NAME

XCreateWindow, XCreateSimpleWindow, XSetWindowAttributes – create windows and window attributes structure

SYNTAX

Window XCreateWindow(Display **display*, Window *parent*, int *x*, int *y*, unsigned int *width*, unsigned int *height*, unsigned int *border_width*, int *depth*, unsigned int *class*, Visual **visual*, unsigned long *valuemask*, XSetWindowAttributes **attributes*);

Window XCreateSimpleWindow(Display **display*, Window *parent*, int *x*, int *y*, unsigned int *width*, unsigned int *height*, unsigned int *border_width*, unsigned long *border*, unsigned long *background*);

ARGUMENTS

attributes Specifies the structure from which the values (as specified by the value mask) are to be taken. The value mask should have the appropriate bits set to indicate which attributes have been set in the structure.

background Specifies the background pixel value of the window.

border Specifies the border pixel value of the window.

border_width Specifies the width of the created window's border in pixels.

class Specifies the created window's class. You can pass **InputOutput**, **InputOnly**, or **CopyFromParent**. A class of **CopyFromParent** means the class is taken from the parent.

depth Specifies the window's depth. A depth of **CopyFromParent** means the depth is taken from the parent.

display Specifies the connection to the X server.

parent Specifies the parent window.

valuemask Specifies which window attributes are defined in the attributes argument. This mask is the bitwise inclusive OR of the valid attribute mask bits. If valuemask is zero, the attributes are ignored and are not referenced.

visual Specifies the visual type. A visual of **CopyFromParent** means the visual type is taken from the parent.

width
height Specify the width and height, which are the created window's inside dimensions and do not include the created window's borders.

x
y Specify the x and y coordinates, which are the top-left outside corner of the window's borders and are relative to the inside of the parent window's borders.

DESCRIPTION

The **XCreateWindow** function creates an unmapped subwindow for a specified parent window, returns the window ID of the created window, and causes the X server to generate a **CreateNotify** event. The created window is placed on top in the stacking order with respect to siblings.

The coordinate system has the X axis horizontal and the Y axis vertical with the origin [0, 0] at the upper-left corner. Coordinates are integral, in terms of pixels, and coincide with pixel centers. Each window and pixmap has its own coordinate system. For a window, the origin is inside the border at the inside, upper-left corner.

The border_width for an **InputOnly** window must be zero, or a **BadMatch** error results. For class **InputOutput**, the visual type and depth must be a combination supported for the screen, or a **BadMatch** error results. The depth need not be the same as the parent, but the parent must not be a window of class **InputOnly**, or a **BadMatch** error results. For an **InputOnly** window, the depth must be zero, and the visual

must be one supported by the screen. If either condition is not met, a **BadMatch** error results. The parent window, however, may have any depth and class. If you specify any invalid window attribute for a window, a **BadMatch** error results.

The created window is not yet displayed (mapped) on the user's display. To display the window, call **XMapWindow**. The new window initially uses the same cursor as its parent. A new cursor can be defined for the new window by calling **XDefineCursor**. The window will not be visible on the screen unless it and all of its ancestors are mapped and it is not obscured by any of its ancestors.

XCreateWindow can generate **BadAlloc BadColor**, **BadCursor**, **BadMatch**, **BadPixmap**, **BadValue**, and **BadWindow** errors.

The **XCreateSimpleWindow** function creates an unmapped **InputOutput** subwindow for a specified parent window, returns the window ID of the created window, and causes the X server to generate a **CreateNotify** event. The created window is placed on top in the stacking order with respect to siblings. Any part of the window that extends outside its parent window is clipped. The border_width for an **InputOnly** window must be zero, or a **BadMatch** error results. **XCreateSimpleWindow** inherits its depth, class, and visual from its parent. All other window attributes, except background and border, have their default values.

XCreateSimpleWindow can generate **BadAlloc**, **BadMatch**, **BadValue**, and **BadWindow** errors.

STRUCTURES

The **XSetWindow**Attributes structure contains:

/* Window attribute value mask bits */

#define	**CWBackPixmap**	(1L<<0)
#define	**CWBackPixel**	(1L<<1)
#define	**CWBorderPixmap**	(1L<<2)
#define	**CWBorderPixel**	(1L<<3)
#define	**CWBitGravity**	(1L<<4)
#define	**CWWinGravity**	(1L<<5)
#define	**CWBackingStore**	(1L<<6)
#define	**CWBackingPlanes**	(1L<<7)
#define	**CWBackingPixel**	(1L<<8)
#define	**CWOverrideRedirect**	(1L<<9)
#define	**CWSaveUnder**	(1L<<10)
#define	**CWEventMask**	(1L<<11)
#define	**CWDontPropagate**	(1L<<12)
#define	**CWColormap**	(1L<<13)
#define	**CWCursor**	(1L<<14)

```
/* Values */

typedef struct {
        Pixmap background_pixmap;       /* background, None, or ParentRelative */
        unsigned long background_pixel; /* background pixel */
        Pixmap border_pixmap;           /* border of the window or CopyFromParent */
        unsigned long border_pixel;     /* border pixel value */
        int bit_gravity;        /* one of bit gravity values */
        int win_gravity;        /* one of the window gravity values */
        int backing_store;      /* NotUseful, WhenMapped, Always */
        unsigned long backing_planes;   /* planes to be preserved if possible */
        unsigned long backing_pixel;    /* value to use in restoring planes */
        Bool save_under;        /* should bits under be saved? (popups) */
        long event_mask;        /* set of events that should be saved */
        long do_not_propagate_mask;     /* set of events that should not propagate *
        Bool override_redirect; /* boolean value for override_redirect */
        Colormap colormap;      /* color map to be associated with window */
        Cursor cursor;          /* cursor to be displayed (or None) */
```

```
} XSetWindowAttributes;
```

For a detailed explanation of the members of this structure, see *Xlib − C Language X Interface*.

DIAGNOSTICS

BadAlloc The server failed to allocate the requested resource or server memory.

BadColor A value for a Colormap argument does not name a defined Colormap.

BadCursor A value for a Cursor argument does not name a defined Cursor.

BadMatch The values do not exist for an **InputOnly** window.

BadMatch Some argument or pair of arguments has the correct type and range but fails to match in some other way required by the request.

BadPixmap A value for a Pixmap argument does not name a defined Pixmap.

BadValue Some numeric value falls outside the range of values accepted by the request. Unless a specific range is specified for an argument, the full range defined by the argument's type is accepted. Any argument defined as a set of alternatives can generate this error.

BadWindow A value for a Window argument does not name a defined Window.

SEE ALSO

XChangeWindowAttributes(3), XConfigureWindow(3), XDefineCursor(3), XDestroyWindow(3), XMapWindow(3), XRaiseWindow(3), XUnmapWindow(3)
Xlib − C Language X Interface