NAME

glNewList, glEndList – create or replace a display list

C SPECIFICATION

void **glNewList**(GLuint *list*,

GLenum *mode*)

PARAMETERS

list Specifies the display-list name.

mode

Specifies the compilation mode, which can be **GL_COMPILE** or **GL_COMPILE_AND_EXECUTE**.

C SPECIFICATION

void **glEndList**(void)

DESCRIPTION

Display lists are groups of GL commands that have been stored for subsequent execution. Display lists are created with **glNewList**. All subsequent commands are placed in the display list, in the order issued, until **glEndList** is called.

glNewList has two arguments. The first argument, *list*, is a positive integer that becomes the unique name for the display list. Names can be created and reserved with **glGenLists** and tested for uniqueness with **glIsList**. The second argument, *mode*, is a symbolic constant that can assume one of two values:

GL_COMPILE   Commands are merely compiled.

GL_COMPILE_AND_EXECUTE

Commands are executed as they are compiled into the display list.

Certain commands are not compiled into the display list but are executed immediately, regardless of the display-list mode. These commands are **glColorPointer**, **glDeleteLists**, **glDisableClientState**, **glEdgeFlagPointer**, **glEnableClientState**, **glFeedbackBuffer**, **glFinish**, **glFlush**, **glGenLists**, **glIndexPointer**, **glInterleavedArrays**, **glIsEnabled**, **glIsList**, **glNormalPointer**, **glPopClientAttrib**, **glPixelStore**, **glPushClientAttrib**, **glReadPixels**, **glRenderMode**, **glSelectBuffer**, **glTexCoordPointer**, **glVertexPointer**, and all of the **glGet** commands.

Similarly, **glTexImage2D** and **glTexImage1D** are executed immediately and not compiled into the display list when their first argument is **GL_PROXY_TEXTURE_2D** or **GL_PROXY_TEXTURE_1D**, respectively.

When **glEndList** is encountered, the display-list definition is completed by associating the list with the unique name *list* (specified in the **glNewList** command). If a display list with name *list* already exists, it is replaced only when **glEndList** is called.

NOTES

glCallList and **glCallLists** can be entered into display lists. Commands in the display list or lists executed by **glCallList** or **glCallLists** are not included in the display list being created, even if the list creation mode is **GL_COMPILE_AND_EXECUTE**.

A display list is just a group of commands and arguments, so errors generated by commands in a display list must be generated when the list is executed. If the list is created in **GL_COMPILE** mode, errors are not generated until the list is executed.

ERRORS

GL_INVALID_VALUE is generated if *list* is 0.

GL_INVALID_ENUM is generated if *mode* is not an accepted value.

GL_INVALID_OPERATION is generated if **glEndList** is called without a preceding **glNewList**, or if **glNewList** is called while a display list is being defined.

GL_INVALID_OPERATION is generated if **glNewList** or **glEndList** is executed between the execution of **glBegin** and the corresponding execution of **glEnd**.

GL_OUT_OF_MEMORY is generated if there is insufficient memory to compile the display list. If the GL version is 1.1 or greater, no change is made to the previous contents of the display list, if any, and no other change is made to the GL state. (It is as if no attempt had been made to create the new display list.)

ASSOCIATED GETS
>**glIsList**
>**glGet** with argument **GL_LIST_INDEX**
>**glGet** with argument **GL_LIST_MODE**

SEE ALSO
>**glCallList**, **glCallLists**, **glDeleteLists**, **glGenLists**