NAME

XReadScreen – returns the displayed colors in a rectangle of the screen

SYNOPSIS

XImage
XReadScreen (Display **display,* **Window** *w,* **int** *x,* **int** *y,* **unsigned int** *width,* **unsigned int** *height,* **Bool** *includeCursor)*

Arguments

display Specifies the connection to the X server.

w Specifies the window from whose screen the data is read.

x, y Specify the X and Y coordinates of the upper-left corner of the rectangle relative to the origin of the window *w.*

width, height
Specify the width and height of the rectangle.

includeCursor
Specifies whether the cursor image is to be included in the colors returned.

DESCRIPTION

This routine provides access to the colors displayed on the screen of the given window. On some types of advanced display devices, the displayed colors can be a composite of the data contained in several different frame stores and these frame stores can be of different depth and visual types.

In addition, there can be overlay/underlay window pairs in which part of the underlay is visible beneath the overlay. Because the data returned by **XGetImage** is undefined for portions of the rectangle that have different depths, **XGetImage** is inadequate to return a picture of the what user is actually seeing on the screen. In addition, **XGetImage** cannot composite pixel information for an overlay/underlay window pair because the pixel information lies in different drawables. **XReadScreen** addresses these problems.

Rather than returning pixel information, **XReadScreen** returns color information-the actual displayed colors visible on the screen. It returns the color information from any window within the boundaries of the specified rectangle. Unlike **XGetImage,** the returned contents of visible regions of inferior or overlapping windows of a different depth than the specified window's depth are not undefined. Instead, the actual displayed colors for these windows is returned.

Note: The colors returned are the ones that would be displayed if an unlimited number of hardware color LUTs were available on the screen. Thus, the colors returned are the theoretical display colors. If colormap flashing is present on the screen because there aren't enough hardware color LUTs to display all of the software colormaps simultaneously, the returned colors may be different from the colors that are actually displayed.

If *w* is an overlay window, the overlay color information is returned everywhere there is opaque paint in the specified rectangle. The color information of the underlay is returned everywhere there is transparent paint in the overlay. In general, since this underlay can be an overlay window containing transparent paint, the color information for a coordinate (x, y) which contains transparent paint is the youngest non-inferior that has opaque paint at (x, y).

The color data is returned as an **XImage.** The returned image has the same width and height as the arguments specified. The format of the image is **ZPixmap.** The depth of the image is 24 and the bits_per_pixel is 32. The most significant 8 bits of color information for each color channel (red, green blue) will be returned in the bit positions defined by *red_mask*, *green_mask*, and *blue_mask* in the **XImage.** The values of the following attributes of the **XImage** are server dependent: *byte_order*, *bitmap_unit*, *bitmap_bit_order*, *bitmap_pad*, *bytes_per_line*, *red_mask*, *green_mask*, *blue_mask.*

If *includeCursor* is **True,** the cursor image is included in the returned colors. Otherwise, it is excluded.

Note that the borders of the argument window (and other windows) can be included and read with this request.

If a problem occurs, **XReadScreen** returns NULL.