

## XGraphicsExposeEvent, XNoExposeEvent – GraphicsExpose and NoExpose event structures

The structures for **GraphicsExpose** and **NoExpose** events contain:

```
typedef struct {
    int type; /* GraphicsExpose */
    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 */
    Drawable drawable;
    int x, y;
    int width, height;
    int count; /* if nonzero, at least this many more */
    int major_code; /* core is CopyArea or CopyPlane */
    int minor_code; /* not defined in the core */
} XGraphicsExposeEvent;

typedef struct {
    int type; /* NoExpose */
    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 */
    Drawable drawable;
    int major_code; /* core is CopyArea or CopyPlane */
    int minor_code; /* not defined in the core */
} XNoExposeEvent;
```

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.

Both structures have these common members: `drawable`, `major_code`, and `minor_code`. The `drawable` member is set to the drawable of the destination region on which the graphics request was to be performed. The `major_code` member is set to the graphics request initiated by the client and can be either **X\_CopyArea** or **X\_CopyPlane**. If it is **X\_CopyArea**, a call to **XCopyArea** initiated the request. If it is **X\_CopyPlane**, a call to **XCopyPlane** initiated the request. These constants are defined in `<X11/Xproto.h>`. The `minor_code` member, like the `major_code` member, indicates which graphics request was initiated by the client. However, the `minor_code` member is not defined by the core X protocol and will be zero in these cases, although it may be used by an extension.

The **XGraphicsExposeEvent** structure has these additional members: `x`, `y`, `width`, `height`, and `count`. The `x` and `y` members are set to the coordinates relative to the drawable's origin and indicate the upper-left corner of the rectangle. The `width` and `height` members are set to the size (extent) of the rectangle. The `count` member is set to the number of **GraphicsExpose** events to follow. If `count` is zero, no more **GraphicsExpose** events follow for this window. However, if `count` is nonzero, at least that number of **GraphicsExpose** events (and possibly more) are to follow for this window.

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