NAME

XGL_WIN_RAS – a Window Raster Device object

OVERVIEW

XGL_WIN_RAS is the type of an object representing a Window Raster Device. It is used as the *type* parameter value for **xgl_object_create**(3)**.**

A specific Device object is the Window Raster Device, which draws into *pixels* that are stored in a *window* on a graphics frame buffer managed by a window system. These pixels are visible to the operator. In contrast, the Memory Raster Device object (see **xgl_object_create**(3) and **XGL_MEM_RAS**(3)) stores images in memory where they are not visible. The application must create the window before creating an XGL Window Raster Device on it.

A window can have only one Window Raster Device associated with it.

A Window Raster Device object is created using the **xgl_object_create**(3) operator. The application must specify XGL_WIN_RAS as the value of the *type* parameter to **xgl_object_create**(3)**.** In addition, the application must also supply the *desc* parameter, which is a pointer to an *Xgl_obj_desc* union. The application needs to supply the information in the *win_ras* field of the *Xgl_obj_desc* union. The *win_ras* structure consists of two fields: *type*, which is the window type, and *desc*, which is a pointer to descriptor information that identifies the window of the given type. The possible values for the window type are (currently, there is only one value):

XGL_WIN_X

The Window Raster Device is associated with an X11 window that may have been obtained from an XView Canvas or created by other means. The value of *desc* is a pointer to an *Xgl_X_window* structure. This example shows how to create an XGL_WIN_RAS on an X window:

```
#include <xgl/xgl.h>
Xgl_X_window     xgl_x;          /* X window descriptor */
Xgl_obj_desc     obj_desc;       /* XGL object descriptor */
Xgl_win_ras      ras;     /* newly created window raster */
Xgl_sys_state    sys_state;      /* XGL system state */

xgl_x.X_display = /* the X display identifier */
xgl_x.X_screen = /* the X screen identifier */
xgl_x.X_window = /* the X window identifier */

obj_desc.win_ras.type = XGL_WIN_X;
obj_desc.win_ras.desc = &xgl_x;

ras = xgl_object_create(sys_state, XGL_WIN_RAS, &obj_desc, 0);
```

The window system type can be modified to indicate which protocol XGL uses to communicate with the graphics device. XGL supports three protocols for the X window system: Xlib, PEX, and DGA (Direct Graphics Access). The protocol is selected by OR’ing in the name of the protocol with the constant XGL_WIN_X. The constants for the names of the protocol are:

```
/* WIN_X protocols */
#define XGL_WIN_X_PROTO_DEFAULT     (0x000)
#define XGL_WIN_X_PROTO_XLIB        (0x100)
#define XGL_WIN_X_PROTO_PEX         (0x200)
#define XGL_WIN_X_PROTO_DGA         (0x400)
```

Only one protocol can be specified. If a protocol is specified, XGL tries to use that protocol to communicate with the graphics device. If it is not available, an error is issued and the Window Raster is not created. If no protocol is specified (or XGL_WIN_X_PROTO_DEFAULT is used), XGL tries to use the *best* protocol for the particular device. If that is not available, XGL will try a backup protocol. In practical terms, XGL tries to use DGA, but if that is not available, it will attempt to use PEX, and if that fails, then it will use Xlib.

In most cases, an XGL application does not need to be concerned with selecting a protocol. However, this facility is available to aid applications that want to make the best use of the graphics resources, and this usually means part of the application that is device-dependent.

After creating the Window Raster Device, using **xgl_object_create**(3), the application can inquire about the values of its attributes by using the **xgl_object_get**(3) operator.

A Window Raster Device object shares many attributes with a Memory Raster Device object. These attributes have the common prefix XGL_RAS_. The attributes that are specific to the Window Raster Device have the prefix XGL_WIN_RAS_.

A brief overview of all of the attributes follows. For more information, see the man page for each attribute.

XGL_DEV_COLOR_MAP(3)
Specifies the Color Map Object used by the Window Raster Device object.

XGL_DEV_COLOR_TYPE(3)
Specifies the color type (that is, the color model) of the Raster. It can be set only once, when the Window Raster Device object is created. The two alternatives are: XGL_COLOR_INDEX and XGL_COLOR_RGB.

XGL_DEV_CONTEXTS(3)
XGL_DEV_CONTEXTS_NUM(3)

These attributes return the Context object handles used by the Window Raster Device object.

XGL_DEV_MAXIMUM_COORDINATES(3)
Returns the maximum coordinate values for the Window Raster Device object.

XGL_DEV_PICK_BUFFER_SIZE(3)
Returns the maximum pick buffer size that a Window Raster Device object can support efficiently.

XGL_DEV_REAL_COLOR_TYPE(3)
Specifies the real color space used by the Window Raster Device object.

XGL_OBJ_APPLICATION_DATA(3)
This attribute allows the application program to store its own data with an object.

XGL_OBJ_SYS_STATE(3)
This read-only attribute returns the System State that was used to create the object.

XGL_OBJ_TYPE(3)
This read-only attribute defines the type of an XGL object.

XGL_RAS_DEPTH(3)
XGL_RAS_HEIGHT(3)
XGL_RAS_WIDTH(3)

These attributes give the size of the Window Raster Device. They are read-only and cannot be set for a Window Raster Device. The width and height may change, however, if the operator changes the size of the window using the window system.

XGL_RAS_RECT_LIST(3)
XGL_RAS_RECT_NUM(3)

These attributes define the number and list of clip rectangles in the application-specified clip list.

XGL_RAS_SOURCE_BUFFER(3)
This attribute specifies the buffer to be used as source during raster operations.

XGL_WIN_RAS_BACKING_STORE(3)
This attribute provides backing store support for a Window Raster.

XGL_WIN_RAS_BUF_DISPLAY(3)
XGL_WIN_RAS_BUF_DRAW(3)
XGL_WIN_RAS_BUF_MIN_DELAY(3)
XGL_WIN_RAS_BUFFERS_ALLOCATED(3)
XGL_WIN_RAS_BUFFERS_REQUESTED(3)

These attributes control double buffering. XGL_WIN_RAS_BUF_DISPLAY specifies the current display buffer. XGL_WIN_RAS_BUF_DRAW specifies the current read/write buffer. XGL_WIN_RAS_BUF_MIN_DELAY specifies the minimum time delay between buffer switches. XGL_WIN_RAS_BUFFERS_ALLOCATED reflects the number of buffers that are available to the application for use. XGL_WIN_RAS_BUFFERS_REQUESTED specifies the number of hardware buffers requested by the application.

XGL_WIN_RAS_DESCRIPTOR(3)
Returns the window descriptor associated with the Window Raster Device. This is the value of the *win_ras_desc* field in the *Xgl_obj_desc* union.

XGL_WIN_RAS_PIXEL_MAPPING(3)
The *pixel mapping array* that maps the application's color indices to hardware color indices.

XGL_WIN_RAS_POSITION(3)
Specifies the position of the Window Raster object.

XGL_WIN_RAS_TYPE(3)
Returns the window type set via the *type* parameter to the create call.

XGL_WIN_RAS_STEREO_MODE(3)
Specifies whether the window being rendered into is a stereo window or a mono window.

NOTE

The current implementation of XGL, using Direct Graphics Access (DGA), does not support using fork() or vfork() if there is an active XGL Window Raster.