NAME

XGL_CTX_ROP – defines the Raster operation (ROP)

SYNOPSIS

#include <xgl/xgl.h>

Xgl_ctx ctx;

xgl_object_set(ctx, XGL_CTX_ROP, Xgl_rop_mode, 0);
xgl_object_get(ctx, XGL_CTX_ROP, Xgl_rop_mode *);

DESCRIPTION

This attribute defines the Raster operation (ROP) for the Context *ctx*. A ROP specifies how the *source* and *destination* pixels are logically combined to produce the result written to the Destination Raster. All XGL graphics primitives are rendered using this logical operation to combine the source and destination information. For primitive operations, the *source* is the set of pixels comprising the graphics primitive (for example, the pixels along a line produced by the **xgl_multipolyline**(3) call), and the *destination* is the Raster associated with the current Context.

The source, destination, and plane_mask are combined, using the algorithm shown below, to yield the new destination pixel values. For each bit in each pixel of the destination drawable, the following expression defines whether that bit is set.

((rop(*src*,*dst*) AND (plane_mask AND plane_mask_mask)) OR (*dst* AND (NOT (plane_mask AND plane_mask_mask)))

where:
src is the source pixel to be rendered
dst is the pixel value in the frame buffer
plane_mask is the value of XGL_CTX_PLANE_MASK
rop is the ROP defined by XGL_CTX_ROP
plane_mask_mask is one of the following cases:

- If the device is NOT a window raster, or if the application manages itself the X colormap (i.e. it has set **XGL_CMAP_NAME**(3) of the Color Map object associated with the device that is associated with the context), then the value used for *plane_mask_mask* is -1.
- If the device is a window raster, and XGL manipulates the X colormap on behalf of the application (ie: the application has NOT set **XGL_CMAP_NAME**(3) of the Color Map object associated with the device that is associated with the context), then *plane_mask_mask* is the set of X pixel bits that XGL has allocated from X.

This is done in hardware for accelerated window rasters.

The options available for this attribute are shown below along with the corresponding logical operation:

XGL_ROP_CLEAR
: Clear destination – *result = 0 (i.e. all bits are reset to 0)*

XGL_ROP_SET
: Set destination – *result = -1 (i.e. all bits are set to 1)*

XGL_ROP_COPY
Copy source to destination – *result = source*

XGL_ROP_COPY_INVERTED
Copy 1's complement of source to destination – *result = !source*

XGL_ROP_INVERT
Invert destination – *result = !dest*

XGL_ROP_AND
AND source and destination – *result = source & dest*

XGL_ROP_AND_REVERSE
For each bit set in the destination, clear the source and write the source to the destination – *result = source & (!dest)*

XGL_ROP_AND_INVERTED
For each bit set in the source, erase destination – *result = (!source) & dest*

XGL_ROP_NAND
NAND source with destination – *result = !(source & dest)*

XGL_ROP_OR
OR source with destination – *result = source | dest*

XGL_ROP_OR_REVERSE
Replace the destination with the OR of the source and the 1's complement of the destination – *result = source | (!dest)*

XGL_ROP_OR_INVERTED
OR of the destination with the 1's complement of the source – *result = !(source) | dest*

XGL_ROP_NOR
NOR source with destination – *result = !(source | dest)*

XGL_ROP_XOR
Exclusive OR source with destination – *result = source ˆ dest*

XGL_ROP_EQUIV
For each bit in source and destination that are equivalent, set the bit in the destination. For those that are not equivalent, clear the destination – *result = !(source ˆ dest)*

XGL_ROP_NOOP
Do nothing – *result = dest*

The default value is XGL_ROP_COPY.

This attribute is pushed by **xgl_context_push**(3).

SEE ALSO

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

XGL_CMAP_NAME(3)
XGL_CTX_PLANE_MASK(3)

NOTE

For accelerated RGB devices, refer to your accelerator guide.