NAME

XcmsStoreColor, XcmsStoreColors - set colors

SYNTAX

Status XcmsStoreColor(Display **display*, Colormap *colormap*, XcmsColor **color*);

Status XcmsStoreColors(Display **display*, Colormap *colormap*, XcmsColor **colors*, unsigned *ncolors*, Bool **compression_flags_return*);

ARGUMENTS

display Specifies the connection to the X server.

color Specifies the color cell and the color to store. Values specified in this **XcmsColor** structure remain unchanged on return.

colors Specifies the color specification array of **XcmsColor** structures, each specifying a color cell and the color to store in that cell. Values specified in the array remain unchanged upon return.

colormap Specifies the colormap.

compression_flags_return
Returns an array of Boolean values indicating compression status. If a non-NULL pointer is supplied, each element of the array is set to **True** if the corresponding color was compressed and **False** otherwise. Pass NULL if the compression status is not useful.

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

DESCRIPTION

The **XcmsStoreColor** function converts the color specified in the **XcmsColor** structure into RGB values. It then uses this RGB specification in an **XColor** structure, whose three flags (**DoRed**, **DoGreen**, and **DoBlue**) are set, in a call to **XStoreColor** to change the color cell specified by the pixel member of the **XcmsColor** structure. This pixel value must be a valid index for the specified colormap, and the color cell specified by the pixel value must be a read/write cell. If the pixel value is not a valid index, a **BadValue** error results. If the color cell is unallocated or is allocated read-only, a **BadAccess** error results. If the colormap is an installed map for its screen, the changes are visible immediately.

Note that **XStoreColor** has no return value; therefore, an **XcmsSuccess** return value from this function indicates that the conversion to RGB succeeded and the call to **XStoreColor** was made. To obtain the

actual color stored, use **XcmsQueryColor**. Because of the screen's hardware limitations or gamut compression, the color stored in the colormap may not be identical to the color specified.

XcmsStoreColor can generate **BadAccess**, **BadColor**, and **BadValue** errors.

The **XcmsStoreColors** function converts the colors specified in the array of **XcmsColor** structures into RGB values and then uses these RGB specifications in **XColor** structures, whose three flags (**DoRed**, **DoGreen**, and **DoBlue**) are set, in a call to **XStoreColors** to change the color cells specified by the pixel member of the corresponding **XcmsColor** structure. Each pixel value must be a valid index for the specified colormap, and the color cell specified by each pixel value must be a read/write cell. If a pixel value is not a valid index, a **BadValue** error results. If a color cell is unallocated or is allocated read-only, a **BadAccess** error results. If more than one pixel is in error, the one that gets reported is arbitrary. If the colormap is an installed map for its screen, the changes are visible immediately.

Note that **XStoreColors** has no return value; therefore, an **XcmsSuccess** return value from this function indicates that conversions to RGB succeeded and the call to **XStoreColors** was made. To obtain the actual colors stored, use **XcmsQueryColors**. Because of the screen's hardware limitations or gamut compression, the colors stored in the colormap may not be identical to the colors specified.

XcmsStoreColors can generate **BadAccess**, **BadColor**, and **BadValue** errors.

DIAGNOSTICS

BadAccess A client attempted to free a color map entry that it did not already allocate.

BadAccess A client attempted to store into a read-only color map entry.

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.

SEE ALSO

XcmsAllocColor(3), XcmsQueryColor(3)
Xlib - C Language X Interface