NAME

xil_to_xgl, xgl_to_xil – convert between XIL images and XGL rasters

SYNOPSIS

#include <xil/xil.h>
#include <xgl/xgl.h>

Xgl_ras xil_to_xgl (XilImage *src***,**
    Xgl_sys_state *xgl_state***);**

XilImage xgl_to_xil (Xgl_ras *src***,**
    XilSystemState *xil_state***);**

DESCRIPTION

These functions convert images between the XIL library's memory/display images and the XGL library's memory/display rasters. *src* is the source image/raster handle. If the function is successful, a handle to a raster/image is returned. Note that it is not possible to convert an XGL memory raster if an XIL memory image's scanlines are not 16-bit aligned, because of the alignment difference between XIL memory images and XGL memory rasters. Supported conversions are as follows:

XGL_COLOR_INDEX display raster	<->	XIL one-band byte display image
XGL_COLOR_RGB display raster	<->	XIL three-band byte display image
XGL_COLOR_INDEX memory raster	<->	XIL one-band byte memory image
XGL_COLOR_RGB memory raster	<->	XIL four-band byte memory image

ERRORS

For a complete list of XIL error messages by number, consult Appendix B of the *XIL Programmer's Guide.*

EXAMPLES

```
XilSystemState State;
Xgl_sys_state xgl_state;
Display *display;
Window window;
XilImage memory_image, display_image;
Xgl_ras win_ras, mem_ras;
Xgl_ctx ctx;

xil_state = xil_open();
xgl_state = xgl_open(NULL);

display_image = xil_create_from_window(State, display, window);
memory_image = xil_create(State, 512, 512, 4, XIL_BYTE);

win_ras = xil_to_xgl(display_image, xgl_state);
mem_ras = xil_to_xgl(memory_image, xgl_state);
ctx = xgl_object_create(xgl_state, XGL_2D_CTX, NULL, XGL_CTX_DEVICE,
        mem_ras, NULL);
```

(XGL library calls)

xgl_context_flush(ctx, XGL_FLUSH_SYNCHRONIZE);

(XIL library calls)

xil_sync(memory_image);

NOTES

An application should call **xgl_context_flush** (*ctx, XGL_FLUSH_SYNCHRONIZE*) before switching from the XGL to the XIL library, and call **xil_sync** (*image*) before switching from the XIL to the XGL library. Applications need to link with the following libraries to use these routines: XIL, XGL, and libxil_to_xgl.

This interface to XIL/XGL interoperability is "uncommitted," which means it may change or be replaced by another mechanism in a future release.

SEE ALSO

xgl_object_create (3), **xil_create**(3), **xil_create_from_device**(3), **xil_create_from_window**(3), **xil_sync**(3).