**NAME**

xgl_multimarker – draws a list of Markers

**SYNOPSIS**

```
#include <xgl/xgl.h>
void  xgl_multimarker (
        Xgl_ctx        ctx,
        Xgl_pt_list    *pl );
```

**DESCRIPTION**

**Purpose**

This operator draws a Marker (a symbol specified by the **XGL_CTX_MARKER**(3) attribute) at each control point in *pl→pts*. The control points in *pl→pts* are transformed and clipped. If a control point is outside the clipping bounds, no Marker is drawn. If the control point is inside the clipping bounds, a Marker is drawn; however, the symbol is clipped at the clipping bounds.

The Marker is drawn in a plane parallel to the plane of the front of the Device. For 3D Contexts, this means its size is not affected by the perspective transformation — Markers at different *z* values have the same size. The size of the Marker is controlled by the attribute **XGL_CTX_MARKER_SCALE_FACTOR**(3)**.** The size (width or height) is the result of the XGL_CTX_MARKER_SCALE_FACTOR multiplied by the nominal Marker size, which is 1 pixel.

The attribute **XGL_CTX_MARKER_COLOR**(3) sets the color of all the Markers in the primitive. If a color is supplied at each point (see the discussion of valid point types below), however, the point color is used instead.

Other attributes that affect the appearance of a Marker are:

- **XGL_CTX_PLANE_MASK**(3)
- **XGL_CTX_ROP**(3)
- **XGL_3D_CTX_DEPTH_CUE_MODE**(3)
- **XGL_3D_CTX_DEPTH_CUE_COLOR**(3)

**Marker Definitions**

The application can use Markers that it has created (see **XGL_MARKER_DESCRIPTION**(3) for a description of application-defined Markers) or it can use pre-defined Markers. (For a list of pre-defined Markers, see **XGL_CTX_MARKER**(3)**.** )

The Markers are drawn on the Device associated with the Context *ctx*. The Context may be 2D or 3D, and it determines the types of points that can be given to xgl_multimarker.

If the application data associated with this Marker object is a pointer to a floating point scale factor, when rendering through the CGM stream pipeline, the marker will be stroked out and scaled by a factor equal to the product of **XGL_CTX_MARKER_SCALE_-FACTOR**(3) and the float pointed to by **XGL_OBJ_APPLICATION_DATA**(3). If there is no object application data associated with the marker, then a marker other than one of the predefined markers will be rendered by the CGM stream pipeline as a dot marker. Note that the **XGL_OBJ_APPLICATION_DATA**(3) scale factor only applies to the CGM stream pipeline. See the *XGL Programmer's Guide* for an example.

**Parameters**

*ctx* The XGL 2D or 3D Context containing the graphics state used when processing **xgl_multimarker**(3)**.**

*pl* A pointer to an *Xgl_pt_list* structure that is defined as:

```
typedef struct {
        Xgl_pt_type        pt_type;          /* type of point */
        Xgl_bbox           *bbox;            /* optional box containing
                                                all points */
        Xgl_usgn32         num_pts;          /* number of points */
        Xgl_usgn32         num_data_values;  /* number of data values at
                                                each point */
        Xgl_pt_ptr_union   pts;              /* union of pointers to array
                                                of points */
} Xgl_pt_list;
```

A Marker is drawn at each point specified in the array *pl*→*pts*.

For a complete description of the possible structures, see **xgl_pt_list**(3) and **xgl_struct**(3)**.**

**RETURN VALUE**

Nothing is returned.

**SEE ALSO**

**xgl_object_get**(3)
**xgl_object_set**(3)
**XGL_CTX_PLANE_MASK**(3)
**XGL_CTX_ROP**(3)
**XGL_3D_CTX_DEPTH_CUE_MODE**(3)
**XGL_3D_CTX_DEPTH_CUE_COLOR**(3)
**XGL_CTX_MARKER**(3)
**XGL_CTX_MARKER_AA_BLEND_EQ**(3)
**XGL_CTX_MARKER_AA_FILTER_SHAPE**(3)
**XGL_CTX_MARKER_AA_FILTER_WIDTH**(3)
**XGL_CTX_MARKER_COLOR**(3)
**XGL_CTX_MARKER_COLOR_SELECTOR**(3)
**XGL_CTX_MARKER_SCALE_FACTOR**(3)