL_ALPHA_BIAS               0x0D1D
#define GL_DEPTH_SCALE              0x0D1E
#define GL_DEPTH_BIAS               0x0D1F
#define GL_PIXEL_MAP_S_TO_S_SIZE        0x0CB1
#define GL_PIXEL_MAP_I_TO_I_SIZE        0x0CB0
#define GL_PIXEL_MAP_I_TO_R_SIZE        0x0CB2
#define GL_PIXEL_MAP_I_TO_G_SIZE        0x0CB3
#define GL_PIXEL_MAP_I_TO_B_SIZE        0x0CB4
#define GL_PIXEL_MAP_I_TO_A_SIZE        0x0CB5
```

```
#define GL_PIXEL_MAP_R_TO_R_SIZE      0x0CB6
#define GL_PIXEL_MAP_G_TO_G_SIZE      0x0CB7
#define GL_PIXEL_MAP_B_TO_B_SIZE      0x0CB8
#define GL_PIXEL_MAP_A_TO_A_SIZE      0x0CB9
#define GL_PIXEL_MAP_S_TO_S           0x0C71
#define GL_PIXEL_MAP_I_TO_I           0x0C70
#define GL_PIXEL_MAP_I_TO_R           0x0C72
#define GL_PIXEL_MAP_I_TO_G           0x0C73
#define GL_PIXEL_MAP_I_TO_B           0x0C74
#define GL_PIXEL_MAP_I_TO_A           0x0C75
#define GL_PIXEL_MAP_R_TO_R           0x0C76
#define GL_PIXEL_MAP_G_TO_G           0x0C77
#define GL_PIXEL_MAP_B_TO_B           0x0C78
#define GL_PIXEL_MAP_A_TO_A           0x0C79
#define GL_PACK_ALIGNMENT             0x0D05
#define GL_PACK_LSB_FIRST         0x0D01
#define GL_PACK_ROW_LENGTH            0x0D02
#define GL_PACK_SKIP_PIXELS           0x0D04
#define GL_PACK_SKIP_ROWS         0x0D03
#define GL_PACK_SWAP_BYTES            0x0D00
#define GL_UNPACK_ALIGNMENT           0x0CF5
#define GL_UNPACK_LSB_FIRST           0x0CF1
#define GL_UNPACK_ROW_LENGTH              0x0CF2
#define GL_UNPACK_SKIP_PIXELS         0x0CF4
#define GL_UNPACK_SKIP_ROWS           0x0CF3
#define GL_UNPACK_SWAP_BYTES          0x0CF0
#define GL_ZOOM_X                 0x0D16
#define GL_ZOOM_Y                 0x0D17

/* Texture mapping */
#define GL_TEXTURE_ENV                0x2300
#define GL_TEXTURE_ENV_MODE           0x2200
#define GL_TEXTURE_1D             0x0DE0
#define GL_TEXTURE_2D             0x0DE1
#define GL_TEXTURE_WRAP_S             0x2802
#define GL_TEXTURE_WRAP_T             0x2803
#define GL_TEXTURE_MAG_FILTER             0x2800
#define GL_TEXTURE_MIN_FILTER         0x2801
#define GL_TEXTURE_ENV_COLOR              0x2201
#define GL_TEXTURE_GEN_S          0x0C60
#define GL_TEXTURE_GEN_T          0x0C61
#define GL_TEXTURE_GEN_R          0x0C62
#define GL_TEXTURE_GEN_Q          0x0C63
#define GL_TEXTURE_GEN_MODE           0x2500
#define GL_TEXTURE_BORDER_COLOR               0x1004
#define GL_TEXTURE_WIDTH          0x1000
#define GL_TEXTURE_HEIGHT         0x1001
#define GL_TEXTURE_BORDER             0x1005
#define GL_TEXTURE_COMPONENTS             0x1003
#define GL_TEXTURE_RED_SIZE           0x805C
#define GL_TEXTURE_GREEN_SIZE             0x805D
#define GL_TEXTURE_BLUE_SIZE          0x805E
#define GL_TEXTURE_ALPHA_SIZE             0x805F
#define GL_TEXTURE_LUMINANCE_SIZE         0x8060
```

```
#define GL_TEXTURE_INTENSITY_SIZE     0x8061
#define GL_NEAREST_MIPMAP_NEAREST         0x2700
#define GL_NEAREST_MIPMAP_LINEAR          0x2702
#define GL_LINEAR_MIPMAP_NEAREST          0x2701
#define GL_LINEAR_MIPMAP_LINEAR           0x2703
#define GL_OBJECT_LINEAR          0x2401
#define GL_OBJECT_PLANE               0x2501
#define GL_EYE_LINEAR             0x2400
#define GL_EYE_PLANE              0x2502
#define GL_SPHERE_MAP             0x2402
#define GL_DECAL              0x2101
#define GL_MODULATE               0x2100
#define GL_NEAREST            0x2600
#define GL_REPEAT             0x2901
#define GL_CLAMP              0x2900
#define GL_S              0x2000
#define GL_T              0x2001
#define GL_R              0x2002
#define GL_Q              0x2003

/* Utility */
#define GL_VENDOR             0x1F00
#define GL_RENDERER               0x1F01
#define GL_VERSION            0x1F02
#define GL_EXTENSIONS             0x1F03

/* Errors */
#define GL_NO_ERROR               0
#define GL_INVALID_ENUM               0x0500
#define GL_INVALID_VALUE          0x0501
#define GL_INVALID_OPERATION          0x0502
#define GL_STACK_OVERFLOW             0x0503
#define GL_STACK_UNDERFLOW            0x0504
#define GL_OUT_OF_MEMORY          0x0505

/* glPush/PopAttrib bits */
#define GL_CURRENT_BIT            0x00000001
#define GL_POINT_BIT              0x00000002
#define GL_LINE_BIT           0x00000004
#define GL_POLYGON_BIT            0x00000008
#define GL_POLYGON_STIPPLE_BIT            0x00000010
#define GL_PIXEL_MODE_BIT         0x00000020
#define GL_LIGHTING_BIT           0x00000040
#define GL_FOG_BIT            0x00000080
#define GL_DEPTH_BUFFER_BIT           0x00000100
#define GL_ACCUM_BUFFER_BIT           0x00000200
#define GL_STENCIL_BUFFER_BIT         0x00000400
#define GL_VIEWPORT_BIT               0x00000800
#define GL_TRANSFORM_BIT          0x00001000
#define GL_ENABLE_BIT             0x00002000
#define GL_COLOR_BUFFER_BIT           0x00004000
#define GL_HINT_BIT           0x00008000
#define GL_EVAL_BIT           0x00010000
#define GL_LIST_BIT           0x00020000
```

```
#define GL_TEXTURE_BIT            0x00040000
#define GL_SCISSOR_BIT            0x00080000
#define GL_ALL_ATTRIB_BITS        0xFFFFFFFF


/* OpenGL 1.1 */
#define GL_PROXY_TEXTURE_1D          0x8063
#define GL_PROXY_TEXTURE_2D          0x8064
#define GL_TEXTURE_PRIORITY          0x8066
#define GL_TEXTURE_RESIDENT          0x8067
#define GL_TEXTURE_BINDING_1D            0x8068
#define GL_TEXTURE_BINDING_2D            0x8069
#define GL_TEXTURE_INTERNAL_FORMAT       0x1003
#define GL_ALPHA4             0x803B
#define GL_ALPHA8             0x803C
#define GL_ALPHA12            0x803D
#define GL_ALPHA16            0x803E
#define GL_LUMINANCE4             0x803F
#define GL_LUMINANCE8             0x8040
#define GL_LUMINANCE12               0x8041
#define GL_LUMINANCE16               0x8042
#define GL_LUMINANCE4_ALPHA4         0x8043
#define GL_LUMINANCE6_ALPHA2         0x8044
#define GL_LUMINANCE8_ALPHA8         0x8045
#define GL_LUMINANCE12_ALPHA4            0x8046
#define GL_LUMINANCE12_ALPHA12           0x8047
#define GL_LUMINANCE16_ALPHA16           0x8048
#define GL_INTENSITY              0x8049
#define GL_INTENSITY4             0x804A
#define GL_INTENSITY8             0x804B
#define GL_INTENSITY12            0x804C
#define GL_INTENSITY16            0x804D
#define GL_R3_G3_B2           0x2A10
#define GL_RGB4               0x804F
#define GL_RGB5               0x8050
#define GL_RGB8               0x8051
#define GL_RGB10              0x8052
#define GL_RGB12              0x8053
#define GL_RGB16              0x8054
#define GL_RGBA2              0x8055
#define GL_RGBA4              0x8056
#define GL_RGB5_A1            0x8057
#define GL_RGBA8              0x8058
#define GL_RGB10_A2           0x8059
#define GL_RGBA12             0x805A
#define GL_RGBA16             0x805B
#define GL_CLIENT_PIXEL_STORE_BIT    0x00000001
#define GL_CLIENT_VERTEX_ARRAY_BIT       0x00000002
#define GL_ALL_CLIENT_ATTRIB_BITS        0xFFFFFFFF
#define GL_CLIENT_ALL_ATTRIB_BITS        0xFFFFFFFF



/*
```

```
 * Miscellaneous
 */

GLAPI void GLAPIENTRY glClearIndex( GLfloat c );

GLAPI void GLAPIENTRY glClearColor( GLclampf red, GLclampf green, GLclampf blue,
GLclampf alpha );

GLAPI void GLAPIENTRY glClear( GLbitfield mask );

GLAPI void GLAPIENTRY glIndexMask( GLuint mask );

GLAPI void GLAPIENTRY glColorMask( GLboolean red, GLboolean green, GLboolean blu
e, GLboolean alpha );

GLAPI void GLAPIENTRY glAlphaFunc( GLenum func, GLclampf ref );

GLAPI void GLAPIENTRY glBlendFunc( GLenum sfactor, GLenum dfactor );

GLAPI void GLAPIENTRY glLogicOp( GLenum opcode );

GLAPI void GLAPIENTRY glCullFace( GLenum mode );

GLAPI void GLAPIENTRY glFrontFace( GLenum mode );

GLAPI void GLAPIENTRY glPointSize( GLfloat size );

GLAPI void GLAPIENTRY glLineWidth( GLfloat width );

GLAPI void GLAPIENTRY glLineStipple( GLint factor, GLushort pattern );

GLAPI void GLAPIENTRY glPolygonMode( GLenum face, GLenum mode );

GLAPI void GLAPIENTRY glPolygonOffset( GLfloat factor, GLfloat units );

GLAPI void GLAPIENTRY glPolygonStipple( const GLubyte *mask );

GLAPI void GLAPIENTRY glGetPolygonStipple( GLubyte *mask );

GLAPI void GLAPIENTRY glEdgeFlag( GLboolean flag );

GLAPI void GLAPIENTRY glEdgeFlagv( const GLboolean *flag );

GLAPI void GLAPIENTRY glScissor( GLint x, GLint y, GLsizei width, GLsizei height);

GLAPI void GLAPIENTRY glClipPlane( GLenum plane, const GLdouble *equation );

GLAPI void GLAPIENTRY glGetClipPlane( GLenum plane, GLdouble *equation );

GLAPI void GLAPIENTRY glDrawBuffer( GLenum mode );

GLAPI void GLAPIENTRY glReadBuffer( GLenum mode );

GLAPI void GLAPIENTRY glEnable( GLenum cap );
GLAPI void GLAPIENTRY glDisable( GLenum cap );
GLAPI GLboolean GLAPIENTRY glIsEnabled( GLenum cap );

GLAPI void GLAPIENTRY glEnableClientState( GLenum cap );  /* 1.1 */
GLAPI void GLAPIENTRY glDisableClientState( GLenum cap );  /* 1.1 */

GLAPI void GLAPIENTRY glGetBooleanv( GLenum pname, GLboolean *params );
GLAPI void GLAPIENTRY glGetDoublev( GLenum pname, GLdouble *params );
GLAPI void GLAPIENTRY glGetFloatv( GLenum pname, GLfloat *params );
GLAPI void GLAPIENTRY glGetIntegerv( GLenum pname, GLint *params );

GLAPI void GLAPIENTRY glPushAttrib( GLbitfield mask );
GLAPI void GLAPIENTRY glPopAttrib( void );

GLAPI void GLAPIENTRY glPushClientAttrib( GLbitfield mask );  /* 1.1 */
GLAPI void GLAPIENTRY glPopClientAttrib( void );  /* 1.1 */

GLAPI GLint GLAPIENTRY glRenderMode( GLenum mode );
GLAPI GLenum GLAPIENTRY glGetError( void );
GLAPI const GLubyte * GLAPIENTRY glGetString( GLenum name );
GLAPI void GLAPIENTRY glFinish( void );
GLAPI void GLAPIENTRY glFlush( void );
GLAPI void GLAPIENTRY glHint( GLenum target, GLenum mode );

/*
 * Depth Buffer
 */

GLAPI void GLAPIENTRY glClearDepth( GLclampd depth );
GLAPI void GLAPIENTRY glDepthFunc( GLenum func );
GLAPI void GLAPIENTRY glDepthMask( GLboolean flag );
GLAPI void GLAPIENTRY glDepthRange( GLclampd near_val, GLclampd far_val );
```

```
/*
 * Accumulation Buffer
 */

GLAPI void GLAPIENTRY glClearAccum( GLfloat red, GLfloat green, GLfloat blue, GLflo
at alpha );

GLAPI void GLAPIENTRY glAccum( GLenum op, GLfloat value );


/*
 * Transformation
 */

GLAPI void GLAPIENTRY glMatrixMode( GLenum mode );

GLAPI void GLAPIENTRY glOrtho( GLdouble left, GLdouble right,
               GLdouble bottom, GLdouble top,
               GLdouble near_val, GLdouble far_val );

GLAPI void GLAPIENTRY glFrustum( GLdouble left, GLdouble right,
                GLdouble bottom, GLdouble top,
                GLdouble near_val, GLdouble far_val );

GLAPI void GLAPIENTRY glViewport( GLint x, GLint y,
                 GLsizei width, GLsizei height );

GLAPI void GLAPIENTRY glPushMatrix( void );

GLAPI void GLAPIENTRY glPopMatrix( void );

GLAPI void GLAPIENTRY glLoadIdentity( void );

GLAPI void GLAPIENTRY glLoadMatrixd( const GLdouble *m );
GLAPI void GLAPIENTRY glLoadMatrixf( const GLfloat *m );

GLAPI void GLAPIENTRY glMultMatrixd( const GLdouble *m );
GLAPI void GLAPIENTRY glMultMatrixf( const GLfloat *m );

GLAPI void GLAPIENTRY glRotated( GLdouble angle,
                GLdouble x, GLdouble y, GLdouble z );
GLAPI void GLAPIENTRY glRotatef( GLfloat angle,
                GLfloat x, GLfloat y, GLfloat z );

GLAPI void GLAPIENTRY glScaled( GLdouble x, GLdouble y, GLdouble z );
GLAPI void GLAPIENTRY glScalef( GLfloat x, GLfloat y, GLfloat z );

GLAPI void GLAPIENTRY glTranslated( GLdouble x, GLdouble y, GLdouble z );
GLAPI void GLAPIENTRY glTranslatef( GLfloat x, GLfloat y, GLfloat z );


/*
```

```
 * Display Lists
 */

GLAPI GLboolean GLAPIENTRY glIsList( GLuint list );

GLAPI void GLAPIENTRY glDeleteLists( GLuint list, GLsizei range );

GLAPI GLuint GLAPIENTRY glGenLists( GLsizei range );

GLAPI void GLAPIENTRY glNewList( GLuint list, GLenum mode );

GLAPI void GLAPIENTRY glEndList( void );

GLAPI void GLAPIENTRY glCallList( GLuint list );

GLAPI void GLAPIENTRY glCallLists( GLsizei n, GLenum type,
                                     const GLvoid *lists );

GLAPI void GLAPIENTRY glListBase( GLuint base );


/*
 * Drawing Functions
 */

GLAPI void GLAPIENTRY glBegin( GLenum mode );

GLAPI void GLAPIENTRY glEnd( void );


GLAPI void GLAPIENTRY glVertex2d( GLdouble x, GLdouble y );
GLAPI void GLAPIENTRY glVertex2f( GLfloat x, GLfloat y );
GLAPI void GLAPIENTRY glVertex2i( GLint x, GLint y );
GLAPI void GLAPIENTRY glVertex2s( GLshort x, GLshort y );

GLAPI void GLAPIENTRY glVertex3d( GLdouble x, GLdouble y, GLdouble z );
GLAPI void GLAPIENTRY glVertex3f( GLfloat x, GLfloat y, GLfloat z );
GLAPI void GLAPIENTRY glVertex3i( GLint x, GLint y, GLint z );
GLAPI void GLAPIENTRY glVertex3s( GLshort x, GLshort y, GLshort z );

GLAPI void GLAPIENTRY glVertex4d( GLdouble x, GLdouble y, GLdouble z, GLdouble
w );
GLAPI void GLAPIENTRY glVertex4f( GLfloat x, GLfloat y, GLfloat z, GLfloat w );
GLAPI void GLAPIENTRY glVertex4i( GLint x, GLint y, GLint z, GLint w );
GLAPI void GLAPIENTRY glVertex4s( GLshort x, GLshort y, GLshort z, GLshort w );

GLAPI void GLAPIENTRY glVertex2dv( const GLdouble *v );
GLAPI void GLAPIENTRY glVertex2fv( const GLfloat *v );
GLAPI void GLAPIENTRY glVertex2iv( const GLint *v );
GLAPI void GLAPIENTRY glVertex2sv( const GLshort *v );

GLAPI void GLAPIENTRY glVertex3dv( const GLdouble *v );
GLAPI void GLAPIENTRY glVertex3fv( const GLfloat *v );
GLAPI void GLAPIENTRY glVertex3iv( const GLint *v );
GLAPI void GLAPIENTRY glVertex3sv( const GLshort *v );

GLAPI void GLAPIENTRY glVertex4dv( const GLdouble *v );
GLAPI void GLAPIENTRY glVertex4fv( const GLfloat *v );
GLAPI void GLAPIENTRY glVertex4iv( const GLint *v );
GLAPI void GLAPIENTRY glVertex4sv( const GLshort *v );


GLAPI void GLAPIENTRY glNormal3b( GLbyte nx, GLbyte ny, GLbyte nz );
GLAPI void GLAPIENTRY glNormal3d( GLdouble nx, GLdouble ny, GLdouble nz );
GLAPI void GLAPIENTRY glNormal3f( GLfloat nx, GLfloat ny, GLfloat nz );
GLAPI void GLAPIENTRY glNormal3i( GLint nx, GLint ny, GLint nz );
GLAPI void GLAPIENTRY glNormal3s( GLshort nx, GLshort ny, GLshort nz );

GLAPI void GLAPIENTRY glNormal3bv( const GLbyte *v );
GLAPI void GLAPIENTRY glNormal3dv( const GLdouble *v );
GLAPI void GLAPIENTRY glNormal3fv( const GLfloat *v );
GLAPI void GLAPIENTRY glNormal3iv( const GLint *v );
GLAPI void GLAPIENTRY glNormal3sv( const GLshort *v );


GLAPI void GLAPIENTRY glIndexd( GLdouble c );
GLAPI void GLAPIENTRY glIndexf( GLfloat c );
GLAPI void GLAPIENTRY glIndexi( GLint c );
GLAPI void GLAPIENTRY glIndexs( GLshort c );
GLAPI void GLAPIENTRY glIndexub( GLubyte c );  /* 1.1 */

GLAPI void GLAPIENTRY glIndexdv( const GLdouble *c );
GLAPI void GLAPIENTRY glIndexfv( const GLfloat *c );
GLAPI void GLAPIENTRY glIndexiv( const GLint *c );
GLAPI void GLAPIENTRY glIndexsv( const GLshort *c );
GLAPI void GLAPIENTRY glIndexubv( const GLubyte *c );  /* 1.1 */

GLAPI void GLAPIENTRY glColor3b( GLbyte red, GLbyte green, GLbyte blue );
GLAPI void GLAPIENTRY glColor3d( GLdouble red, GLdouble green, GLdouble blue );
GLAPI void GLAPIENTRY glColor3f( GLfloat red, GLfloat green, GLfloat blue );
GLAPI void GLAPIENTRY glColor3i( GLint red, GLint green, GLint blue );
GLAPI void GLAPIENTRY glColor3s( GLshort red, GLshort green, GLshort blue );
GLAPI void GLAPIENTRY glColor3ub( GLubyte red, GLubyte green, GLubyte blue );
GLAPI void GLAPIENTRY glColor3ui( GLuint red, GLuint green, GLuint blue );
GLAPI void GLAPIENTRY glColor3us( GLushort red, GLushort green, GLushort blue );

GLAPI void GLAPIENTRY glColor4b( GLbyte red, GLbyte green,
                    GLbyte blue, GLbyte alpha );
GLAPI void GLAPIENTRY glColor4d( GLdouble red, GLdouble green,
                    GLdouble blue, GLdouble alpha );
GLAPI void GLAPIENTRY glColor4f( GLfloat red, GLfloat green,
                    GLfloat blue, GLfloat alpha );
GLAPI void GLAPIENTRY glColor4i( GLint red, GLint green,
                    GLint blue, GLint alpha );
GLAPI void GLAPIENTRY glColor4s( GLshort red, GLshort green,
                    GLshort blue, GLshort alpha );
GLAPI void GLAPIENTRY glColor4ub( GLubyte red, GLubyte green,
                     GLubyte blue, GLubyte alpha );
```

```
GLAPI void GLAPIENTRY glColor4ui( GLuint red, GLuint green,
                                  GLuint blue, GLuint alpha );
GLAPI void GLAPIENTRY glColor4us( GLushort red, GLushort green,
                                  GLushort blue, GLushort alpha );


GLAPI void GLAPIENTRY glColor3bv( const GLbyte *v );
GLAPI void GLAPIENTRY glColor3dv( const GLdouble *v );
GLAPI void GLAPIENTRY glColor3fv( const GLfloat *v );
GLAPI void GLAPIENTRY glColor3iv( const GLint *v );
GLAPI void GLAPIENTRY glColor3sv( const GLshort *v );
GLAPI void GLAPIENTRY glColor3ubv( const GLubyte *v );
GLAPI void GLAPIENTRY glColor3uiv( const GLuint *v );
GLAPI void GLAPIENTRY glColor3usv( const GLushort *v );

GLAPI void GLAPIENTRY glColor4bv( const GLbyte *v );
GLAPI void GLAPIENTRY glColor4dv( const GLdouble *v );
GLAPI void GLAPIENTRY glColor4fv( const GLfloat *v );
GLAPI void GLAPIENTRY glColor4iv( const GLint *v );
GLAPI void GLAPIENTRY glColor4sv( const GLshort *v );
GLAPI void GLAPIENTRY glColor4ubv( const GLubyte *v );
GLAPI void GLAPIENTRY glColor4uiv( const GLuint *v );
GLAPI void GLAPIENTRY glColor4usv( const GLushort *v );


GLAPI void GLAPIENTRY glTexCoord1d( GLdouble s );
GLAPI void GLAPIENTRY glTexCoord1f( GLfloat s );
GLAPI void GLAPIENTRY glTexCoord1i( GLint s );
GLAPI void GLAPIENTRY glTexCoord1s( GLshort s );

GLAPI void GLAPIENTRY glTexCoord2d( GLdouble s, GLdouble t );
GLAPI void GLAPIENTRY glTexCoord2f( GLfloat s, GLfloat t );
GLAPI void GLAPIENTRY glTexCoord2i( GLint s, GLint t );
GLAPI void GLAPIENTRY glTexCoord2s( GLshort s, GLshort t );

GLAPI void GLAPIENTRY glTexCoord3d( GLdouble s, GLdouble t, GLdouble r );
GLAPI void GLAPIENTRY glTexCoord3f( GLfloat s, GLfloat t, GLfloat r );
GLAPI void GLAPIENTRY glTexCoord3i( GLint s, GLint t, GLint r );
GLAPI void GLAPIENTRY glTexCoord3s( GLshort s, GLshort t, GLshort r );

GLAPI void GLAPIENTRY glTexCoord4d( GLdouble s, GLdouble t, GLdouble r, GLdoubl
e q );
GLAPI void GLAPIENTRY glTexCoord4f( GLfloat s, GLfloat t, GLfloat r, GLfloat q );
GLAPI void GLAPIENTRY glTexCoord4i( GLint s, GLint t, GLint r, GLint q );
GLAPI void GLAPIENTRY glTexCoord4s( GLshort s, GLshort t, GLshort r, GLshort q );

GLAPI void GLAPIENTRY glTexCoord1dv( const GLdouble *v );
GLAPI void GLAPIENTRY glTexCoord1fv( const GLfloat *v );
GLAPI void GLAPIENTRY glTexCoord1iv( const GLint *v );
GLAPI void GLAPIENTRY glTexCoord1sv( const GLshort *v );

GLAPI void GLAPIENTRY glTexCoord2dv( const GLdouble *v );
GLAPI void GLAPIENTRY glTexCoord2fv( const GLfloat *v );
GLAPI void GLAPIENTRY glTexCoord2iv( const GLint *v );
```

```c
GLAPI void GLAPIENTRY glTexCoord2sv( const GLshort *v );

GLAPI void GLAPIENTRY glTexCoord3dv( const GLdouble *v );
GLAPI void GLAPIENTRY glTexCoord3fv( const GLfloat *v );
GLAPI void GLAPIENTRY glTexCoord3iv( const GLint *v );
GLAPI void GLAPIENTRY glTexCoord3sv( const GLshort *v );

GLAPI void GLAPIENTRY glTexCoord4dv( const GLdouble *v );
GLAPI void GLAPIENTRY glTexCoord4fv( const GLfloat *v );
GLAPI void GLAPIENTRY glTexCoord4iv( const GLint *v );
GLAPI void GLAPIENTRY glTexCoord4sv( const GLshort *v );


GLAPI void GLAPIENTRY glRasterPos2d( GLdouble x, GLdouble y );
GLAPI void GLAPIENTRY glRasterPos2f( GLfloat x, GLfloat y );
GLAPI void GLAPIENTRY glRasterPos2i( GLint x, GLint y );
GLAPI void GLAPIENTRY glRasterPos2s( GLshort x, GLshort y );

GLAPI void GLAPIENTRY glRasterPos3d( GLdouble x, GLdouble y, GLdouble z );
GLAPI void GLAPIENTRY glRasterPos3f( GLfloat x, GLfloat y, GLfloat z );
GLAPI void GLAPIENTRY glRasterPos3i( GLint x, GLint y, GLint z );
GLAPI void GLAPIENTRY glRasterPos3s( GLshort x, GLshort y, GLshort z );

GLAPI void GLAPIENTRY glRasterPos4d( GLdouble x, GLdouble y, GLdouble z, GLdouble w );
GLAPI void GLAPIENTRY glRasterPos4f( GLfloat x, GLfloat y, GLfloat z, GLfloat w );
GLAPI void GLAPIENTRY glRasterPos4i( GLint x, GLint y, GLint z, GLint w );
GLAPI void GLAPIENTRY glRasterPos4s( GLshort x, GLshort y, GLshort z, GLshort w );

GLAPI void GLAPIENTRY glRasterPos2dv( const GLdouble *v );
GLAPI void GLAPIENTRY glRasterPos2fv( const GLfloat *v );
GLAPI void GLAPIENTRY glRasterPos2iv( const GLint *v );
GLAPI void GLAPIENTRY glRasterPos2sv( const GLshort *v );

GLAPI void GLAPIENTRY glRasterPos3dv( const GLdouble *v );
GLAPI void GLAPIENTRY glRasterPos3fv( const GLfloat *v );
GLAPI void GLAPIENTRY glRasterPos3iv( const GLint *v );
GLAPI void GLAPIENTRY glRasterPos3sv( const GLshort *v );

GLAPI void GLAPIENTRY glRasterPos4dv( const GLdouble *v );
GLAPI void GLAPIENTRY glRasterPos4fv( const GLfloat *v );
GLAPI void GLAPIENTRY glRasterPos4iv( const GLint *v );
GLAPI void GLAPIENTRY glRasterPos4sv( const GLshort *v );


GLAPI void GLAPIENTRY glRectd( GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2 );
GLAPI void GLAPIENTRY glRectf( GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2 );
GLAPI void GLAPIENTRY glRecti( GLint x1, GLint y1, GLint x2, GLint y2 );
GLAPI void GLAPIENTRY glRects( GLshort x1, GLshort y1, GLshort x2, GLshort y2 );


GLAPI void GLAPIENTRY glRectdv( const GLdouble *v1, const GLdouble *v2 );
GLAPI void GLAPIENTRY glRectfv( const GLfloat *v1, const GLfloat *v2 );
GLAPI void GLAPIENTRY glRectiv( const GLint *v1, const GLint *v2 );
GLAPI void GLAPIENTRY glRectsv( const GLshort *v1, const GLshort *v2 );


/*
 * Vertex Arrays  (1.1)
 */

GLAPI void GLAPIENTRY glVertexPointer( GLint size, GLenum type,
                                       GLsizei stride, const GLvoid *ptr );

GLAPI void GLAPIENTRY glNormalPointer( GLenum type, GLsizei stride,
                                       const GLvoid *ptr );

GLAPI void GLAPIENTRY glColorPointer( GLint size, GLenum type,
                                      GLsizei stride, const GLvoid *ptr );

GLAPI void GLAPIENTRY glIndexPointer( GLenum type, GLsizei stride,
                                      const GLvoid *ptr );

GLAPI void GLAPIENTRY glTexCoordPointer( GLint size, GLenum type,
                                         GLsizei stride, const GLvoid *ptr );

GLAPI void GLAPIENTRY glEdgeFlagPointer( GLsizei stride, const GLvoid *ptr );

GLAPI void GLAPIENTRY glGetPointerv( GLenum pname, GLvoid **params );

GLAPI void GLAPIENTRY glArrayElement( GLint i );

GLAPI void GLAPIENTRY glDrawArrays( GLenum mode, GLint first, GLsizei count );

GLAPI void GLAPIENTRY glDrawElements( GLenum mode, GLsizei count,
                                      GLenum type, const GLvoid *indices );

GLAPI void GLAPIENTRY glInterleavedArrays( GLenum format, GLsizei stride,
                                           const GLvoid *pointer );

/*
 * Lighting
 */

GLAPI void GLAPIENTRY glShadeModel( GLenum mode );

GLAPI void GLAPIENTRY glLightf( GLenum light, GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glLighti( GLenum light, GLenum pname, GLint param );
GLAPI void GLAPIENTRY glLightfv( GLenum light, GLenum pname,
                                 const GLfloat *params );
GLAPI void GLAPIENTRY glLightiv( GLenum light, GLenum pname,
                                 const GLint *params );

GLAPI void GLAPIENTRY glGetLightfv( GLenum light, GLenum pname,
                                    GLfloat *params );
GLAPI void GLAPIENTRY glGetLightiv( GLenum light, GLenum pname,
                                    GLint *params );
```

```c
GLAPI void GLAPIENTRY glLightModelf( GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glLightModeli( GLenum pname, GLint param );
GLAPI void GLAPIENTRY glLightModelfv( GLenum pname, const GLfloat *params );
GLAPI void GLAPIENTRY glLightModeliv( GLenum pname, const GLint *params );

GLAPI void GLAPIENTRY glMaterialf( GLenum face, GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glMateriali( GLenum face, GLenum pname, GLint param );
GLAPI void GLAPIENTRY glMaterialfv( GLenum face, GLenum pname, const GLfloat *params );
GLAPI void GLAPIENTRY glMaterialiv( GLenum face, GLenum pname, const GLint *params );

GLAPI void GLAPIENTRY glGetMaterialfv( GLenum face, GLenum pname, GLfloat *params );
GLAPI void GLAPIENTRY glGetMaterialiv( GLenum face, GLenum pname, GLint *params );

GLAPI void GLAPIENTRY glColorMaterial( GLenum face, GLenum mode );


/*
 * Raster functions
 */

GLAPI void GLAPIENTRY glPixelZoom( GLfloat xfactor, GLfloat yfactor );

GLAPI void GLAPIENTRY glPixelStoref( GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glPixelStorei( GLenum pname, GLint param );

GLAPI void GLAPIENTRY glPixelTransferf( GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glPixelTransferi( GLenum pname, GLint param );

GLAPI void GLAPIENTRY glPixelMapfv( GLenum map, GLsizei mapsize,
                                    const GLfloat *values );
GLAPI void GLAPIENTRY glPixelMapuiv( GLenum map, GLsizei mapsize,
                                     const GLuint *values );
GLAPI void GLAPIENTRY glPixelMapusv( GLenum map, GLsizei mapsize,
                                     const GLushort *values );

GLAPI void GLAPIENTRY glGetPixelMapfv( GLenum map, GLfloat *values );
GLAPI void GLAPIENTRY glGetPixelMapuiv( GLenum map, GLuint *values );
GLAPI void GLAPIENTRY glGetPixelMapusv( GLenum map, GLushort *values );

GLAPI void GLAPIENTRY glBitmap( GLsizei width, GLsizei height,
                                GLfloat xorig, GLfloat yorig,
                                GLfloat xmove, GLfloat ymove,
                                const GLubyte *bitmap );

GLAPI void GLAPIENTRY glReadPixels( GLint x, GLint y,
                                    GLsizei width, GLsizei height,
                                    GLenum format, GLenum type,
                                    GLvoid *pixels );
```

```
GLAPI void GLAPIENTRY glDrawPixels( GLsizei width, GLsizei height,
                                    GLenum format, GLenum type,
                                    const GLvoid *pixels );

GLAPI void GLAPIENTRY glCopyPixels( GLint x, GLint y,
                                    GLsizei width, GLsizei height,
                                    GLenum type );

/*
 * Stenciling
 */

GLAPI void GLAPIENTRY glStencilFunc( GLenum func, GLint ref, GLuint mask );

GLAPI void GLAPIENTRY glStencilMask( GLuint mask );

GLAPI void GLAPIENTRY glStencilOp( GLenum fail, GLenum zfail, GLenum zpass );

GLAPI void GLAPIENTRY glClearStencil( GLint s );



/*
 * Texture mapping
 */

GLAPI void GLAPIENTRY glTexGend( GLenum coord, GLenum pname, GLdouble param
);
GLAPI void GLAPIENTRY glTexGenf( GLenum coord, GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glTexGeni( GLenum coord, GLenum pname, GLint param );

GLAPI void GLAPIENTRY glTexGendv( GLenum coord, GLenum pname, const GLdouble
*params );
GLAPI void GLAPIENTRY glTexGenfv( GLenum coord, GLenum pname, const GLfloat *p
arams );
GLAPI void GLAPIENTRY glTexGeniv( GLenum coord, GLenum pname, const GLint *par
ams );

GLAPI void GLAPIENTRY glGetTexGendv( GLenum coord, GLenum pname, GLdouble *p
arams );
GLAPI void GLAPIENTRY glGetTexGenfv( GLenum coord, GLenum pname, GLfloat *par
ams );
GLAPI void GLAPIENTRY glGetTexGeniv( GLenum coord, GLenum pname, GLint *para
ms );


GLAPI void GLAPIENTRY glTexEnvf( GLenum target, GLenum pname, GLfloat param );
GLAPI void GLAPIENTRY glTexEnvi( GLenum target, GLenum pname, GLint param );

GLAPI void GLAPIENTRY glTexEnvfv( GLenum target, GLenum pname, const GLfloat *p
arams );
GLAPI void GLAPIENTRY glTexEnviv( GLenum target, GLenum pname, const GLint *par
ams );
```

```
GLAPI void GLAPIENTRY glGetTexEnvfv( GLenum target, GLenum pname, GLfloat *par
ams );
GLAPI void GLAPIENTRY glGetTexEnviv( GLenum target, GLenum pname, GLint *param
s );


GLAPI void GLAPIENTRY glTexParameterf( GLenum target, GLenum pname, GLfloat para
m );
GLAPI void GLAPIENTRY glTexParameteri( GLenum target, GLenum pname, GLint para
m );

GLAPI void GLAPIENTRY glTexParameterfv( GLenum target, GLenum pname,
                    const GLfloat *params );
GLAPI void GLAPIENTRY glTexParameteriv( GLenum target, GLenum pname,
                    const GLint *params );

GLAPI void GLAPIENTRY glGetTexParameterfv( GLenum target,
                    GLenum pname, GLfloat *params);
GLAPI void GLAPIENTRY glGetTexParameteriv( GLenum target,
                    GLenum pname, GLint *params );

GLAPI void GLAPIENTRY glGetTexLevelParameterfv( GLenum target, GLint level,
                      GLenum pname, GLfloat *params );
GLAPI void GLAPIENTRY glGetTexLevelParameteriv( GLenum target, GLint level,
                      GLenum pname, GLint *params );


GLAPI void GLAPIENTRY glTexImage1D( GLenum target, GLint level,
                 GLint internalFormat,
                 GLsizei width, GLint border,
                 GLenum format, GLenum type,
                 const GLvoid *pixels );

GLAPI void GLAPIENTRY glTexImage2D( GLenum target, GLint level,
                 GLint internalFormat,
                 GLsizei width, GLsizei height,
                 GLint border, GLenum format, GLenum type,
                 const GLvoid *pixels );

GLAPI void GLAPIENTRY glGetTexImage( GLenum target, GLint level,
                 GLenum format, GLenum type,
                 GLvoid *pixels );


/* 1.1 functions */

GLAPI void GLAPIENTRY glGenTextures( GLsizei n, GLuint *textures );

GLAPI void GLAPIENTRY glDeleteTextures( GLsizei n, const GLuint *textures);

GLAPI void GLAPIENTRY glBindTexture( GLenum target, GLuint texture );

GLAPI void GLAPIENTRY glPrioritizeTextures( GLsizei n,
                    const GLuint *textures,
                                            const GLclampf *priorities );

GLAPI GLboolean GLAPIENTRY glAreTexturesResident( GLsizei n,
                                                  const GLuint *textures,
                                                  GLboolean *residences );

GLAPI GLboolean GLAPIENTRY glIsTexture( GLuint texture );


GLAPI void GLAPIENTRY glTexSubImage1D( GLenum target, GLint level,
                                       GLint xoffset,
                                       GLsizei width, GLenum format,
                                       GLenum type, const GLvoid *pixels );


GLAPI void GLAPIENTRY glTexSubImage2D( GLenum target, GLint level,
                                       GLint xoffset, GLint yoffset,
                                       GLsizei width, GLsizei height,
                                       GLenum format, GLenum type,
                                       const GLvoid *pixels );


GLAPI void GLAPIENTRY glCopyTexImage1D( GLenum target, GLint level,
                                        GLenum internalformat,
                                        GLint x, GLint y,
                                        GLsizei width, GLint border );


GLAPI void GLAPIENTRY glCopyTexImage2D( GLenum target, GLint level,
                                        GLenum internalformat,
                                        GLint x, GLint y,
                                        GLsizei width, GLsizei height,
                                        GLint border );


GLAPI void GLAPIENTRY glCopyTexSubImage1D( GLenum target, GLint level,
                                           GLint xoffset, GLint x, GLint y,
                                           GLsizei width );


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


/*
 * Evaluators
 */

GLAPI void GLAPIENTRY glMap1d( GLenum target, GLdouble u1, GLdouble u2,
                               GLint stride,
                               GLint order, const GLdouble *points );
GLAPI void GLAPIENTRY glMap1f( GLenum target, GLfloat u1, GLfloat u2,
               GLint stride,
               GLint order, const GLfloat *points );

GLAPI void GLAPIENTRY glMap2d( GLenum target,
          GLdouble u1, GLdouble u2, GLint ustride, GLint uorder,
          GLdouble v1, GLdouble v2, GLint vstride, GLint vorder,
          const GLdouble *points );
GLAPI void GLAPIENTRY glMap2f( GLenum target,
          GLfloat u1, GLfloat u2, GLint ustride, GLint uorder,
          GLfloat v1, GLfloat v2, GLint vstride, GLint vorder,
          const GLfloat *points );

GLAPI void GLAPIENTRY glGetMapdv( GLenum target, GLenum query, GLdouble *v );
GLAPI void GLAPIENTRY glGetMapfv( GLenum target, GLenum query, GLfloat *v );
GLAPI void GLAPIENTRY glGetMapiv( GLenum target, GLenum query, GLint *v );

GLAPI void GLAPIENTRY glEvalCoord1d( GLdouble u );
GLAPI void GLAPIENTRY glEvalCoord1f( GLfloat u );

GLAPI void GLAPIENTRY glEvalCoord1dv( const GLdouble *u );
GLAPI void GLAPIENTRY glEvalCoord1fv( const GLfloat *u );

GLAPI void GLAPIENTRY glEvalCoord2d( GLdouble u, GLdouble v );
GLAPI void GLAPIENTRY glEvalCoord2f( GLfloat u, GLfloat v );

GLAPI void GLAPIENTRY glEvalCoord2dv( const GLdouble *u );
GLAPI void GLAPIENTRY glEvalCoord2fv( const GLfloat *u );

GLAPI void GLAPIENTRY glMapGrid1d( GLint un, GLdouble u1, GLdouble u2 );
GLAPI void GLAPIENTRY glMapGrid1f( GLint un, GLfloat u1, GLfloat u2 );

GLAPI void GLAPIENTRY glMapGrid2d( GLint un, GLdouble u1, GLdouble u2,
                 GLint vn, GLdouble v1, GLdouble v2 );
GLAPI void GLAPIENTRY glMapGrid2f( GLint un, GLfloat u1, GLfloat u2,
                 GLint vn, GLfloat v1, GLfloat v2 );

GLAPI void GLAPIENTRY glEvalPoint1( GLint i );

GLAPI void GLAPIENTRY glEvalPoint2( GLint i, GLint j );

GLAPI void GLAPIENTRY glEvalMesh1( GLenum mode, GLint i1, GLint i2 );

GLAPI void GLAPIENTRY glEvalMesh2( GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2 );


/*
 * Fog
 */

GLAPI void GLAPIENTRY glFogf( GLenum pname, GLfloat param );

GLAPI void GLAPIENTRY glFogi( GLenum pname, GLint param );
GLAPI void GLAPIENTRY glFogfv( GLenum pname, const GLfloat *params );

GLAPI void GLAPIENTRY glFogiv( GLenum pname, const GLint *params );


/*
 * Selection and Feedback
 */

GLAPI void GLAPIENTRY glFeedbackBuffer( GLsizei size, GLenum type, GLfloat *buffer
);

GLAPI void GLAPIENTRY glPassThrough( GLfloat token );

GLAPI void GLAPIENTRY glSelectBuffer( GLsizei size, GLuint *buffer );

GLAPI void GLAPIENTRY glInitNames( void );

GLAPI void GLAPIENTRY glLoadName( GLuint name );

GLAPI void GLAPIENTRY glPushName( GLuint name );

GLAPI void GLAPIENTRY glPopName( void );



/*
 * OpenGL 1.2
 */

#define GL_RESCALE_NORMAL           0x803A
#define GL_CLAMP_TO_EDGE        0x812F
#define GL_MAX_ELEMENTS_VERTICES        0x80E8
#define GL_MAX_ELEMENTS_INDICES         0x80E9
#define GL_BGR                0x80E0
#define GL_BGRA               0x80E1
#define GL_UNSIGNED_BYTE_3_3_2          0x8032
#define GL_UNSIGNED_BYTE_2_3_3_REV      0x8362
#define GL_UNSIGNED_SHORT_5_6_5         0x8363
#define GL_UNSIGNED_SHORT_5_6_5_REV     0x8364
#define GL_UNSIGNED_SHORT_4_4_4_4   0x8033
#define GL_UNSIGNED_SHORT_4_4_4_4_REV       0x8365
#define GL_UNSIGNED_SHORT_5_5_5_1   0x8034
#define GL_UNSIGNED_SHORT_1_5_5_5_REV       0x8366
#define GL_UNSIGNED_INT_8_8_8_8         0x8035
#define GL_UNSIGNED_INT_8_8_8_8_REV     0x8367
#define GL_UNSIGNED_INT_10_10_10_2  0x8036
#define GL_UNSIGNED_INT_2_10_10_10_REV      0x8368
#define GL_LIGHT_MODEL_COLOR_CONTROL        0x81F8
#define GL_SINGLE_COLOR             0x81F9
#define GL_SEPARATE_SPECULAR_COLOR      0x81FA
#define GL_TEXTURE_MIN_LOD          0x813A
#define GL_TEXTURE_MAX_LOD          0x813B
#define GL_TEXTURE_BASE_LEVEL           0x813C
```

```
#define GL_TEXTURE_MAX_LEVEL                 0x813D
#define GL_SMOOTH_POINT_SIZE_RANGE           0x0B12
#define GL_SMOOTH_POINT_SIZE_GRANULARITY     0x0B13
#define GL_SMOOTH_LINE_WIDTH_RANGE           0x0B22
#define GL_SMOOTH_LINE_WIDTH_GRANULARITY  0x0B23
#define GL_ALIASED_POINT_SIZE_RANGE          0x846D
#define GL_ALIASED_LINE_WIDTH_RANGE          0x846E
#define GL_PACK_SKIP_IMAGES              0x806B
#define GL_PACK_IMAGE_HEIGHT             0x806C
#define GL_UNPACK_SKIP_IMAGES                0x806D
#define GL_UNPACK_IMAGE_HEIGHT               0x806E
#define GL_TEXTURE_3D                0x806F
#define GL_PROXY_TEXTURE_3D              0x8070
#define GL_TEXTURE_DEPTH             0x8071
#define GL_TEXTURE_WRAP_R                0x8072
#define GL_MAX_3D_TEXTURE_SIZE               0x8073
#define GL_TEXTURE_BINDING_3D                0x806A

GLAPI void GLAPIENTRY glDrawRangeElements( GLenum mode, GLuint start,
	GLuint end, GLsizei count, GLenum type, const GLvoid *indices );

GLAPI void GLAPIENTRY glTexImage3D( GLenum target, GLint level,
                                   GLint internalFormat,
                                   GLsizei width, GLsizei height,
                                   GLsizei depth, GLint border,
                                   GLenum format, GLenum type,
                                   const GLvoid *pixels );

GLAPI void GLAPIENTRY glTexSubImage3D( GLenum target, GLint level,
                                      GLint xoffset, GLint yoffset,
                                      GLint zoffset, GLsizei width,
                                      GLsizei height, GLsizei depth,
                                      GLenum format,
                                      GLenum type, const GLvoid *pixels);

GLAPI void GLAPIENTRY glCopyTexSubImage3D( GLenum target, GLint level,
                                          GLint xoffset, GLint yoffset,
                                          GLint zoffset, GLint x,
                                          GLint y, GLsizei width,
                                          GLsizei height );

typedef void (APIENTRYP PFNGLDRAWRANGEELEMENTSPROC) (GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, const GLvoid *indices);
typedef void (APIENTRYP PFNGLTEXIMAGE3DPROC) (GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, const GLvoid *pixels);
typedef void (APIENTRYP PFNGLTEXSUBIMAGE3DPROC) (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, const GLvoid *pixels);
typedef void (APIENTRYP PFNGLCOPYTEXSUBIMAGE3DPROC) (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height);
```

```
/*
 * GL_ARB_imaging
 */

#define GL_CONSTANT_COLOR                        0x8001
#define GL_ONE_MINUS_CONSTANT_COLOR              0x8002
#define GL_CONSTANT_ALPHA                        0x8003
#define GL_ONE_MINUS_CONSTANT_ALPHA              0x8004
#define GL_COLOR_TABLE                           0x80D0
#define GL_POST_CONVOLUTION_COLOR_TABLE          0x80D1
#define GL_POST_COLOR_MATRIX_COLOR_TABLE         0x80D2
#define GL_PROXY_COLOR_TABLE                     0x80D3
#define GL_PROXY_POST_CONVOLUTION_COLOR_TABLE    0x80D4
#define GL_PROXY_POST_COLOR_MATRIX_COLOR_TABLE   0x80D5
#define GL_COLOR_TABLE_SCALE                     0x80D6
#define GL_COLOR_TABLE_BIAS                      0x80D7
#define GL_COLOR_TABLE_FORMAT                    0x80D8
#define GL_COLOR_TABLE_WIDTH                     0x80D9
#define GL_COLOR_TABLE_RED_SIZE                  0x80DA
#define GL_COLOR_TABLE_GREEN_SIZE                0x80DB
#define GL_COLOR_TABLE_BLUE_SIZE                 0x80DC
#define GL_COLOR_TABLE_ALPHA_SIZE                0x80DD
#define GL_COLOR_TABLE_LUMINANCE_SIZE            0x80DE
#define GL_COLOR_TABLE_INTENSITY_SIZE            0x80DF
#define GL_CONVOLUTION_1D                        0x8010
#define GL_CONVOLUTION_2D                        0x8011
#define GL_SEPARABLE_2D                          0x8012
#define GL_CONVOLUTION_BORDER_MODE               0x8013
#define GL_CONVOLUTION_FILTER_SCALE              0x8014
#define GL_CONVOLUTION_FILTER_BIAS               0x8015
#define GL_REDUCE                                0x8016
#define GL_CONVOLUTION_FORMAT                    0x8017
#define GL_CONVOLUTION_WIDTH                     0x8018
#define GL_CONVOLUTION_HEIGHT                    0x8019
#define GL_MAX_CONVOLUTION_WIDTH                 0x801A
#define GL_MAX_CONVOLUTION_HEIGHT                0x801B
#define GL_POST_CONVOLUTION_RED_SCALE            0x801C
#define GL_POST_CONVOLUTION_GREEN_SCALE          0x801D
#define GL_POST_CONVOLUTION_BLUE_SCALE           0x801E
#define GL_POST_CONVOLUTION_ALPHA_SCALE          0x801F
#define GL_POST_CONVOLUTION_RED_BIAS             0x8020
#define GL_POST_CONVOLUTION_GREEN_BIAS           0x8021
#define GL_POST_CONVOLUTION_BLUE_BIAS            0x8022
#define GL_POST_CONVOLUTION_ALPHA_BIAS           0x8023
#define GL_CONSTANT_BORDER                       0x8151
#define GL_REPLICATE_BORDER                      0x8153
#define GL_CONVOLUTION_BORDER_COLOR              0x8154
#define GL_COLOR_MATRIX                          0x80B1
#define GL_COLOR_MATRIX_STACK_DEPTH              0x80B2
#define GL_MAX_COLOR_MATRIX_STACK_DEPTH          0x80B3
#define GL_POST_COLOR_MATRIX_RED_SCALE           0x80B4
#define GL_POST_COLOR_MATRIX_GREEN_SCALE         0x80B5
#define GL_POST_COLOR_MATRIX_BLUE_SCALE          0x80B6
#define GL_POST_COLOR_MATRIX_ALPHA_SCALE         0x80B7
```

```
#define GL_POST_COLOR_MATRIX_RED_BIAS           0x80B8
#define GL_POST_COLOR_MATRIX_GREEN_BIAS              0x80B9
#define GL_POST_COLOR_MATRIX_BLUE_BIAS          0x80BA
#define GL_POST_COLOR_MATRIX_ALPHA_BIAS              0x80BB
#define GL_HISTOGRAM                       0x8024
#define GL_PROXY_HISTOGRAM                      0x8025
#define GL_HISTOGRAM_WIDTH                      0x8026
#define GL_HISTOGRAM_FORMAT                     0x8027
#define GL_HISTOGRAM_RED_SIZE                        0x8028
#define GL_HISTOGRAM_GREEN_SIZE                      0x8029
#define GL_HISTOGRAM_BLUE_SIZE                       0x802A
#define GL_HISTOGRAM_ALPHA_SIZE                      0x802B
#define GL_HISTOGRAM_LUMINANCE_SIZE                       0x802C
#define GL_HISTOGRAM_SINK                  0x802D
#define GL_MINMAX                     0x802E
#define GL_MINMAX_FORMAT                   0x802F
#define GL_MINMAX_SINK                          0x8030
#define GL_TABLE_TOO_LARGE                      0x8031
#define GL_BLEND_EQUATION                  0x8009
#define GL_MIN                        0x8007
#define GL_MAX                        0x8008
#define GL_FUNC_ADD                        0x8006
#define GL_FUNC_SUBTRACT                   0x800A
#define GL_FUNC_REVERSE_SUBTRACT                     0x800B
#define GL_BLEND_COLOR                          0x8005


GLAPI void GLAPIENTRY glColorTable( GLenum target, GLenum internalformat,
                                    GLsizei width, GLenum format,
                                    GLenum type, const GLvoid *table );

GLAPI void GLAPIENTRY glColorSubTable( GLenum target,
                                       GLsizei start, GLsizei count,
                                       GLenum format, GLenum type,
                                       const GLvoid *data );

GLAPI void GLAPIENTRY glColorTableParameteriv(GLenum target, GLenum pname,
                                              const GLint *params);

GLAPI void GLAPIENTRY glColorTableParameterfv(GLenum target, GLenum pname,
                                              const GLfloat *params);

GLAPI void GLAPIENTRY glCopyColorSubTable( GLenum target, GLsizei start,
                                           GLint x, GLint y, GLsizei width );

GLAPI void GLAPIENTRY glCopyColorTable( GLenum target, GLenum internalformat,
                                        GLint x, GLint y, GLsizei width );

GLAPI void GLAPIENTRY glGetColorTable( GLenum target, GLenum format,
                                       GLenum type, GLvoid *table );

GLAPI void GLAPIENTRY glGetColorTableParameterfv( GLenum target, GLenum pname,
                                                  GLfloat *params );
```

```
GLAPI void GLAPIENTRY glGetColorTableParameteriv( GLenum target, GLenum pname,
                       GLint *params );

GLAPI void GLAPIENTRY glBlendEquation( GLenum mode );

GLAPI void GLAPIENTRY glBlendColor( GLclampf red, GLclampf green,
                 GLclampf blue, GLclampf alpha );

GLAPI void GLAPIENTRY glHistogram( GLenum target, GLsizei width,
                GLenum internalformat, GLboolean sink );

GLAPI void GLAPIENTRY glResetHistogram( GLenum target );

GLAPI void GLAPIENTRY glGetHistogram( GLenum target, GLboolean reset,
                  GLenum format, GLenum type,
                  GLvoid *values );

GLAPI void GLAPIENTRY glGetHistogramParameterfv( GLenum target, GLenum pname,
                       GLfloat *params );

GLAPI void GLAPIENTRY glGetHistogramParameteriv( GLenum target, GLenum pname,
                       GLint *params );

GLAPI void GLAPIENTRY glMinmax( GLenum target, GLenum internalformat,
               GLboolean sink );

GLAPI void GLAPIENTRY glResetMinmax( GLenum target );

GLAPI void GLAPIENTRY glGetMinmax( GLenum target, GLboolean reset,
                GLenum format, GLenum types,
                GLvoid *values );

GLAPI void GLAPIENTRY glGetMinmaxParameterfv( GLenum target, GLenum pname,
                     GLfloat *params );

GLAPI void GLAPIENTRY glGetMinmaxParameteriv( GLenum target, GLenum pname,
                     GLint *params );

GLAPI void GLAPIENTRY glConvolutionFilter1D( GLenum target,
    GLenum internalformat, GLsizei width, GLenum format, GLenum type,
    const GLvoid *image );

GLAPI void GLAPIENTRY glConvolutionFilter2D( GLenum target,
    GLenum internalformat, GLsizei width, GLsizei height, GLenum format,
    GLenum type, const GLvoid *image );

GLAPI void GLAPIENTRY glConvolutionParameterf( GLenum target, GLenum pname,
    GLfloat params );

GLAPI void GLAPIENTRY glConvolutionParameterfv( GLenum target, GLenum pname,
    const GLfloat *params );

GLAPI void GLAPIENTRY glConvolutionParameteri( GLenum target, GLenum pname,
    GLint params );
```

```
GLAPI void GLAPIENTRY glConvolutionParameteriv( GLenum target, GLenum pname,
        const GLint *params );

GLAPI void GLAPIENTRY glCopyConvolutionFilter1D( GLenum target,
        GLenum internalformat, GLint x, GLint y, GLsizei width );

GLAPI void GLAPIENTRY glCopyConvolutionFilter2D( GLenum target,
        GLenum internalformat, GLint x, GLint y, GLsizei width,
        GLsizei height);

GLAPI void GLAPIENTRY glGetConvolutionFilter( GLenum target, GLenum format,
        GLenum type, GLvoid *image );

GLAPI void GLAPIENTRY glGetConvolutionParameterfv( GLenum target, GLenum pnam
e,
        GLfloat *params );

GLAPI void GLAPIENTRY glGetConvolutionParameteriv( GLenum target, GLenum pname,
        GLint *params );

GLAPI void GLAPIENTRY glSeparableFilter2D( GLenum target,
        GLenum internalformat, GLsizei width, GLsizei height, GLenum format,
        GLenum type, const GLvoid *row, const GLvoid *column );

GLAPI void GLAPIENTRY glGetSeparableFilter( GLenum target, GLenum format,
        GLenum type, GLvoid *row, GLvoid *column, GLvoid *span );



/*
 * OpenGL 1.3
 */

/* multitexture */
#define GL_TEXTURE0                         0x84C0
#define GL_TEXTURE1                         0x84C1
#define GL_TEXTURE2                         0x84C2
#define GL_TEXTURE3                         0x84C3
#define GL_TEXTURE4                         0x84C4
#define GL_TEXTURE5                         0x84C5
#define GL_TEXTURE6                         0x84C6
#define GL_TEXTURE7                         0x84C7
#define GL_TEXTURE8                         0x84C8
#define GL_TEXTURE9                         0x84C9
#define GL_TEXTURE10                        0x84CA
#define GL_TEXTURE11                        0x84CB
#define GL_TEXTURE12                        0x84CC
#define GL_TEXTURE13                        0x84CD
#define GL_TEXTURE14                        0x84CE
#define GL_TEXTURE15                        0x84CF
#define GL_TEXTURE16                        0x84D0
#define GL_TEXTURE17                        0x84D1
```

```
#define GL_TEXTURE18                0x84D2
#define GL_TEXTURE19                0x84D3
#define GL_TEXTURE20                0x84D4
#define GL_TEXTURE21                0x84D5
#define GL_TEXTURE22                0x84D6
#define GL_TEXTURE23                0x84D7
#define GL_TEXTURE24                0x84D8
#define GL_TEXTURE25                0x84D9
#define GL_TEXTURE26                0x84DA
#define GL_TEXTURE27                0x84DB
#define GL_TEXTURE28                0x84DC
#define GL_TEXTURE29                0x84DD
#define GL_TEXTURE30                0x84DE
#define GL_TEXTURE31                0x84DF
#define GL_ACTIVE_TEXTURE           0x84E0
#define GL_CLIENT_ACTIVE_TEXTURE        0x84E1
#define GL_MAX_TEXTURE_UNITS            0x84E2
/* texture_cube_map */
#define GL_NORMAL_MAP                   0x8511
#define GL_REFLECTION_MAP           0x8512
#define GL_TEXTURE_CUBE_MAP             0x8513
#define GL_TEXTURE_BINDING_CUBE_MAP             0x8514
#define GL_TEXTURE_CUBE_MAP_POSITIVE_X          0x8515
#define GL_TEXTURE_CUBE_MAP_NEGATIVE_X              0x8516
#define GL_TEXTURE_CUBE_MAP_POSITIVE_Y          0x8517
#define GL_TEXTURE_CUBE_MAP_NEGATIVE_Y              0x8518
#define GL_TEXTURE_CUBE_MAP_POSITIVE_Z          0x8519
#define GL_TEXTURE_CUBE_MAP_NEGATIVE_Z              0x851A
#define GL_PROXY_TEXTURE_CUBE_MAP           0x851B
#define GL_MAX_CUBE_MAP_TEXTURE_SIZE            0x851C
/* texture_compression */
#define GL_COMPRESSED_ALPHA             0x84E9
#define GL_COMPRESSED_LUMINANCE                 0x84EA
#define GL_COMPRESSED_LUMINANCE_ALPHA           0x84EB
#define GL_COMPRESSED_INTENSITY             0x84EC
#define GL_COMPRESSED_RGB               0x84ED
#define GL_COMPRESSED_RGBA              0x84EE
#define GL_TEXTURE_COMPRESSION_HINT             0x84EF
#define GL_TEXTURE_COMPRESSED_IMAGE_SIZE        0x86A0
#define GL_TEXTURE_COMPRESSED               0x86A1
#define GL_NUM_COMPRESSED_TEXTURE_FORMATS           0x86A2
#define GL_COMPRESSED_TEXTURE_FORMATS           0x86A3
/* multisample */
#define GL_MULTISAMPLE                  0x809D
#define GL_SAMPLE_ALPHA_TO_COVERAGE             0x809E
#define GL_SAMPLE_ALPHA_TO_ONE              0x809F
#define GL_SAMPLE_COVERAGE              0x80A0
#define GL_SAMPLE_BUFFERS           0x80A8
#define GL_SAMPLES              0x80A9
#define GL_SAMPLE_COVERAGE_VALUE            0x80AA
#define GL_SAMPLE_COVERAGE_INVERT           0x80AB
#define GL_MULTISAMPLE_BIT              0x20000000
/* transpose_matrix */
#define GL_TRANSPOSE_MODELVIEW_MATRIX           0x84E3
```

```
#define GL_TRANSPOSE_PROJECTION_MATRIX      0x84E4
#define GL_TRANSPOSE_TEXTURE_MATRIX         0x84E5
#define GL_TRANSPOSE_COLOR_MATRIX       0x84E6
/* texture_env_combine */
#define GL_COMBINE                0x8570
#define GL_COMBINE_RGB                   0x8571
#define GL_COMBINE_ALPHA             0x8572
#define GL_SOURCE0_RGB                   0x8580
#define GL_SOURCE1_RGB                   0x8581
#define GL_SOURCE2_RGB                   0x8582
#define GL_SOURCE0_ALPHA             0x8588
#define GL_SOURCE1_ALPHA             0x8589
#define GL_SOURCE2_ALPHA             0x858A
#define GL_OPERAND0_RGB                  0x8590
#define GL_OPERAND1_RGB                  0x8591
#define GL_OPERAND2_RGB                  0x8592
#define GL_OPERAND0_ALPHA                0x8598
#define GL_OPERAND1_ALPHA                0x8599
#define GL_OPERAND2_ALPHA                0x859A
#define GL_RGB_SCALE                 0x8573
#define GL_ADD_SIGNED                0x8574
#define GL_INTERPOLATE               0x8575
#define GL_SUBTRACT                  0x84E7
#define GL_CONSTANT                  0x8576
#define GL_PRIMARY_COLOR             0x8577
#define GL_PREVIOUS               0x8578
/* texture_env_dot3 */
#define GL_DOT3_RGB                  0x86AE
#define GL_DOT3_RGBA                 0x86AF
/* texture_border_clamp */
#define GL_CLAMP_TO_BORDER               0x812D

GLAPI void GLAPIENTRY glActiveTexture( GLenum texture );

GLAPI void GLAPIENTRY glClientActiveTexture( GLenum texture );

GLAPI void GLAPIENTRY glCompressedTexImage1D( GLenum target, GLint level, GLen
um internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid *data );

GLAPI void GLAPIENTRY glCompressedTexImage2D( GLenum target, GLint level, GLen
um internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const G
Lvoid *data );

GLAPI void GLAPIENTRY glCompressedTexImage3D( GLenum target, GLint level, GLen
um internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imag
eSize, const GLvoid *data );

GLAPI void GLAPIENTRY glCompressedTexSubImage1D( GLenum target, GLint level, G
Lint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid *data );

GLAPI void GLAPIENTRY glCompressedTexSubImage2D( GLenum target, GLint level, G
Lint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSi
ze, const GLvoid *data );
```

```
GLAPI void GLAPIENTRY glCompressedTexSubImage3D( GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid *data );

GLAPI void GLAPIENTRY glGetCompressedTexImage( GLenum target, GLint lod, GLvoid *img );

GLAPI void GLAPIENTRY glMultiTexCoord1d( GLenum target, GLdouble s );

GLAPI void GLAPIENTRY glMultiTexCoord1dv( GLenum target, const GLdouble *v );

GLAPI void GLAPIENTRY glMultiTexCoord1f( GLenum target, GLfloat s );

GLAPI void GLAPIENTRY glMultiTexCoord1fv( GLenum target, const GLfloat *v );

GLAPI void GLAPIENTRY glMultiTexCoord1i( GLenum target, GLint s );

GLAPI void GLAPIENTRY glMultiTexCoord1iv( GLenum target, const GLint *v );

GLAPI void GLAPIENTRY glMultiTexCoord1s( GLenum target, GLshort s );

GLAPI void GLAPIENTRY glMultiTexCoord1sv( GLenum target, const GLshort *v );

GLAPI void GLAPIENTRY glMultiTexCoord2d( GLenum target, GLdouble s, GLdouble t );

GLAPI void GLAPIENTRY glMultiTexCoord2dv( GLenum target, const GLdouble *v );

GLAPI void GLAPIENTRY glMultiTexCoord2f( GLenum target, GLfloat s, GLfloat t );

GLAPI void GLAPIENTRY glMultiTexCoord2fv( GLenum target, const GLfloat *v );

GLAPI void GLAPIENTRY glMultiTexCoord2i( GLenum target, GLint s, GLint t );

GLAPI void GLAPIENTRY glMultiTexCoord2iv( GLenum target, const GLint *v );

GLAPI void GLAPIENTRY glMultiTexCoord2s( GLenum target, GLshort s, GLshort t );

GLAPI void GLAPIENTRY glMultiTexCoord2sv( GLenum target, const GLshort *v );

GLAPI void GLAPIENTRY glMultiTexCoord3d( GLenum target, GLdouble s, GLdouble t, GLdouble r );

GLAPI void GLAPIENTRY glMultiTexCoord3dv( GLenum target, const GLdouble *v );

GLAPI void GLAPIENTRY glMultiTexCoord3f( GLenum target, GLfloat s, GLfloat t, GLfloat r );

GLAPI void GLAPIENTRY glMultiTexCoord3fv( GLenum target, const GLfloat *v );

GLAPI void GLAPIENTRY glMultiTexCoord3i( GLenum target, GLint s, GLint t, GLint r );

GLAPI void GLAPIENTRY glMultiTexCoord3iv( GLenum target, const GLint *v );

GLAPI void GLAPIENTRY glMultiTexCoord3s( GLenum target, GLshort s, GLshort t, GLs
hort r );

GLAPI void GLAPIENTRY glMultiTexCoord3sv( GLenum target, const GLshort *v );

GLAPI void GLAPIENTRY glMultiTexCoord4d( GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q );

GLAPI void GLAPIENTRY glMultiTexCoord4dv( GLenum target, const GLdouble *v );

GLAPI void GLAPIENTRY glMultiTexCoord4f( GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q );

GLAPI void GLAPIENTRY glMultiTexCoord4fv( GLenum target, const GLfloat *v );

GLAPI void GLAPIENTRY glMultiTexCoord4i( GLenum target, GLint s, GLint t, GLint r, GLint q );

GLAPI void GLAPIENTRY glMultiTexCoord4iv( GLenum target, const GLint *v );

GLAPI void GLAPIENTRY glMultiTexCoord4s( GLenum target, GLshort s, GLshort t, GLshort r, GLshort q );

GLAPI void GLAPIENTRY glMultiTexCoord4sv( GLenum target, const GLshort *v );


GLAPI void GLAPIENTRY glLoadTransposeMatrixd( const GLdouble m[16] );

GLAPI void GLAPIENTRY glLoadTransposeMatrixf( const GLfloat m[16] );

GLAPI void GLAPIENTRY glMultTransposeMatrixd( const GLdouble m[16] );

GLAPI void GLAPIENTRY glMultTransposeMatrixf( const GLfloat m[16] );

GLAPI void GLAPIENTRY glSampleCoverage( GLclampf value, GLboolean invert );


typedef void (APIENTRYP PFNGLACTIVETEXTUREPROC) (GLenum texture);
typedef void (APIENTRYP PFNGLSAMPLECOVERAGEPROC) (GLclampf value, GLboolean invert);
typedef void (APIENTRYP PFNGLCOMPRESSEDTEXIMAGE3DPROC) (GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, const GLvoid *data);
typedef void (APIENTRYP PFNGLCOMPRESSEDTEXIMAGE2DPROC) (GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, const GLvoid *data);
typedef void (APIENTRYP PFNGLCOMPRESSEDTEXIMAGE1DPROC) (GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, const GLvoid *data);
typedef void (APIENTRYP PFNGLCOMPRESSEDTEXSUBIMAGE3DPROC) (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, const GLvoid *data);
typedef void (APIENTRYP PFNGLCOMPRESSEDTEXSUBIMAGE2DPROC) (GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, const GLvoid *data);
typedef void (APIENTRYP PFNGLCOMPRESSEDTEXSUBIMAGE1DPROC) (GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, const GLvoid *data);
typedef void (APIENTRYP PFNGLGETCOMPRESSEDTEXIMAGEPROC) (GLenum target, GLint level, GLvoid *img);



/*
 * GL_ARB_multitexture (ARB extension 1 and OpenGL 1.2.1)
 */
#ifndef GL_ARB_multitexture
#define GL_ARB_multitexture 1

#define GL_TEXTURE0_ARB                     0x84C0
#define GL_TEXTURE1_ARB                     0x84C1
#define GL_TEXTURE2_ARB                     0x84C2
#define GL_TEXTURE3_ARB                     0x84C3
#define GL_TEXTURE4_ARB                     0x84C4
#define GL_TEXTURE5_ARB                     0x84C5
#define GL_TEXTURE6_ARB                     0x84C6
#define GL_TEXTURE7_ARB                     0x84C7
#define GL_TEXTURE8_ARB                     0x84C8
#define GL_TEXTURE9_ARB                     0x84C9
#define GL_TEXTURE10_ARB                0x84CA
#define GL_TEXTURE11_ARB                0x84CB
#define GL_TEXTURE12_ARB                0x84CC
#define GL_TEXTURE13_ARB                0x84CD
#define GL_TEXTURE14_ARB                0x84CE
#define GL_TEXTURE15_ARB                0x84CF
#define GL_TEXTURE16_ARB                0x84D0
#define GL_TEXTURE17_ARB                0x84D1
#define GL_TEXTURE18_ARB                0x84D2
#define GL_TEXTURE19_ARB                0x84D3
#define GL_TEXTURE20_ARB                0x84D4
#define GL_TEXTURE21_ARB                0x84D5
#define GL_TEXTURE22_ARB                0x84D6
#define GL_TEXTURE23_ARB                0x84D7
#define GL_TEXTURE24_ARB                0x84D8
#define GL_TEXTURE25_ARB                0x84D9
#define GL_TEXTURE26_ARB                0x84DA
#define GL_TEXTURE27_ARB                0x84DB
#define GL_TEXTURE28_ARB                0x84DC
#define GL_TEXTURE29_ARB                0x84DD
#define GL_TEXTURE30_ARB                0x84DE
#define GL_TEXTURE31_ARB                0x84DF
#define GL_ACTIVE_TEXTURE_ARB                   0x84E0
#define GL_CLIENT_ACTIVE_TEXTURE_ARB                0x84E1
#define GL_MAX_TEXTURE_UNITS_ARB                0x84E2

GLAPI void GLAPIENTRY glActiveTextureARB(GLenum texture);
GLAPI void GLAPIENTRY glClientActiveTextureARB(GLenum texture);
GLAPI void GLAPIENTRY glMultiTexCoord1dARB(GLenum target, GLdouble s);
GLAPI void GLAPIENTRY glMultiTexCoord1dvARB(GLenum target, const GLdouble *v);
GLAPI void GLAPIENTRY glMultiTexCoord1fARB(GLenum target, GLfloat s);
GLAPI void GLAPIENTRY glMultiTexCoord1fvARB(GLenum target, const GLfloat *v);
GLAPI void GLAPIENTRY glMultiTexCoord1iARB(GLenum target, GLint s);
GLAPI void GLAPIENTRY glMultiTexCoord1ivARB(GLenum target, const GLint *v);
GLAPI void GLAPIENTRY glMultiTexCoord1sARB(GLenum target, GLshort s);
GLAPI void GLAPIENTRY glMultiTexCoord1svARB(GLenum target, const GLshort *v);
GLAPI void GLAPIENTRY glMultiTexCoord2dARB(GLenum target, GLdouble s, GLdouble t);
GLAPI void GLAPIENTRY glMultiTexCoord2dvARB(GLenum target, const GLdouble *v);
GLAPI void GLAPIENTRY glMultiTexCoord2fARB(GLenum target, GLfloat s, GLfloat t);
GLAPI void GLAPIENTRY glMultiTexCoord2fvARB(GLenum target, const GLfloat *v);
GLAPI void GLAPIENTRY glMultiTexCoord2iARB(GLenum target, GLint s, GLint t);
GLAPI void GLAPIENTRY glMultiTexCoord2ivARB(GLenum target, const GLint *v);
GLAPI void GLAPIENTRY glMultiTexCoord2sARB(GLenum target, GLshort s, GLshort t);
GLAPI void GLAPIENTRY glMultiTexCoord2svARB(GLenum target, const GLshort *v);
GLAPI void GLAPIENTRY glMultiTexCoord3dARB(GLenum target, GLdouble s, GLdouble t, GLdouble r);
GLAPI void GLAPIENTRY glMultiTexCoord3dvARB(GLenum target, const GLdouble *v);
GLAPI void GLAPIENTRY glMultiTexCoord3fARB(GLenum target, GLfloat s, GLfloat t, GLfloat r);
GLAPI void GLAPIENTRY glMultiTexCoord3fvARB(GLenum target, const GLfloat *v);
GLAPI void GLAPIENTRY glMultiTexCoord3iARB(GLenum target, GLint s, GLint t, GLint r);
GLAPI void GLAPIENTRY glMultiTexCoord3ivARB(GLenum target, const GLint *v);
GLAPI void GLAPIENTRY glMultiTexCoord3sARB(GLenum target, GLshort s, GLshort t, GLshort r);
GLAPI void GLAPIENTRY glMultiTexCoord3svARB(GLenum target, const GLshort *v);
GLAPI void GLAPIENTRY glMultiTexCoord4dARB(GLenum target, GLdouble s, GLdouble t, GLdouble r, GLdouble q);
GLAPI void GLAPIENTRY glMultiTexCoord4dvARB(GLenum target, const GLdouble *v);
GLAPI void GLAPIENTRY glMultiTexCoord4fARB(GLenum target, GLfloat s, GLfloat t, GLfloat r, GLfloat q);
GLAPI void GLAPIENTRY glMultiTexCoord4fvARB(GLenum target, const GLfloat *v);
GLAPI void GLAPIENTRY glMultiTexCoord4iARB(GLenum target, GLint s, GLint t, GLint r, GLint q);
GLAPI void GLAPIENTRY glMultiTexCoord4ivARB(GLenum target, const GLint *v);
GLAPI void GLAPIENTRY glMultiTexCoord4sARB(GLenum target, GLshort s, GLshort t, GLshort r, GLshort q);
GLAPI void GLAPIENTRY glMultiTexCoord4svARB(GLenum target, const GLshort *v);

typedef void (APIENTRYP PFNGLACTIVETEXTUREARBPROC) (GLenum texture);
typedef void (APIENTRYP PFNGLCLIENTACTIVETEXTUREARBPROC) (GLenum texture);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1DARBPROC) (GLenum target, GLdouble s);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1DVARBPROC) (GLenum target, const GLdouble *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1FARBPROC) (GLenum target, GLfloat s);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1FVARBPROC) (GLenum target, const GLfloat *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1IARBPROC) (GLenum target, GLint s);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1IVARBPROC) (GLenum target, c
onst GLint *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1SARBPROC) (GLenum target, G
Lshort s);
typedef void (APIENTRYP PFNGLMULTITEXCOORD1SVARBPROC) (GLenum target, c
onst GLshort *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2DARBPROC) (GLenum target, G
Ldouble s, GLdouble t);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2DVARBPROC) (GLenum target, c
onst GLdouble *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2FARBPROC) (GLenum target, G
Lfloat s, GLfloat t);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2FVARBPROC) (GLenum target, c
onst GLfloat *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2IARBPROC) (GLenum target, GL
int s, GLint t);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2IVARBPROC) (GLenum target, c
onst GLint *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2SARBPROC) (GLenum target, G
Lshort s, GLshort t);
typedef void (APIENTRYP PFNGLMULTITEXCOORD2SVARBPROC) (GLenum target, c
onst GLshort *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3DARBPROC) (GLenum target, G
Ldouble s, GLdouble t, GLdouble r);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3DVARBPROC) (GLenum target, c
onst GLdouble *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3FARBPROC) (GLenum target, G
Lfloat s, GLfloat t, GLfloat r);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3FVARBPROC) (GLenum target, c
onst GLfloat *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3IARBPROC) (GLenum target, GL
int s, GLint t, GLint r);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3IVARBPROC) (GLenum target, c
onst GLint *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3SARBPROC) (GLenum target, G
Lshort s, GLshort t, GLshort r);
typedef void (APIENTRYP PFNGLMULTITEXCOORD3SVARBPROC) (GLenum target, c
onst GLshort *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4DARBPROC) (GLenum target, G
Ldouble s, GLdouble t, GLdouble r, GLdouble q);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4DVARBPROC) (GLenum target, c
onst GLdouble *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4FARBPROC) (GLenum target, G
Lfloat s, GLfloat t, GLfloat r, GLfloat q);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4FVARBPROC) (GLenum target, c
onst GLfloat *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4IARBPROC) (GLenum target, GL
int s, GLint t, GLint r, GLint q);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4IVARBPROC) (GLenum target, c
onst GLint *v);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4SARBPROC) (GLenum target, G
Lshort s, GLshort t, GLshort r, GLshort q);
typedef void (APIENTRYP PFNGLMULTITEXCOORD4SVARBPROC) (GLenum target, c
onst GLshort *v);
```

```
#endif /* GL_ARB_multitexture */



/*
 * Define this token if you want "old-style" header file behaviour (extensions
 * defined in gl.h).  Otherwise, extensions will be included from glext.h.
 */
#if defined(GL_GLEXT_LEGACY)

/* All extensions that used to be here are now found in glext.h */

#else  /* GL_GLEXT_LEGACY */

#include <GL/glext.h>

#endif  /* GL_GLEXT_LEGACY */



/*
 * ???. GL_MESA_packed_depth_stencil
 * XXX obsolete
 */
#ifndef GL_MESA_packed_depth_stencil
#define GL_MESA_packed_depth_stencil 1

#define GL_DEPTH_STENCIL_MESA			0x8750
#define GL_UNSIGNED_INT_24_8_MESA		0x8751
#define GL_UNSIGNED_INT_8_24_REV_MESA		0x8752
#define GL_UNSIGNED_SHORT_15_1_MESA		0x8753
#define GL_UNSIGNED_SHORT_1_15_REV_MESA		0x8754

#endif /* GL_MESA_packed_depth_stencil */


#ifndef GL_ATI_blend_equation_separate
#define GL_ATI_blend_equation_separate 1

#define GL_ALPHA_BLEND_EQUATION_ATI		0x883D

GLAPI void GLAPIENTRY glBlendEquationSeparateATI( GLenum modeRGB, GLenum modeA );
typedef void (APIENTRYP PFNGLBLENDEQUATIONSEPARATEATIPROC) (GLenum modeRGB, GLenum modeA);

#endif /* GL_ATI_blend_equation_separate */


/* GL_OES_EGL_image */
#ifndef GL_OES_EGL_image
typedef void* GLeglImageOES;
#endif
```

```
#ifndef GL_OES_EGL_image
#define GL_OES_EGL_image 1
#ifdef GL_GLEXT_PROTOTYPES
GLAPI void APIENTRY glEGLImageTargetTexture2DOES (GLenum target, GLeglImageOES image);
GLAPI void APIENTRY glEGLImageTargetRenderbufferStorageOES (GLenum target, GLeglImageOES image);
#endif
typedef void (APIENTRYP PFNGLEGLIMAGETARGETTEXTURE2DOESPROC) (GLenum target, GLeglImageOES image);
typedef void (APIENTRYP PFNGLEGLIMAGETARGETRENDERBUFFERSTORAGEOESPROC) (GLenum target, GLeglImageOES image);
#endif


/**
 ** NOTE!!!!!  If you add new functions to this file, or update
 ** glext.h be sure to regenerate the gl_mangle.h file.  See comments
 ** in that file for details.
 **/


#ifdef __cplusplus
}
#endif

#endif /* __gl_h_ */
```