XButtonEvent, XKeyEvent, XMotionEvent – KeyPress, KeyRelease, ButtonPress, ButtonRelease, and MotionNotify event structures

The structures for KeyPress, **KeyRelease**, **ButtonPress**, **ButtonRelease**, and **MotionNotify** events contain:

```
typedef struct {
        int type;                       /* ButtonPress or ButtonRelease */
        unsigned long serial;           /* # of last request processed by server */
        Bool send_event;                /* true if this came from a SendEvent request */
        Display *display;               /* Display the event was read from */
        Window window;                  /* ‘‘event’’ window it is reported relative to */
        Window root;                    /* root window that the event occurred on */
        Window subwindow;               /* child window */
        Time time;                      /* milliseconds */
        int x, y;                       /* pointer x, y coordinates in event window */
        int x_root, y_root;             /* coordinates relative to root */
        unsigned int state;             /* key or button mask */
        unsigned int button;            /* detail */
        Bool same_screen;               /* same screen flag */
} XButtonEvent;
typedef XButtonEvent XButtonPressedEvent;
typedef XButtonEvent XButtonReleasedEvent;

typedef struct {
        int type;                       /* KeyPress or KeyRelease */
        unsigned long serial;           /* # of last request processed by server */
        Bool send_event;                /* true if this came from a SendEvent request */
        Display *display;               /* Display the event was read from */
        Window window;                  /* ‘‘event’’ window it is reported relative to */
        Window root;                    /* root window that the event occurred on */
        Window subwindow;               /* child window */
        Time time;                      /* milliseconds */
        int x, y;                       /* pointer x, y coordinates in event window */
        int x_root, y_root;             /* coordinates relative to root */
        unsigned int state;             /* key or button mask */
        unsigned int keycode;           /* detail */
        Bool same_screen;               /* same screen flag */
} XKeyEvent;
typedef XKeyEvent XKeyPressedEvent;
typedef XKeyEvent XKeyReleasedEvent;

typedef struct {
        int type;                       /* MotionNotify */
        unsigned long serial;           /* # of last request processed by server */
        Bool send_event;                /* true if this came from a SendEvent request */
        Display *display;               /* Display the event was read from */
        Window window;                  /* ‘‘event’’ window reported relative to */
        Window root;                    /* root window that the event occurred on */
        Window subwindow;               /* child window */
        Time time;                      /* milliseconds */
        int x, y;                       /* pointer x, y coordinates in event window */
        int x_root, y_root;             /* coordinates relative to root */
        unsigned int state;             /* key or button mask */
        char is_hint;                   /* detail */
        Bool same_screen;               /* same screen flag */
} XMotionEvent;
typedef XMotionEvent XPointerMovedEvent;
```

When you receive these events, their structure members are set as follows.

The type member is set to the event type constant name that uniquely identifies it. For example, when the X server reports a **GraphicsExpose** event to a client application, it sends an **XGraphicsExposeEvent** structure with the type member set to **GraphicsExpose**. The display member is set to a pointer to the display the event was read on. The send_event member is set to **True** if the event came from a **SendEvent** protocol request. The serial member is set from the serial number reported in the protocol but expanded from the 16-bit least-significant bits to a full 32-bit value. The window member is set to the window that is most useful to toolkit dispatchers.

These structures have the following common members: window, root, subwindow, time, x, y, x_root, y_root, state, and same_screen. The window member is set to the window on which the event was generated and is referred to as the event window. As long as the conditions previously discussed are met, this is the window used by the X server to report the event. The root member is set to the source window's root window. The x_root and y_root members are set to the pointer's coordinates relative to the root window's origin at the time of the event.

The same_screen member is set to indicate whether the event window is on the same screen as the root window and can be either **True** or **False**. If **True**, the event and root windows are on the same screen. If **False**, the event and root windows are not on the same screen.

If the source window is an inferior of the event window, the subwindow member of the structure is set to the child of the event window that is the source window or the child of the event window that is an ancestor of the source window. Otherwise, the X server sets the subwindow member to **None**. The time member is set to the time when the event was generated and is expressed in milliseconds.

If the event window is on the same screen as the root window, the x and y members are set to the coordinates relative to the event window's origin. Otherwise, these members are set to zero.

The state member is set to indicate the logical state of the pointer buttons and modifier keys just prior to the event, which is the bitwise inclusive OR of one or more of the button or modifier key masks: **Button1Mask**, **Button2Mask**, **Button3Mask**, **Button4Mask**, **Button5Mask**, **ShiftMask**, **LockMask**, **ControlMask**, **Mod1Mask**, **Mod2Mask**, **Mod3Mask**, **Mod4Mask**, and **Mod5Mask**.

Each of these structures also has a member that indicates the detail. For the **XKeyPressedEvent** and **XKeyReleasedEvent** structures, this member is called a keycode. It is set to a number that represents a physical key on the keyboard. The keycode is an arbitrary representation for any key on the keyboard (see sections 12.7 and 16.1).

For the **XButtonPressedEvent** and **XButtonReleasedEvent** structures, this member is called button. It represents the pointer button that changed state and can be the **Button1**, **Button2**, **Button3**, **Button4**, or **Button5** value. For the **XPointerMovedEvent** structure, this member is called is_hint. It can be set to **NotifyNormal** or **NotifyHint**.

XAnyEvent(3X11), XCreateWindowEvent(3X11), XCirculateEvent(3X11), XCirculateRequestEvent(3X11), XColormapEvent(3X11), XConfigureEvent(3X11), XConfigureRequestEvent(3X11), XCrossingEvent(3X11), XDestroyWindowEvent(3X11), XErrorEvent(3X11), XExposeEvent(3X11), XFocusChangeEvent(3X11), XGraphicsExposeEvent(3X11), XGravityEvent(3X11), XKeymapEvent(3X11), XMapEvent(3X11), XMapRequestEvent(3X11), XPropertyEvent(3X11), XReparentEvent(3X11), XResizeRequestEvent(3X11), XSelectionClearEvent(3X11), XSelectionEvent(3X11), XSelectionRequestEvent(3X11), XUnmapEvent(3X11), XVisibilityEvent(3X11)
Xlib – C Language X Interface