NAME

XGL_CTX_VDC_MAP,
XGL_CTX_VDC_WINDOW,
XGL_CTX_DC_VIEWPORT – control mapping from Virtual Device Coordinates (VDC) to Device Coordinates (DC)

SYNOPSIS

#include <xgl/xgl.h>

Xgl_ctx ctx;

xgl_object_set(ctx, XGL_CTX_VDC_MAP, Xgl_vdc_map, 0);
xgl_object_get(ctx, XGL_CTX_VDC_MAP, Xgl_vdc_map ∗);

For a 2D Context:
xgl_object_set(ctx, XGL_CTX_VDC_WINDOW, Xgl_bounds_d2d ∗, 0);
xgl_object_set(ctx, XGL_CTX_DC_VIEWPORT, Xgl_bounds_d2d ∗, 0);

xgl_object_get(ctx, XGL_CTX_VDC_WINDOW, Xgl_bounds_d2d ∗);
xgl_object_get(ctx, XGL_CTX_DC_VIEWPORT, Xgl_bounds_d2d ∗);

For a 3D Context:
xgl_object_set(ctx, XGL_CTX_VDC_WINDOW, Xgl_bounds_d3d ∗, 0);
xgl_object_set(ctx, XGL_CTX_DC_VIEWPORT, Xgl_bounds_d3d ∗, 0);

xgl_object_get(ctx, XGL_CTX_VDC_WINDOW, Xgl_bounds_d3d ∗);
xgl_object_get(ctx, XGL_CTX_DC_VIEWPORT, Xgl_bounds_d3d ∗);

DESCRIPTION

These attributes control the mapping of the primitives' geometrical data from virtual device coordinates (VDC) to device coordinates (DC). The mapping transforms a rectangular region (an area for 2D Contexts and a volume for 3D Contexts), aligned with the axes, from VDC to DC. It cannot represent rotations or the perspective transform; these types of transformations are handled by the XGL Transform objects specified by the **XGL_CTX_LOCAL_MODEL_TRANS**(3)**, XGL_CTX_GLOBAL_MODEL_TRANS**(3)**,** and **XGL_CTX_VIEW_TRANS**(3) attributes.

The attribute **XGL_CTX_VDC_MAP**(3) specifies whether the application provides the mapping from VDC to DC, or whether XGL provides part of the mapping. Another Context attribute, **XGL_CTX_VDC_ORIENTATION**(3) controls the interpretation of coordinates in the VDC system.

XGL_CTX_VDC_MAP can take on one of four values as specified by the enumerated type *Xgl_vdc_map*:

XGL_VDC_MAP_OFF

The application specifies the VDC mapping by specifying both XGL_CTX_VDC_WINDOW and XGL_CTX_DC_VIEWPORT. In this case, to supply the correct regions for the mapping, the application must know the size limits of the Device and the value of XGL_CTX_VDC_ORIENTATION.

If the XGL_CTX_VDC_ORIENTATION is XGL_Y_DOWN_Z_AWAY, the VDC window points are mapped to the DC viewport points as follows:

VDC		DC
(xmin, ymin [, zmin])	→	*(xmin, ymin [, zmin])*

(xmax, ymax [, zmax])	→	*(xmax, ymax [, zmax])*

If the XGL_CTX_VDC_ORIENTATION is XGL_Y_UP_Z_TOWARD, the VDC window points are mapped to the DC viewport points as follows:

VDC		DC
(xmin, ymin [, zmin])	→	*(xmin, ymax [, zmax])*
(xmax, ymax [, zmax])	→	*(xmax, ymin [, zmin])*

If you specify the VDC window or DC viewport bounds, XGL always requires that *xmin* < *xmax*, *ymin* < *ymax*, and, for 3D, *zmin* < *zmax*.

If the application tries to set the value of XGL_CTX_DC_VIEWPORT so that the DC viewport extends partly or entirely outside of the Device's boundaries, XGL will not accept this value. Instead, XGL will set XGL_CTX_DC_VIEWPORT to the entire extent of the Device. This is usually not the desired effect, so applications should detect this condition and act accordingly. If the DC viewport is partially outside the Device's boundaries, the application should reduce the size of the DC viewport so that it is entirely inside the Device, and it should reduce the VDC window by a proportional amount. If the DC viewport is entirely outside of the Device's boundaries, the application can set the value of XGL_CTX_RENDERING to FALSE so that primitives will not draw into the Device.

XGL_VDC_MAP_ALL

The application supplies the XGL_CTX_VDC_WINDOW and XGL sets the XGL_CTX_-DC_VIEWPORT to use the full range of the Device's coordinate space. This may result in the image being scaled differently along the *x*-axis and *y*-axis.

XGL_VDC_MAP_ASPECT

Similar to XGL_VDC_MAP_ALL, except that the aspect ratio (ratio of the width to the height) of the XGL_CTX_VDC_WINDOW is preserved in the transformation to DC. The VDC window is scaled so that it uses as much of the DC space as possible while maintaining its aspect ratio. Thus, not all of the DC space may be used. The top left corner of the XGL_CTX_VDC_WINDOW is mapped to the top left corner (0, 0) of the Device.

XGL_VDC_MAP_DEVICE

XGL_CTX_VDC_WINDOW is ignored. XGL sets the XGL_CTX_DC_VIEWPORT to use the full range of the Device's coordinate space, and sets the DC mapping so that VDC maps directly to DC; one unit in VDC is equal to one unit in DC.

In the case of a Raster Device, this allows the application to access pixels directly.

If the Device is a Window Raster Device, its size can be changed by the operator; this affects the VDC-to-DC mapping. When the window is resized, the application must adjust the VDC-to-DC map itself if the XGL_CTX_VDC_MAP is XGL_VDC_MAP_OFF. If the XGL_CTX_VDC_MAP is some other value, the application needs to call **xgl_window_raster_-resize**(3)**.**

The default value for XGL_CTX_VDC_MAP is XGL_VDC_MAP_DEVICE.

The default value for XGL_CTX_VDC_WINDOW is [-1, 1]×[-1, 1] for 2D Contexts and [-1, 1]×[-1, 1]×[0, 1] for 3D Contexts.

The default value for XGL_CTX_DC_VIEWPORT is [-1, 1]×[-1, 1] for 2D Contexts and [-1, 1]×[-1, 1]×[0, 1] for 3D Contexts. However, as long as XGL_CTX_VDC_MAP is different from XGL_VDC_MAP_OFF, each time the XGL_CTX_DEVICE is changed, XGL_CTX_DC_VIEWPORT is reset to [0, M_x]×[0, M_y] for 2D Contexts and [0, M_x]×[0, M_y]×[0, M_z] for 3D Contexts where (M_x, M_y, M_z) are a function of both XGL_CTX_VDC_MAP and the maximum dimensions of the device **XGL_DEV_MAXIMUM_COORDINATES**(3) (see the rules above).

These attributes are pushed by **xgl_context_push**(3).

SEE ALSO

xgl_object_set(3)
xgl_object_get(3)
xgl_context_push(3)
xgl_window_raster_resize(3)

XGL_CTX_CLIP_PLANES(3)
XGL_CTX_GLOBAL_MODEL_TRANS(3)
XGL_CTX_LOCAL_MODEL_TRANS(3)
XGL_CTX_VDC_ORIENTATION(3)
XGL_CTX_VIEW_CLIP_BOUNDS(3)
XGL_CTX_VIEW_TRANS(3)
XGL_DEV_MAXIMUM_COORDINATES(3)