

NAME

XMapEvent, XMappingEvent - MapNotify and MappingNotify event structures

STRUCTURES

The structure for **MapNotify** events contains:

```
typedef struct {
    int type; /* MapNotify */
    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 event;
    Window window;
    Bool override_redirect; /* boolean, is override set... */
} XMapEvent;
```

When you receive this event, 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 event member is set either to the window that was mapped or to its parent, depending on whether **StructureNotify** or **SubstructureNotify** was selected. The window member is set to the window that was mapped. The override_redirect member is set to the override-redirect attribute of the window. Window manager clients normally should ignore this window if the override-redirect attribute is **True**, because these events usually are generated from pop-ups, which override structure control.

The structure for **MappingNotify** events is:

```
typedef struct {
    int type; /* MappingNotify */
    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; /* unused */
}
```

```

    int request; /* one of MappingModifier, MappingKeyboard,
                 MappingPointer */
    int first_keycode; /* first keycode */
    int count; /* defines range of change with first_keycode*/
} XMappingEvent;

```

When you receive this event, 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 request member is set to indicate the kind of mapping change that occurred and can be **MappingModifier**, **MappingKeyboard**, **MappingPointer**. If it is **MappingModifier**, the modifier mapping was changed. If it is **MappingKeyboard**, the keyboard mapping was changed. If it is **MappingPointer**, the pointer button mapping was changed. The first_keycode and count members are set only if the request member was set to **MappingKeyboard**. The number in first_keycode represents the first number in the range of the altered mapping, and count represents the number of keycodes altered.

SEE ALSO

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