

**NAME**

XWarpPointer - move pointer

**SYNTAX**

```
int XWarpPointer(Display *display, Window src_w, Window dest_w, int src_x, int src_y, unsigned int
src_width, unsigned int src_height, int dest_x, int dest_y);
```

**ARGUMENTS**

*dest\_w* Specifies the destination window or **None**.

*dest\_x*

*dest\_y* Specify the x and y coordinates within the destination window.

*display* Specifies the connection to the X server.

*src\_x*

*src\_y*

*src\_width*

*src\_height* Specify a rectangle in the source window.

*src\_w* Specifies the source window or **None**.

**DESCRIPTION**

If *dest\_w* is **None**, **XWarpPointer** moves the pointer by the offsets (*dest\_x*, *dest\_y*) relative to the current position of the pointer. If *dest\_w* is a window, **XWarpPointer** moves the pointer to the offsets (*dest\_x*, *dest\_y*) relative to the origin of *dest\_w*. However, if *src\_w* is a window, the move only takes place if the window *src\_w* contains the pointer and if the specified rectangle of *src\_w* contains the pointer.

The *src\_x* and *src\_y* coordinates are relative to the origin of *src\_w*. If *src\_height* is zero, it is replaced with the current height of *src\_w* minus *src\_y*. If *src\_width* is zero, it is replaced with the current width of *src\_w* minus *src\_x*.

There is seldom any reason for calling this function. The pointer should normally be left to the user. If you do use this function, however, it generates events just as if the user had instantaneously moved the pointer from one position to another. Note that you cannot use **XWarpPointer** to move the pointer

outside the `confine_to` window of an active pointer grab. An attempt to do so will only move the pointer as far as the closest edge of the `confine_to` window.

**XWarpPointer** can generate a **BadWindow** error.

## DIAGNOSTICS

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

## SEE ALSO

XSetInputFocus(3)

*Xlib - C Language X Interface*