

**NAME**

XInitThreads, XLockDisplay, XUnlockDisplay - multi-threading support

**SYNTAX**

Status XInitThreads(void);

Status XFreeThreads(void);

void XLockDisplay(Display \**display*);

void XUnlockDisplay(Display \**display*);

**ARGUMENTS**

*display*            Specifies the connection to the X server.

**DESCRIPTION**

The **XInitThreads** function initializes Xlib support for concurrent threads. This function must be the first Xlib function a multi-threaded program calls, and it must complete before any other Xlib call is made. This function returns a nonzero status if initialization was successful; otherwise, it returns zero. On systems that do not support threads, this function always returns zero.

It is only necessary to call this function if multiple threads might use Xlib concurrently. If all calls to Xlib functions are protected by some other access mechanism (for example, a mutual exclusion lock in a toolkit or through explicit client programming), Xlib thread initialization is not required. It is recommended that single-threaded programs not call this function.

The **XFreeThreads** function frees the memory allocated by **XInitThreads**.

The **XLockDisplay** function locks out all other threads from using the specified display. Other threads attempting to use the display will block until the display is unlocked by this thread. Nested calls to **XLockDisplay** work correctly; the display will not actually be unlocked until **XUnlockDisplay** has been called the same number of times as **XLockDisplay**. This function has no effect unless Xlib was successfully initialized for threads using **XInitThreads**.

The **XUnlockDisplay** function allows other threads to use the specified display again. Any threads that have blocked on the display are allowed to continue. Nested locking works correctly; if **XLockDisplay** has been called multiple times by a thread, then **XUnlockDisplay** must be called an equal number of times before the display is actually unlocked. This function has no effect unless Xlib was successfully initialized for threads using **XInitThreads**.

**SEE ALSO**

*Xlib - C Language X Interface*