NAME

XCrossingEvent - EnterNotify and LeaveNotify event structure

STRUCTURES

The structure for **EnterNotify** and **LeaveNotify** events contains:

```
typedef struct {
    int type;       /* EnterNotify or LeaveNotify */
    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 */
    int mode;       /* NotifyNormal, NotifyGrab, NotifyUngrab */
    int detail;
        /*
        * NotifyAncestor, NotifyVirtual, NotifyInferior,
        * NotifyNonlinear,NotifyNonlinearVirtual
        */
    Bool same_screen;       /* same screen flag */
    Bool focus;     /* boolean focus */
    unsigned int state;     /* key or button mask */
} XCrossingEvent;
typedef XCrossingEvent XEnterWindowEvent;
typedef XCrossingEvent XLeaveWindowEvent;
```

When you receive these events, the 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.

The window member is set to the window on which the **EnterNotify** or **LeaveNotify** event was generated and is referred to as the event window. This is the window used by the X server to report the event, and is relative to the root window on which the event occurred. The root member is set to the root window of the screen on which the event occurred.

For a **LeaveNotify** event, if a child of the event window contains the initial position of the pointer, the subwindow component is set to that child. Otherwise, the X server sets the subwindow member to **None**. For an **EnterNotify** event, if a child of the event window contains the final pointer position, the subwindow component is set to that child or **None**.

The time member is set to the time when the event was generated and is expressed in milliseconds. The x and y members are set to the coordinates of the pointer position in the event window. This position is always the pointer's final position, not its initial position. If the event window is on the same screen as the root window, x and y are the pointer coordinates relative to the event window's origin. Otherwise, x and y are set to zero. 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.

The focus member is set to indicate whether the event window is the focus window or an inferior of the focus window. The X server can set this member to either **True** or **False**. If **True**, the event window is the focus window or an inferior of the focus window. If **False**, the event window is not the focus window or an inferior of the focus window.

The state member is set to indicate the state of the pointer buttons and modifier keys just prior to the event. The X server can set this member to 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**, **Mod5Mask**.

The mode member is set to indicate whether the events are normal events, pseudo-motion events when a grab activates, or pseudo-motion events when a grab deactivates. The X server can set this member to **NotifyNormal**, **NotifyGrab**, or **NotifyUngrab**.

The detail member is set to indicate the notify detail and can be **NotifyAncestor**, **NotifyVirtual**, **NotifyInferior**, **NotifyNonlinear**, or **NotifyNonlinearVirtual**.

SEE ALSO

XAnyEvent(3), XButtonEvent(3), XCreateWindowEvent(3), XCirculateEvent(3),

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