NAME

XInitImage, XCreateImage, XGetPixel, XPutPixel, XSubImage, XAddPixel, XDestroyImage - image utilities

SYNTAX

Status XInitImage(XImage *_image_);

XImage *XCreateImage(Display *_display_, Visual *_visual_, unsigned int _depth_, int _format_, int _offset_, char *_data_, unsigned int _width_, unsigned int _height_, int _bitmap_pad_, int _bytes_per_line_);

unsigned long XGetPixel(XImage *_ximage_, int _x_, int _y_);

int XPutPixel(XImage *_ximage_, int _x_, int _y_, unsigned long _pixel_);

XImage *XSubImage(XImage *_ximage_, int _x_, int _y_, unsigned int _subimage_width_, unsigned int _subimage_height_);

int XAddPixel(XImage *_ximage_, long _value_);

int XDestroyImage(XImage *_ximage_);

ARGUMENTS

bitmap_pad Specifies the quantum of a scanline (8, 16, or 32). In other words, the start of one scanline is separated in client memory from the start of the next scanline by an integer multiple of this many bits.

bytes_per_line Specifies the number of bytes in the client image between the start of one scanline and the start of the next.

data Specifies the image data.

depth Specifies the depth of the image.

display Specifies the connection to the X server.

format Specifies the format for the image. You can pass **XYBitmap**, **XYPixmap**, or **ZPixmap**.

height Specifies the height of the image, in pixels.

offset Specifies the number of pixels to ignore at the beginning of the scanline.

pixel Specifies the new pixel value.

subimage_height
Specifies the height of the new subimage, in pixels.

subimage_width
Specifies the width of the new subimage, in pixels.

value Specifies the constant value that is to be added.

visual Specifies the **Visual** structure.

width Specifies the width of the image, in pixels.

ximage Specifies the image.

x

y Specify the x and y coordinates.

DESCRIPTION

The **XInitImage** function initializes the internal image manipulation routines of an image structure, based on the values of the various structure members. All fields other than the manipulation routines must already be initialized. If the bytes_per_line member is zero, **XInitImage** will assume the image data is contiguous in memory and set the bytes_per_line member to an appropriate value based on the other members; otherwise, the value of bytes_per_line is not changed. All of the manipulation routines are initialized to functions that other Xlib image manipulation functions need to operate on the type of image specified by the rest of the structure.

This function must be called for any image constructed by the client before passing it to any other Xlib function. Image structures created or returned by Xlib do not need to be initialized in this fashion.

This function returns a nonzero status if initialization of the structure is successful. It returns zero if it detected some error or inconsistency in the structure, in which case the image is not changed.

The **XCreateImage** function allocates the memory needed for an **XImage** structure for the specified display but does not allocate space for the image itself. Rather, it initializes the structure byte-order, bit-order, and bitmap-unit values from the display and returns a pointer to the **XImage** structure. The red, green, and blue mask values are defined for Z format images only and are derived from the **Visual** structure passed in. Other values also are passed in. The offset permits the rapid displaying of the

image without requiring each scanline to be shifted into position. If you pass a zero value in bytes_per_line, Xlib assumes that the scanlines are contiguous in memory and calculates the value of bytes_per_line itself.

Note that when the image is created using **XCreateImage**, **XGetImage**, or **XSubImage**, the destroy procedure that the **XDestroyImage** function calls frees both the image structure and the data pointed to by the image structure.

The basic functions used to get a pixel, set a pixel, create a subimage, and add a constant value to an image are defined in the image object. The functions in this section are really macro invocations of the functions in the image object and are defined in **X11/Xutil.h**.

The **XGetPixel** function returns the specified pixel from the named image. The pixel value is returned in normalized format (that is, the least significant byte of the long is the least significant byte of the pixel). The image must contain the x and y coordinates.

The **XPutPixel** function overwrites the pixel in the named image with the specified pixel value. The input pixel value must be in normalized format (that is, the least significant byte of the long is the least significant byte of the pixel). The image must contain the x and y coordinates.

The **XSubImage** function creates a new image that is a subsection of an existing one. It allocates the memory necessary for the new **XImage** structure and returns a pointer to the new image. The data is copied from the source image, and the image must contain the rectangle defined by x, y, subimage_width, and subimage_height.

The **XAddPixel** function adds a constant value to every pixel in an image. It is useful when you have a base pixel value from allocating color resources and need to manipulate the image to that form.

The **XDestroyImage** function deallocates the memory associated with the **XImage** structure.

SEE ALSO

XPutImage(3)
Xlib - C Language X Interface