NAME

XpmRead - read an XPM file

SYNOPSIS

int XpmReadFileToImage(Display ****display*, char ****filename***,**
XImage *****image_return*, XImage *****shapeimage_return***,**
XpmAttributes ****attributes*);**

int XpmReadFileToPixmap(Display ****display*, Drawable** *d***, char ****filename***,**
Pixmap ****pixmap_return*, Pixmap ****shapemask_return***,**
XpmAttributes ****attributes*);**

int XpmReadFileToXpmImage(char ****filename*, XpmImage ****image***,**
XpmInfo ****info*);**

int XpmReadFileToBuffer(char ****filename*, char *****buffer_return***);**

int XpmReadFileToData(char ****filename*, char ******data_return***);**

ARGUMENTS

display
Specifies the connection to the X server.

filename
Specifies the file name to use.

image_return
Returns the image which is created.

shapeimage_return
Returns the shape mask image which is created if the color None is used.

attributes
Specifies the location of a structure to get and store information (or NULL).

buffer_return
Returns the buffer created.

data_return
Returns the data array created.

image
Specifies the image structure location.

info
Specifies the location of a structure to store possible information (or NULL).

DESCRIPTION

XpmReadFileToImage

The **XpmReadFileToImage**() function reads in a file in the XPM format. If the file cannot be opened it returns **XpmOpenFailed**. If the file can be opened but does not contain valid XPM data, it returns **XpmFileInvalid**. If insufficient working storage is allocated, it returns **XpmNoMemory**. If the passed XpmAttributes structure pointer is not NULL, **XpmReadFileToImage**() looks for the following attributes: XpmVisual, XpmColormap, XpmDepth, XpmColorSymbols, XpmExactColors, XpmCloseness, XpmRGBCloseness, XpmAllocCloseColors, XpmReturnPixels, XpmReturnAllocPixels, XpmAllocColor, XpmFreeColors, XpmColorClosure, XpmReturnExtensions, XpmReturnColorTable, XpmBitmapFormat, sets the XpmSize, the XpmCharsPerPixel, and possibly the XpmHotspot attributes when returning. As a backward compatibility feature, **XpmReadFileToImage**() also looks for the XpmReturnInfos attributes. As specified in the table (page 12), if the data related to the attributes XpmReturnExtensions, XpmReturnColorTable, and XpmReturnInfos cannot be returned as requested because of insufficient memory storage, **XpmReadFileToImage**() will change the valuemask to mention this and will try to continue. So the caller should check on this before accessing this data.

Note: The valuemask of the passed XpmAttributes must be set to some valid value, at least zero, otherwise unpredictable errors can occur.

XpmReadFileToImage() allocates colors, as read from the file or possibly overridden as specified in the XpmColorSymbols attributes. The colors are allocated using the color settings for the visual specified by the XpmColorKey attribute, which has the value XPM_MONO, XPM_GRAY4, XPM_GRAY, or XPM_COLOR. If the XpmColorKey attribute is not set it is determined by examining the type of visual. If no default value exists for the specified visual, it first looks for other defaults nearer to the monochrome visual type and secondly nearer to the color visual type. If the color which is found is not valid (cannot be parsed), it looks for another default one according to the same algorithm. If allocating a color fails, and the closeness attribute is set, it tries to find a color already in the colormap that is closest to the desired color, and uses that. If the alloc_close_colors attribute is set to False, the found close color is not allocated but it is used anyway. This is especially useful for

applications which use a private colormap containing read/write cells and have complete control over the colormap. On the other hand, since in such a case there is no guarantee that the color pixel will not change any time, this should be avoided when using the default colormap. If no color can be found that is within closeness of the Red, Green, and Blue components of the desired color, it reverts to trying other default values as explained above. For finer control over the closeness requirements of a particular icon, the red_closeness, green_closeness, and blue_closeness attributes may be used instead of the more general closeness attribute.

The RGB components are integers within the range 0 (black) to 65535 (white). A closeness of less than 10000, for example, will cause only quite close colors to be matched, while a closeness of more than 50000 will allow quite dissimilar colors to match. Specifying a closeness of more than 65535 will allow any color to match, thus forcing the icon to be drawn in color no matter how bad the colormap is. The value 40000 seems to be about right for many situations requiring reasonable but not perfect matches. With this setting the color must only be within the same general area of the RGB cube as the desired color. If the exactColors attribute is set it then returns **XpmColorError**, otherwise it creates the images and returns **XpmSuccess**. If no color is found, and no close color exists or is wanted, and all visuals have been exhausted, **XpmColorFailed** is returned.

XpmReadFileToImage() returns the created image to *image_return* if not NULL and possibly the created shapemask to *shapeimage_return* if not NULL and the color None is used. If required it stores into the XpmAttributes structure the list of the used pixels. When the image depth is one, the image format is either as specified by the bitmap_format attribute if set or ZPixmap. When the depth is different from one the image format is always ZPixmap. When finished the caller must free the images using **XDestroyImage**(3), the allocated colors using **XFreeColors**(3) or the application equivalent function when the standard Xlib functions are not used, and possibly the data returned into the XpmAttributes using **XpmFreeAttributes**(3). In addition, on systems which support such features **XpmReadFileToImage**() deals with compressed files by forking an uncompress or gzip process and reading from the piped result. It assumes that the specified file is compressed if the given file name ends by '.Z' or '.gz'. In case the file name does not end so, **XpmReadFileToImage**() looks for the given file name assuming it is not a compressed file. And if instead of a file name NULL is passed to **XpmReadFileToImage**(), it reads from the standard input.

XpmReadFileToPixmap

The **XpmReadFileToPixmap**() function creates X images using **XpmReadFileToImage**() and thus returns the same errors. In addition on success it then creates the related pixmaps, using **XPutImage**(3), which are returned to *pixmap_return* and *shapemask_return* if not NULL, and finally destroys the created images using **XDestroyImage**(3). When finished the caller must free the pixmaps using **XFreePixmap**(3), the allocated colors using **XFreeColors**(3) or the application equivalent function when the standard Xlib functions are not used, and possibly the data returned into the XpmAttributes

using **XpmFreeAttributes**(3).

XpmReadFileToBuffer

XpmReadFileToBuffer() allocates and fills a buffer from a file. **XpmReadFileToBuffer**() returns **XpmOpenFailed** if it cannot open the file, returns **XpmNoMemory** if insufficient working storage is allocated, and **XpmSuccess** otherwise. The allocated buffer returned by **XpmReadFileToBuffer**() should be freed with **XpmFree**(3) when done.

As a convenience, the **XpmReadFileToBuffer**() and **XpmWriteFileFromBuffer**(3) are provided to copy a file to a buffer and to write a file from a buffer. Thus for instance one may decide to use **XpmReadFileToBuffer**(), **XpmCreatePixmapFromBuffer**(3), and **XpmFree**(3) instead of **XpmReadFileToPixmap**(). On some systems this may lead to a performance improvement, since the parsing will be performed in memory, but it uses more memory.

XpmReadFileToData

XpmReadFileToData() returns **XpmOpenFailed** if it cannot open the file, **XpmNoMemory** if insufficient working storage is allocated, **XpmFileInvalid** if this is not a valid XPM file, and **XpmSuccess** otherwise. The allocated data returned by **XpmReadFileToData**() should be freed with **XpmFree**(3) when done.

XpmReadFileToXpmImage

The **XpmReadFileToXpmImage**() function reads in a file in the XPM format. If the file cannot be opened it returns **XpmOpenFailed**. If the file can be opened but does not contain valid XPM data, it returns **XpmFileInvalid**. If insufficient working storage is allocated, it returns **XpmNoMemory**. On success it fills in the given XpmImage structure and returns **XpmSuccess**.

SEE ALSO

XpmCreateBuffer(3), **XpmCreateData**(3), **XpmCreateImage**(3), **XpmCreatePixmap**(3), **XpmCreateXpmImage**(3), **XpmFreeAttributes**(3), **XpmWrite**(3)