

**NAME**

XIfEvent, XCheckIfEvent, XPeekIfEvent - check the event queue with a predicate procedure

**SYNTAX**

```
int XIfEvent(Display *display, XEvent *event_return, Bool (*predicate)(), XPointer arg);
```

```
Bool XCheckIfEvent(Display *display, XEvent *event_return, Bool (*predicate)(), XPointer arg);
```

```
int XPeekIfEvent(Display *display, XEvent *event_return, Bool (*predicate)(), XPointer arg);
```

**ARGUMENTS**

- |                     |   |
|---------------------|---|
| <i>arg</i>          | Specifies the user-supplied argument that will be passed to the predicate procedure.                            |
| <i>display</i>      | Specifies the connection to the X server.   |
| <i>event_return</i> | Returns either a copy of or the matched event's associated structure.   |
| <i>predicate</i>    | Specifies the procedure that is to be called to determine if the next event in the queue matches what you want. |

**DESCRIPTION**

The **XIfEvent** function completes only when the specified predicate procedure returns **True** for an event, which indicates an event in the queue matches. **XIfEvent** flushes the output buffer if it blocks waiting for additional events. **XIfEvent** removes the matching event from the queue and copies the structure into the client-supplied **XEvent** structure.

When the predicate procedure finds a match, **XCheckIfEvent** copies the matched event into the client-supplied **XEvent** structure and returns **True**. (This event is removed from the queue.) If the predicate procedure finds no match, **XCheckIfEvent** returns **False**, and the output buffer will have been flushed. All earlier events stored in the queue are not discarded.

The **XPeekIfEvent** function returns only when the specified predicate procedure returns **True** for an event. After the predicate procedure finds a match, **XPeekIfEvent** copies the matched event into the client-supplied **XEvent** structure without removing the event from the queue. **XPeekIfEvent** flushes the output buffer if it blocks waiting for additional events.

Each of these functions requires you to pass a predicate procedure that determines if an event matches what you want. Your predicate procedure must decide if the event is useful without calling any Xlib functions. If the predicate directly or indirectly causes the state of the event queue to change, the result is not defined. If Xlib has been initialized for threads, the predicate is called with the display locked

and the result of a call by the predicate to any Xlib function that locks the display is not defined unless the caller has first called **XLockDisplay**.

The predicate procedure and its associated arguments are:

Bool (*\*predicate*)(Display *\*display*, XEvent *\*event*, XPointer *arg*)

*display*        Specifies the connection to the X server.

*event*         Specifies the **XEvent** structure.

*arg*            Specifies the argument passed in from the **XIfEvent**, **XCheckIfEvent**, or **XPeekIfEvent** function.

The predicate procedure is called once for each event in the queue until it finds a match. After finding a match, the predicate procedure must return **True**. If it did not find a match, it must return **False**.

Note that the predicate procedure is called with the display locked, and thus is not allowed to call any function that would take the display lock again.

#### **SEE ALSO**

XAnyEvent(3), XNextEvent(3), XPutBackEvent(3) XSendEvent(3)

*Xlib - C Language X Interface*