NAME

XStoreColors, XStoreColor, XStoreNamedColor - set colors

SYNTAX

int XStoreColors(Display **display*, Colormap *colormap*, XColor *color*[], int *ncolors*);

int XStoreColor(Display **display*, Colormap *colormap*, XColor **color*);

int XStoreNamedColor(Display **display*, Colormap *colormap*, _Xconst char **color*, unsigned long *pixel*, int *flags*);

ARGUMENTS

color Specifies the pixel and RGB values or the color name string (for example, red).

color Specifies an array of color definition structures to be stored.

colormap Specifies the colormap.

display Specifies the connection to the X server.

flags Specifies which red, green, and blue components are set.

ncolors Specifies the number of **XColor** structures in the color definition array.

pixel Specifies the entry in the colormap.

DESCRIPTION

The **XStoreColors** function changes the colormap entries of the pixel values specified in the pixel members of the **XColor** structures. You specify which color components are to be changed by setting **DoRed**, **DoGreen**, and/or **DoBlue** in the flags member of the **XColor** structures. If the colormap is an installed map for its screen, the changes are visible immediately. **XStoreColors** changes the specified pixels if they are allocated writable in the colormap by any client, even if one or more pixels generates an error. If a specified pixel is not a valid index into the colormap, a **BadValue** error results. If a specified pixel either 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.

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

The **XStoreColor** function changes the colormap entry of the pixel value specified in the pixel member of the **XColor** structure. You specified this value in the pixel member of the **XColor** structure. This

pixel value must be a read/write cell and a valid index into the colormap. If a specified pixel is not a valid index into the colormap, a **BadValue** error results. **XStoreColor** also changes the red, green, and/or blue color components. You specify which color components are to be changed by setting **DoRed**, **DoGreen**, and/or **DoBlue** in the flags member of the **XColor** structure. If the colormap is an installed map for its screen, the changes are visible immediately.

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

The **XStoreNamedColor** function looks up the named color with respect to the screen associated with the colormap and stores the result in the specified colormap. The pixel argument determines the entry in the colormap. The flags argument determines which of the red, green, and blue components are set. You can set this member to the bitwise inclusive OR of the bits **DoRed**, **DoGreen**, and **DoBlue**. 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. If the specified pixel is not a valid index into the colormap, a **BadValue** error results. If the specified pixel either is unallocated or is allocated read-only, a **BadAccess** error results.

XStoreNamedColor can generate **BadAccess**, **BadColor**, **BadName**, 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.

BadName A font or color of the specified name does not exist.

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

XAllocColor(3), XCreateColormap(3), XQueryColor(3)
Xlib - C Language X Interface