

**NAME**

XRaiseWindow, XLowerWindow, XCirculateSubwindows, XCirculateSubwindowsUp, XCirculateSubwindowsDown, XRestackWindows - change window stacking order

**SYNTAX**

```
int XRaiseWindow(Display *display, Window w);
```

```
int XLowerWindow(Display *display, Window w);
```

```
int XCirculateSubwindows(Display *display, Window w, int direction);
```

```
int XCirculateSubwindowsUp(Display *display, Window w);
```

```
int XCirculateSubwindowsDown(Display *display, Window w);
```

```
int XRestackWindows(Display *display, Window windows[], int nwindows);
```

**ARGUMENTS**

*direction* Specifies the direction (up or down) that you want to circulate the window. You can pass **RaiseLowest** or **LowerHighest**.

*display* Specifies the connection to the X server.

*nwindows* Specifies the number of windows to be restacked.

*w* Specifies the window.

*windows* Specifies an array containing the windows to be restacked.

**DESCRIPTION**

The **XRaiseWindow** function raises the specified window to the top of the stack so that no sibling window obscures it. If the windows are regarded as overlapping sheets of paper stacked on a desk, then raising a window is analogous to moving the sheet to the top of the stack but leaving its x and y location on the desk constant. Raising a mapped window may generate **Expose** events for the window and any mapped subwindows that were formerly obscured.

If the override-redirect attribute of the window is **False** and some other client has selected **SubstructureRedirectMask** on the parent, the X server generates a **ConfigureRequest** event, and no processing is performed. Otherwise, the window is raised.

**XRaiseWindow** can generate a **BadWindow** error.

The **XLowerWindow** function lowers the specified window to the bottom of the stack so that it does not obscure any sibling windows. If the windows are regarded as overlapping sheets of paper stacked on a desk, then lowering a window is analogous to moving the sheet to the bottom of the stack but leaving its x and y location on the desk constant. Lowering a mapped window will generate **Expose** events on any windows it formerly obscured.

If the `override-redirect` attribute of the window is **False** and some other client has selected **SubstructureRedirectMask** on the parent, the X server generates a **ConfigureRequest** event, and no processing is performed. Otherwise, the window is lowered to the bottom of the stack.

**XLowerWindow** can generate a **BadWindow** error.

The **XCirculateSubwindows** function circulates children of the specified window in the specified direction. If you specify **RaiseLowest**, **XCirculateSubwindows** raises the lowest mapped child (if any) that is occluded by another child to the top of the stack. If you specify **LowerHighest**, **XCirculateSubwindows** lowers the highest mapped child (if any) that occludes another child to the bottom of the stack. Exposure processing is then performed on formerly obscured windows. If some other client has selected **SubstructureRedirectMask** on the window, the X server generates a **CirculateRequest** event, and no further processing is performed. If a child is actually restacked, the X server generates a **CirculateNotify** event.

**XCirculateSubwindows** can generate **BadValue** and **BadWindow** errors.

The **XCirculateSubwindowsUp** function raises the lowest mapped child of the specified window that is partially or completely occluded by another child. Completely unobscured children are not affected. This is a convenience function equivalent to **XCirculateSubwindows** with **RaiseLowest** specified.

**XCirculateSubwindowsUp** can generate a **BadWindow** error.

The **XCirculateSubwindowsDown** function lowers the highest mapped child of the specified window that partially or completely occludes another child. Completely unobscured children are not affected. This is a convenience function equivalent to **XCirculateSubwindows** with **LowerHighest** specified.

**XCirculateSubwindowsDown** can generate a **BadWindow** error.

The **XRestackWindows** function restacks the windows in the order specified, from top to bottom. The stacking order of the first window in the windows array is unaffected, but the other windows in the array are stacked underneath the first window, in the order of the array. The stacking order of the other

windows is not affected. For each window in the window array that is not a sibling of the first window, a **BadMatch** error results.

If the `override-redirect` attribute of a window is **False** and some other client has selected **SubstructureRedirectMask** on the parent, the X server generates **ConfigureRequest** events for each window whose `override-redirect` flag is not set, and no further processing is performed. Otherwise, the windows will be restacked in top-to-bottom order.

**XRestackWindows** can generate **BadWindow** error.

## DIAGNOSTICS

**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)`, `XCreateWindow(3)`, `XDestroyWindow(3)`,  
`XMapWindow(3)`, `XUnmapWindow(3)`

*Xlib - C Language X Interface*