XIL-XGL Interoperability C

The current release of the XIL imaging library includes two functions that enable you to process an image using both XIL and XGL 3.*x* calls in the same program.

Note – This interface for XIL-XGL interoperability may change or be replaced by another mechanism in a future release.

One of these functions, `xil_to_xgl()`, enables you to convert an XIL memory image to an XGL memory raster *or* to convert an XIL display image to an XGL window raster. The prototype for this function is shown below.

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

The parameter `src` is a handle to an XIL memory image or display image, and `xgl_state` is a handle to an XGL System State object. If the function is successful, the return value will be a handle to an XGL memory raster or window raster. If the functions fails, the return value will be `NULL`.

The other function, `xgl_to_xil()`, enables you to convert an XGL memory raster to an XIL memory image or to convert an XGL window raster to an XIL display image.

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

The parameter `src` is a handle to an XGL memory raster or window raster, and `xil_state` is an XIL system state. The function's return value is a handle to an XIL memory image or display image. If the function fails, the return value will be `NULL`.

Table C-1 below lists the specific conversions that these functions make possible. Within each row, the XIL image listed in the left column can be converted to the XGL object in the right column and vice versa.

Table C-1 XIL-XGL Interoperability

XIL Image	XGL Object
A one-band `XIL_BYTE` memory image	A memory raster with the attribute `XGL_COLOR_INDEX`
A four-band `XIL_BYTE` memory image	A memory raster with the attribute `XGL_COLOR_RGB`
A one-band `XIL_BYTE` display image	A window raster with the attribute `XGL_COLOR_INDEX`
A three-band `XIL_BYTE` display image	A window raster with the attribute `XGL_COLOR_RGB`

Note – The conversions shown in the first row of Table C-1 are possible only if the width of the XIL image is an even number.

Once you have both an XIL and an XGL handle to an object, you can process it using either XIL functions or XGL functions. The only restriction on combining these calls is that there cannot be any outstanding deferred operations on the object at the time you switch from making XIL calls to XGL calls, or vice versa. This means that if you have been processing an image using XIL functions, you must call `xil_sync()` to flush all outstanding operations on the image before operating on the image using XGL functions. Similarly, if you have been processing a raster using XGL functions, you must call `xgl_context_flush()` before performing any XIL operations on the raster.

One last note. The XIL-XGL conversion functions are not literally part of the XIL library. Instead, they reside in a separate library called `libxil_to_xgl.a`. This means that programs that use the conversion functions must link with

- `-lxil`
- `-lxgl`
- `-lxil_to_xgl`