XFocusChangeEvent – FocusIn and FocusOut event structure

The structure for FocusIn and **FocusOut** events contains:

```
typedef struct {
        int type;                       /* FocusIn or FocusOut */
        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;                  /* window of event */
        int mode;                       /* NotifyNormal, NotifyGrab, NotifyUngrab */
        int detail;
                                        /*
                                        * NotifyAncestor, NotifyVirtual, NotifyInferior,
                                        * NotifyNonlinear,NotifyNonlinearVirtual, NotifyPointer,
                                        * NotifyPointerRoot, NotifyDetailNone
                                        */
} XFocusChangeEvent;
typedef XFocusChangeEvent XFocusInEvent;
typedef XFocusChangeEvent XFocusOutEvent;
```

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 **FocusIn** or **FocusOut** event was generated. This is the window used by the X server to report the event. The mode member is set to indicate whether the focus events are normal focus events, focus events while grabbed, focus events when a grab activates, or focus events when a grab deactivates. The X server can set the mode member to **NotifyNormal**, **NotifyWhileGrabbed**, **NotifyGrab**, or **NotifyUngrab**.

All **FocusOut** events caused by a window unmap are generated after any **UnmapNotify** event; however, the X protocol does not constrain the ordering of **FocusOut** events with respect to generated **EnterNotify**, **LeaveNotify**, **VisibilityNotify**, and **Expose** events.

Depending on the event mode, the detail member is set to indicate the notify detail and can be **NotifyAncestor**, **NotifyVirtual**, **NotifyInferior**, **NotifyNonlinear**, **NotifyNonlinearVirtual**, **NotifyPointer**, **NotifyPointerRoot**, or **NotifyDetailNone**.

XAnyEvent(3X11), XButtonEvent(3X11), XCreateWindowEvent(3X11), XCirculateEvent(3X11), XCirculateRequestEvent(3X11), XColormapEvent(3X11), XConfigureEvent(3X11), XConfigureRequestEvent(3X11), XCrossingEvent(3X11), XDestroyWindowEvent(3X11), XErrorEvent(3X11), XExposeEvent(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