NAME

intro – introduces the XGL graphics library

DESCRIPTION

These Reference Manual pages describe the syntax for using the operators (functions) and attributes in the XGL Graphics Library. Special sections immediately following this introduction describe the XGL enumerated types, XGL macro values, XGL point list structures, and XGL data structures. The next two sections describe, in alphabetical order, the XGL attributes and operators.

The XGL programming paradigm is built around *objects*, which are instances of abstract data types. The application can create an object and apply operators to it. The application cannot, however, extend the definition of an object or modify the behavior of operators. Therefore, XGL is not an object-oriented programming system, although its design does incorporate some object-oriented concepts.

The XGL operators are standard C language library calls and are documented as such. Note that some operators can take a variable number of parameters. For example, the operator xgl_object_set is described this way:

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

void  xgl_object_set (
          Xgl_object          obj,
          <attribute-list>    attributes );
```

The first parameter is of the type **Xgl_object**, which is a handle to the XGL object data. The next argument, whose type is shown as *<attribute-list>*, is a *variable length* list of attribute-value pairs. A zero (ASCII NULL character) is used to mark the end of the list.

Each XGL object has a set of attributes associated with it. An attribute is a named field inside the object. The application can set an attribute's value, or it can retrieve its value, or both. The documentation for the XGL attributes specifies the C data type needed for setting the attribute and the C data type needed for getting (retrieving) its value. The following example describes how to set and get the attribute **XGL_CTX_LINE_WIDTH_SCALE_-FACTOR**(3), which controls the width of lines in the Context object.

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

Xgl_ctx        ctx;

xgl_object_set(ctx, XGL_CTX_LINE_WIDTH_SCALE_FACTOR, float, 0);
xgl_object_get(ctx, XGL_CTX_LINE_WIDTH_SCALE_FACTOR, float *);
```

This notation indicates that to set the attribute value, the application passes a *float*, and to get the value, the application passes a *pointer* to a *float*, which is overwritten with a new value.

Naming Scheme

The XGL library defines symbols that start with *XGL*, *Xgl*, *xgl*, *XGLI*, *Xgli*, and *xgli*. The application program should not define any symbols (type names, enumerated values, cpp definitions, function names, and so on) beginning with these prefixes.

Printed Manual vs. On-line Manual

Most attributes and operators are discussed on their own separate manual page. In some cases, however, several related attributes are discussed together on a single manual page. In the printed reference manual, only the page where they are discussed is printed, but for the on-line manual, you can use **man** (1) for any of the topics discussed on that page.

For example, both **XGL_3D_CTX_SURF_BACK_AMBIENT**(3) and **XGL_3D_CTX_SURF_FRONT_-AMBIENT**(3) are discussed on the page for **XGL_3D_CTX_SURF_FRONT_AMBIENT**(3)**.** In the printed reference manual, you will find only the page for **XGL_3D_CTX_SURF_FRONT_-AMBIENT**(3) printed. You can look up **XGL_3D_CTX_SURF_BACK_AMBIENT**(3) in the index to see where it is discussed. You can issue either of the following commands for the on-line version, however:

man XGL_3D_CTX_SURF_BACK_AMBIENT

man XGL_3D_CTX_SURF_FRONT_AMBIENT

You can also access the following descriptions of other aspects of XGL:

Data Structures
: described in **xgl_struct**(3)

Enumerated Types
: described in **xgl_enum_types**(3)

Macro Values
: described in **xgl_macro_values**(3)

Point Lists
: described in **xgl_pt_list**(3)

Object Types
: described on their individual man pages

XGL_2D_CTX(3)
XGL_3D_CTX(3)
XGL_CGM_DEV(3)
XGL_CMAP(3)
XGL_DMAP_TEXTURE(3)
XGL_GCACHE(3)
XGL_LIGHT(3)
XGL_LPAT(3)
XGL_MARKER(3)
XGL_MEM_RAS(3)
XGL_MIPMAP_TEXTURE(3)
XGL_PCACHE(3)
XGL_SFONT(3)
XGL_STREAM(3)
XGL_SYS_STATE(3)
XGL_TMAP(3)
XGL_TRANS(3)
XGL_WIN_RAS(3)