XcmsQueryColor, XcmsQueryColors, XcmsLookupColor – obtain color values

Status XcmsQueryColor(*display***,** *colormap***,** *color_in_out***,** *result_format***)**
Display ***display*;**
Colormap *colormap***;**
XcmsColor ***color_in_out*;**
XcmsColorFormat *result_format***;**

Status XcmsQueryColors(*display*, *colormap*, *colors_in_out*, *ncolors*, *result_format*)
Display **display*;
Colormap *colormap*;
XcmsColor *colors_in_out*[ ];
unsigned int *ncolors*;
XcmsColorFormat *result_format*;

Status XcmsLookupColor(*display*, *colormap*, *color_string*, *color_exact_return*, *color_screen_return*, *result_format*)
Display **display*;
Colormap *colormap*;
char **color_string*;
XcmsColor **color_exact_return*, **color_screen_return*;
XcmsColorFormat *result_format*;

display Specifies the connection to the X server.

colormap Specifies the colormap.

color_exact_return Returns the color specification parsed from the color string or parsed from the corresponding string found in a color-name database.

color_in_out Specifies the pixel member that indicates the color cell to query. The color specification stored for the color cell is returned in this **XcmsColor** structure.

color_screen_return Returns the color that can be reproduced on the screen.

color_string Specifies the color string.

result_format Specifies the color format for the returned color specifications (color_screen_return and color_exact_return arguments). If the format is **XcmsUndefinedFormat** and the color string contains a numerical color specification, the specification is returned in the format used in that numerical color specification. If the format is **XcmsUndefinedFormat** and the color string contains a color name, the specification is returned in the format used to store the color in the database.

ncolors Specifies the number of **XcmsColor** structures in the color-specification array.

The XcmsQueryColor function obtains the RGB value for the pixel value in the pixel member of the specified **XcmsColor** structure and then converts the value to the target format as specified by the result_format argument. If the pixel is not a valid index in the specified colormap, a **BadValue** error results. The **XcmsQueryColors** function obtains the RGB values for pixel values in the pixel members of **XcmsColor** structures and then converts the values to the target format as specified by the result_format argument. If a pixel is not a valid index into the specified colormap, a **BadValue** error results. If more than one pixel is in error, the one that gets reported is arbitrary.

XcmsQueryColor and **XcmsQueryColors** can generate **BadColor** and **BadValue** errors.

The **XcmsLookupColor** function looks up the string name of a color with respect to the screen associated with the specified colormap. It returns both the exact color values and the closest values provided by the screen with respect to the visual type of the specified colormap. The values are returned in the format specified by result_format. If the color name is not in the Host Portable Character Encoding, the result is implementation-dependent. Use of uppercase or lowercase does not matter. **XcmsLookupColor** returns

XcmsSuccess or **XcmsSuccessWithCompression** if the name is resolved; otherwise, it returns **XcmsFailure**. If **XcmsSuccessWithCompression** is returned, the color specification returned in color_screen_return is the result of gamut compression.

BadColor A value for a Colormap argument does not name a defined Colormap. **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.

XcmsAllocColor(3X11), XcmsStoreColor(3X11), XQueryColor(3X11)
Xlib – C Language X Interface