

## XPutImage, XGetImage, XGetSubImage – transfer images

**XPutImage**(*display, d, gc, image, src\_x, src\_y, dest\_x, dest\_y, width, height*)

```
Display *display;  
Drawable d;  
GC gc;  
XImage *image;  
int src_x, src_y;  
int dest_x, dest_y;  
unsigned int width, height;
```

XImage \*XGetImage(*display, d, x, y, width, height, plane\_mask, format*)

```
Display *display;  
Drawable d;  
int x, y;  
unsigned int width, height;  
unsigned long plane_mask;  
int format;
```

XImage \*XGetSubImage(*display, d, x, y, width, height, plane\_mask, format, dest\_image, dest\_x, dest\_y*)

```
Display *display;  
Drawable d;  
int x, y;  
unsigned int width, height;  
unsigned long plane_mask;  
int format;  
XImage *dest_image;  
int dest_x, dest_y;
```

<i>d</i>	Specifies the drawable.
<i>dest_image</i>	Specifies the destination image.
<i>dest_x</i> <i>dest_y</i>	Specify the x and y coordinates, which are relative to the origin of the drawable and are the coordinates of the subimage or which are relative to the origin of the destination rectangle, specify its upper-left corner, and determine where the subimage is placed in the destination image.
<i>display</i>	Specifies the connection to the X server.
<i>format</i>	Specifies the format for the image. You can pass <b>XYPixmap</b> or <b>ZPixmap</b> .
<i>gc</i>	Specifies the GC.
<i>image</i>	Specifies the image you want combined with the rectangle.
<i>plane_mask</i>	Specifies the plane mask.
<i>src_x</i>	Specifies the offset in X from the left edge of the image defined by the <b>XImage</b> structure.
<i>src_y</i>	Specifies the offset in Y from the top edge of the image defined by the <b>XImage</b> structure.
<i>width</i> <i>height</i>	Specify the width and height of the subimage, which define the dimensions of the rectangle.
<i>x</i> <i>y</i>	Specify the x and y coordinates, which are relative to the origin of the drawable and define the upper-left corner of the rectangle.

The **XPutImage** function combines an image with a rectangle of the specified drawable. The section of the image defined by the `src_x`, `src_y`, `width`, and `height` arguments is drawn on the specified part of the drawable. If **XYBitmap** format is used, the depth of the image must be one, or a **BadMatch** error results. The foreground pixel in the GC defines the source for the one bits in the image, and the background pixel defines the source for the zero bits. For **XPixmap** and **ZPixmap**, the depth of the image must match the depth of the drawable, or a **BadMatch** error results.

If the characteristics of the image (for example, `byte_order` and `bitmap_unit`) differ from what the server requires, **XPutImage** automatically makes the appropriate conversions.

This function uses these GC components: `function`, `plane-mask`, `subwindow-mode`, `clip-x-origin`, `clip-y-origin`, and `clip-mask`. It also uses these GC mode-dependent components: `foreground` and `background`.

**XPutImage** can generate **BadDrawable**, **BadGC**, **BadMatch**, and **BadValue** errors.

The **XGetImage** function returns a pointer to an **XImage** structure. This structure provides you with the contents of the specified rectangle of the drawable in the format you specify. If the format argument is **XPixmap**, the image contains only the bit planes you passed to the `plane_mask` argument. If the `plane_mask` argument only requests a subset of the planes of the display, the depth of the returned image will be the number of planes requested. If the format argument is **ZPixmap**, **XGetImage** returns as zero the bits in all planes not specified in the `plane_mask` argument. The function performs no range checking on the values in `plane_mask` and ignores extraneous bits.

**XGetImage** returns the depth of the image to the `depth` member of the **XImage** structure. The depth of the image is as specified when the drawable was created, except when getting a subset of the planes in **XPixmap** format, when the depth is given by the number of bits set to 1 in `plane_mask`.

If the drawable is a pixmap, the given rectangle must be wholly contained within the pixmap, or a **BadMatch** error results. If the drawable is a window, the window must be viewable, and it must be the case that if there were no inferiors or overlapping windows, the specified rectangle of the window would be fully visible on the screen and wholly contained within the outside edges of the window, or a **BadMatch** error results. Note that the borders of the window can be included and read with this request. If the window has backing-store, the backing-store contents are returned for regions of the window that are obscured by noninferior windows. If the window does not have backing-store, the returned contents of such obscured regions are undefined. The returned contents of visible regions of inferiors of a different depth than the specified window's depth are also undefined. The pointer cursor image is not included in the returned contents. If a problem occurs, **XGetImage** returns NULL.

**XGetImage** can generate **BadDrawable**, **BadMatch**, and **BadValue** errors.

The **XGetSubImage** function updates `dest_image` with the specified subimage in the same manner as **XGetImage**. If the format argument is **XPixmap**, the image contains only the bit planes you passed to the `plane_mask` argument. If the format argument is **ZPixmap**, **XGetSubImage** returns as zero the bits in all planes not specified in the `plane_mask` argument. The function performs no range checking on the values in `plane_mask` and ignores extraneous bits. As a convenience, **XGetSubImage** returns a pointer to the same **XImage** structure specified by `dest_image`.

The depth of the destination **XImage** structure must be the same as that of the drawable. If the specified subimage does not fit at the specified location on the destination image, the right and bottom edges are clipped. If the drawable is a pixmap, the given rectangle must be wholly contained within the pixmap, or a **BadMatch** error results. If the drawable is a window, the window must be viewable, and it must be the case that if there were no inferiors or overlapping windows, the specified rectangle of the window would be fully visible on the screen and wholly contained within the outside edges of the window, or a **BadMatch** error results. If the window has backing-store, then the backing-store contents are returned for regions of the window that are obscured by noninferior windows. If the window does not have backing-store, the returned contents of such obscured regions are undefined. The returned contents of visible regions of inferiors of a different depth than the specified window's depth are also undefined. If a problem occurs, **XGetSubImage** returns NULL.

**XGetSubImage** can generate **BadDrawable**, **BadGC**, **BadMatch**, and **BadValue** errors.

**BadDrawable** A value for a Drawable argument does not name a defined Window or Pixmap. **BadGC** A value for a GContext argument does not name a defined GContext. **BadMatch** An **InputOnly** window is used as a Drawable. **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. **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.

*Xlib – C Language X Interface*