**XNextEvent, XPeekEvent, XWindowEvent, XCheckWindowEvent, XMaskEvent, XCheckMaskEvent, XCheckTypedEvent, XCheckTypedWindowEvent** – select events by type

**XNextEvent**(*display, event_return*)

Display *display;
XEvent *event_return;

**XPeekEvent**(*display, event_return*)

Display *display;
XEvent *event_return;

**XWindowEvent**(*display, w, event_mask, event_return*)

Display *display;
Window *w;
long event_mask;
XEvent *event_return;

**Bool XCheckWindowEvent**(*display, w, event_mask, event_return*)

Display *display;
Window *w;
long event_mask;
XEvent *event_return;

**XMaskEvent**(*display, event_mask, event_return*)

Display *display;
long event_mask;
XEvent *event_return;

**Bool XCheckMaskEvent**(*display, event_mask, event_return*)

Display *display;
long event_mask;
XEvent *event_return;

**Bool XCheckTypedEvent**(*display, event_type, event_return*)

Display *display;
int event_type;
XEvent *event_return;

**Bool XCheckTypedWindowEvent**(*display, w, event_type, event_return*)

Display *display;
Window *w;
int event_type;
XEvent *event_return;

- `display` specifies the connection to the X server.
- `event_mask` specifies the event mask.
- `event_return` returns the matched event’s associated structure.
- `event_return` returns the next event in the queue.
- `event_return` returns a copy of the matched event’s associated structure.
- `event_type` specifies the event type to be compared.
- `w` specifies the window whose events you are interested in.

The **XNextEvent** function copies the first event from the event queue into the specified **XEvent** structure and then removes it from the queue. If the event queue is empty, **XNextEvent** flushes the output buffer and blocks until an event is received.
The **XPeekEvent** function returns the first event from the event queue, but it does not remove the event from the queue. If the queue is empty, **XPeekEvent** flushes the output buffer and blocks until an event is received. It then copies the event into the client-supplied **XEvent** structure without removing it from the event queue.

The **XWindowEvent** function searches the event queue for an event that matches both the specified window and event mask. When it finds a match, **XWindowEvent** removes that event from the queue and copies it into the specified **XEvent** structure. The other events stored in the queue are not discarded. If a matching event is not in the queue, **XWindowEvent** flushes the output buffer and blocks until one is received.

The **XCheckWindowEvent** function searches the event queue and then the events available on the server connection for the first event that matches the specified window and event mask. If it finds a match, **XCheckWindowEvent** removes that event, copies it into the specified **XEvent** structure, and returns `True`. The other events stored in the queue are not discarded. If the event you requested is not available, **XCheckWindowEvent** returns `False`, and the output buffer will have been flushed.

The **XMaskEvent** function searches the event queue for the events associated with the specified mask. When it finds a match, **XMaskEvent** removes that event and copies it into the specified **XEvent** structure. The other events stored in the queue are not discarded. If the event you requested is not in the queue, **XMaskEvent** flushes the output buffer and blocks until one is received.

The **XCheckMaskEvent** function searches the event queue and then any events available on the server connection for the first event that matches the specified mask. If it finds a match, **XCheckMaskEvent** removes that event, copies it into the specified **XEvent** structure, and returns `True`. The other events stored in the queue are not discarded. If the event you requested is not available, **XCheckMaskEvent** returns `False`, and the output buffer will have been flushed.

The **XCheckTypedEvent** function searches the event queue and then any events available on the server connection for the first event that matches the specified type. If it finds a match, **XCheckTypedEvent** removes that event, copies it into the specified **XEvent** structure, and returns `True`. The other events in the queue are not discarded. If the event is not available, **XCheckTypedEvent** returns `False`, and the output buffer will have been flushed.

The **XCheckTypedWindowEvent** function searches the event queue and then any events available on the server connection for the first event that matches the specified type and window. If it finds a match, **XCheckTypedWindowEvent** removes the event from the queue, copies it into the specified **XEvent** structure, and returns `True`. The other events in the queue are not discarded. If the event is not available, **XCheckTypedWindowEvent** returns `False`, and the output buffer will have been flushed.

XAnyEvent(3X11), XIfEvent(3X11), XPutBackEvent(3X11), XSendEvent(3X11)

*Xlib – C Language X Interface*