Color and Pixmaps

Motif uses colors and pixmaps for several general purposes:

- To fill window backgrounds and borders
- To draw text and graphics in window foregrounds
- To generate shadows with a three−dimensional appearance
- To highlight the widget that has keyboard focus
- To indicate that a button is armed or selected

Motif uses other pixmaps for specific purposes:

- As the application's icon for use by the window manager
- For drag icons and drop site animation
- As a CascadeButton symbol indicating that a menu is attached to the CascadeButton
- As a MessageBox symbol indicating the type of message displayed
- As the face of a button when the button is insensitive

All of these colors and pixmaps are represented as resources. The user or application can set the resource values using resource files, and the application can set them using **XtSetValues**.

Motif also uses a number of pixmaps that are not represented as resources. The user and application cannot change these. Among these fixed pixmaps are the pixmaps used to draw arrows in ScrollBars.

Default Colors and Pixmaps

The following resources determine the colors or pixmaps generally used in Motif:

Borders

Core resources **XmNborderColor** and **XmNborderPixmap**

Backgrounds

Core resources **XmNbackground** and **XmNbackgroundPixmap**

Foregrounds

Gadget, Primitive, and Manager resource **XmNforeground**; Label and LabelGadget resources **XmNlabelPixmap** and **XmNlabelInsensitivePixmap**

Shadows

Gadget, Primitive, and Manager resources **XmNbottomShadowColor**, **XmNbottomShadowPixmap**, **XmNtopShadowColor**, and **XmNtopShadowPixmap**

Focus highlights

Gadget, Primitive, and Manager resources **XmNhighlightColor** and **XmNhighlightPixmap**

Arming and selection

PushButton and PushButtonGadget resources **XmNarmColor** and **XmNarmPixmap**; ToggleButton and ToggleButtonGadget resources **XmNarmColor**, **XmNindeterminateInsensitivePixmap**, **XmNindeterminatePixmap**, **XmNselectInsensitivePixmap**, **XmNselectPixmap**, and **XmNunselectColor**; ScrollBar resource **XmNtroughColor**; Display resource **XmNenableToggleColor**

The following sections describe these groups of resources and their defaults.

Borders

The border color or border pixmap is used to fill the border of a widget if **XmNborderWidth** is greater than 0. Note that the border is outside the widget; that is, it is not within the area determined by the widget's **XmNheight** and **XmNwidth**. If the user or application supplies a value for **XmNborderPixmap**, that pixmap is used to fill the border; otherwise, **XmNborderColor** is used.

If the application resource *reverseVideo* is False or unspecified, the default for **XmNborderColor** is the black pixel of the widget's screen. If *reverseVideo* is True, the default for **XmNborderColor** is the white pixel of the widget's screen.

Backgrounds

The background color or background pixmap is used to fill a widget before anything else is displayed in it. If the user or application supplies a value for **XmNbackgroundPixmap**, that pixmap is used to fill the background; otherwise, the **XmNbackground** color is used. A gadget inherits the background color and background pixmap of its parent, if those resources are not set.

The default for **XmNbackground** is implementation dependent.

Foregrounds

The foreground color is used to display text and most graphics in a widget. Most widgets use the **XmNforeground** color for this purpose. Label, LabelGadget, and their subclasses (except for ToggleButton and ToggleButtonGadget which are special cases described later in this chapter) have pixmap resources that are used for the face of the label or button when **XmNlabelType** is set to **XmPIXMAP**. In this case, **XmNlabelPixmap** is used for the face when the widget is sensitive, and **XmNlabelInsensitivePixmap** is used when the widget is insensitive. A gadget inherits the foreground color of its parent (if none is set).

The default for **XmNforeground** is a color that contrasts with the background color, based on the XmScreen resource **XmNforegroundThreshold**. The value of this resource is an integer between 0 and 100, inclusive, that specifies a level of perceived brightness for a color. If the perceived brightness of the background color is equal to or below this level, Motif treats the background as "dark" when computing the default foreground color. If the perceived brightness of the background color is above this level, Motif treats the background as "light" when computing the default foreground color. When the background is "dark," the default foreground is white; when the background is "light," the default foreground is black.

Shadows

The top shadow color or top shadow pixmap is used to draw the top and left sides of the three−dimensional shadow at the edge of some widgets. If the user or application supplies a value for **XmNtopShadowPixmap**, that pixmap is used for the top and left sides; otherwise, **XmNtopShadowColor** is used.

The bottom shadow color or bottom shadow pixmap is used to draw the bottom and right sides of the three−dimensional shadow. If the user or application supplies a value for **XmNbottomShadowPixmap**, that pixmap is used for the bottom and right sides; otherwise, **XmNbottomShadowColor** is used.

A gadget inherits the top and bottom shadow colors and pixmaps of its parent, if those resources are not set.

In computing the defaults for **XmNtopShadowColor** and **XmNbottomShadowColor**, Motif uses the XmScreen resources **XmNdarkThreshold** and **XmNlightThreshold**. The value of each resource is an integer between 0 and 100, inclusive, that specifies a level of perceived brightness for a color. If the perceived brightness of the background color is equal to or below the **XmNdarkThreshold**, Motif treats the background as "dark" when computing the default shadow colors. If the perceived brightness of the background color is above the **XmNlightThreshold**, Motif treats the background as "light" when computing the default shadow colors. Otherwise, Motif treats the background as "medium" when computing the defaults.

Motif computes the defaults in the following way, depending on the perceived brightness of the background:

Dark background

> The top and bottom shadow colors are interpolated toward white from the background, with the top shadow color shifted more toward white than the bottom shadow color.

Light background

> The top and bottom shadow colors are interpolated toward black from the background, with the bottom shadow color shifted more toward black than the top shadow color.

Medium background

> The top shadow color is interpolated toward white from the background, and the bottom shadow color is interpolated toward black from the background.

Focus Highlights

The highlight color or highlight pixmap is used to draw the highlighting rectangle around widgets that have keyboard focus. If the user or application supplies a value for **XmNhighlightPixmap**, that pixmap is used for the highlight; otherwise, **XmNhighlightColor** is used. The highlight color is also used to draw the location cursor around List items that have keyboard focus. A gadget inherits the highlight color and highlight pixmap of its parent if those resources are not set.

The default highlight color is the same as the default foreground color.

Arming and Selection

In PushButtons and PushButtonGadgets outside menus, **XmNarmColor** is used as the button background when the **XmNfillOnArm** resource is True and the user arms the button. In PushButtons and PushButtonGadgets outside menus, **XmNarmPixmap** is used as the button face (the label area) when **XmNlabelType** is **XmPIXMAP** and the user arms the button. These also work inside menus if the **XmNenableEtchedInMenu** resource of **XmDisplay** is **True**.

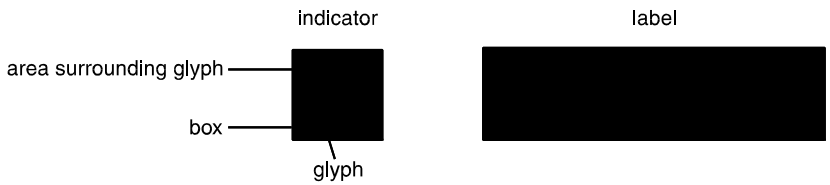
Selection rules in ToggleButton and ToggleButtonGadget are rather complicated. Many different resources control the colors and pixmaps

displayed by ToggleButtons. Before getting into the selection rules, we should define a few terms. ToggleButton and ToggleButtonGadgets display the following visual areas:

- A label, which consists of either text or a pixmap.
- An optional indicator, which consists of an optional glyph, an enclosing border, and the background area between the enclosing border and the glyph. **XmNindicatorOn** specifies whether or not the ToggleButton or ToggleButtonGadget contains an indicator area. If it does, **XmNindicatorOn** also specifies the kind of indicator. The optional glyph is either a checkmark or a cross. If **XmNindicatorType** is **XmN_OF_MANY**, then the glyph is included as part of the indicator area. If **XmNindicatorType** is something other than **XmN_OF_MANY**, then the indicator area does not contain a glyph.

Figure 12–1 illustrates a ToggleButton displaying these two visual areas:

Figure 1 Visual Areas of a ToggleButton



XmNset specifies whether a user or application has set or unset a ToggleButton or ToggleButtonGadget. A third possibility is that the ToggleButton or ToggleButtonGadget is neither set nor unset, but is in an indeterminate state. ToggleButton and ToggleButtonGadget must visually convey the value of **XmNset** through colors and pixmaps.

Table 12–1 summarizes the color used to paint various "background areas" in ToggleButtons or ToggleButtonGadgets. For example, the background area of an indicator is the area surrounding the glyph. Table 12–1 assumes that the ToggleButton or ToggleButtonGadget is outside of a menu and that **XmNindicatorOn** is something other than **XmINDICATOR_NONE**.

Table 1 Color of Various Background Areas in ToggleButton

XmNfillOnSelect =	**XmNset**	*Resulting Color of "Background Area"*
True	**XmSET**	**XmNselectColor**
True	**XmUNSET**	**XmNunselectColor**
True	**XmINDETERMINATE**	Stipple of **XmNselectColor** and **XmNunselectColor**
False	any value	**XmNbackground**

If **XmNindicatorOn** is **XmINDICATOR_NONE**, then the ToggleButton or ToggleButtonGadget displays no indicator area. In this case, the colors in Table 12–1 apply to the background area of the label.

If the **XmNindicatorType** is **XmN_OF_MANY**, then the indicator area contains a glyph. Table 12–2 summarizes the color of the glyph.

Table 2 Color of Glyph in ToggleButton

XmNset =	*Resulting Glyph Color*
XmSET	**XmNforeground**
XmUNSET	No glyph is drawn
XmINDETERMINATE	Stipple of **XmNforeground** and **XmNbackground**

If a user or application does not specify a value for **XmNselectColor**, then the **XmNenableToggleColor** resource of **XmDisplay** determines the background color.

If the label area is a pixmap, then the ToggleButton's sensitivity and the value of its **XmNset** resource determines which pixmap is rendered. Table 12–3 summarizes these combinations.

Table 3 Pixmap Used in Label Area of ToggleButton

XmNset =	*Sensitive?*	*Resulting Pixmap*
XmSET	Yes	**XmNselectPixmap**
XmUNSET	Yes	**XmNlabelPixmap**
XmINDETERMINATE	Yes	**XmNindeterminatePixmap**
XmSET	No	**XmNselectInsensitivePixmap**
XmUNSET	No	**XmNlabelInsensitivePixmap**
XmINDETERMINATE	No	**XmNindeterminateInsensitivePixmap**

In ScrollBars, **XmNtroughColor** is used to fill the part of the slider area that is not taken up by the slider.

Motif computes a single default, known as the select color, for **XmNarmColor**, **XmNselectColor**, and **XmNtroughColor**. Motif uses the XmScreen resources **XmNdarkThreshold** and **XmNlightThreshold** to determine whether the background is "dark," "light," or "medium" in the same way as for shadow colors. Motif then computes the default in the following way:

Dark background

The select color is interpolated toward white from the background.

Light background

The select color is interpolated toward black from the background.

Medium background

The select color is interpolated toward black from the background.

Application–Defined Color Generation

Motif generates default colors whenever the application creates a widget that has no specified value for one or more of the color resources. Motif does not regenerate default colors when an application changes the value of **XmNbackground** using **XtSetValues**.

An application can use **XmChangeColor** to recalculate default colors for a widget based on a new background and set the appropriate color resources in the widget. For primitives and managers, **XmChangeColor** sets **XmNbackground**, **XmNforeground**, **XmNtopShadowColor**, and **XmNbottomShadowColor**. For widgets and gadgets with select colors, **XmChangeColor** also sets the appropriate resources for those colors.

An application can use **XmGetColors** to produce default colors for a given background color without setting any resources. **XmGetColors** takes as arguments a screen pointer, a colormap, and a background pixel representing a color allocated in the colormap. **XmGetColors** also has return arguments that are pointers to pixel values for the foreground, top shadow, bottom shadow, and select colors. The function generates default colors for the given background. For each of the return arguments that is not NULL, **XmGetColors** allocates a color in the colormap and returns the pixel value at the address specified by the argument.

By default, **XmChangeColor** and **XmGetColors** calculate colors as described in the previous discussion of default colors. An application can use **XmSetColorCalculation** to change the procedure that these routines use and that Motif uses to calculate default colors when the application creates a widget. **XmSetColorCalculation** takes as its only argument a procedure of type **XmColorProc**. It sets Motif's color–calculation procedure to the new **XmColorProc** and returns the color–calculation procedure used previously. **XmSetColorCalculation** does not change the procedure used by **XmChangeColor**, **XmGetColors**, and Motif to calculate default colors for a monochrome screen.

Motif calls the **XmColorProc** when it needs to compute default colors or when the application calls **XmChangeColor** or **XmGetColors**. The **XmColorProc** takes five arguments, all pointers to **XColor** structures. The *red*, *green*, *blue*, and *pixel* members of the first structure are filled in with the background color. The procedure calculates *red*, *green*, and *blue* values for the foreground, select, top shadow, and bottom shadow colors and fills in the other four **XColor** structures with these values.

The procedure should not allocate color cells for any of these colors. Motif caches the returned **XColor** structures and allocates a color when it needs a pixel value. Usually Motif allocates a color when it computes the default value for a resource, when the application calls **XmChangeColor**, or when the application calls **XmGetColors** with a non–NULL value for one of the return pixel values. When allocating colors as a result of widget creation or a call to **XmChangeColor**, Motif uses the colormap of the widget. When allocating colors as a result of a call to **XmGetColors**, Motif uses the colormap passed as an argument to the function.

XmGetColorCalculation returns the color–calculation procedure being used at the time of the call to that routine. Calling **XmSetColorCalculation** with an argument of NULL restores the Motif default color–calculation procedure.

Colormaps

The colormap used by a widget is the value of the Core resource **XmNcolormap**. An application that does not supply its own colormap does not need to set this resource. The default for a top−level shell is the default colormap of the screen. For other widgets, the default is copied from the parent.

An application that uses its own colormap should not use **XtAppInitialize** to create the top−level shell, because the shell would then use the screen's default colormap. Instead, the application should open the display, create the colormap, and then call **XtAppCreateShell** with the colormap as the **XmNcolormap** argument.

If an application uses different colormaps for some windows in its hierarchy, it must tell the window manager about those colormaps by setting a WM_COLORMAP_WINDOWS property on the top−level window. See Chapter 18 for more information.

For more information about colormaps, see *Xlib−−C Language X Interface*.

Using Pixmaps

Motif uses pixmaps supplied by the application or the user for widget borders, backgrounds, labels, shadows, focus highlights, and button arming or selection indicators. Motif also uses other pixmaps that the application or user can supply for more specific purposes: as application icons, drag icons, CascadeButton menu indicators, MessageBox symbols, and labels for insensitive buttons.

Motif provides facilities for an application to install and cache images and pixmaps. Motif also has string−to−pixmap resource converters that retrieve pixmaps from the cache or install them from files in X bitmap format or in X pixmap format. Because of these converters, both applications and users can specify pixmaps as resource values from resource files or the command line.

An application can use **XmGetPixmapByDepth** to retrieve or create a pixmap with a specified name, screen, foreground, background, and depth. **XmGetPixmapByDepth** finds or creates a pixmap in the following way:

- It searches the pixmap cache for a pixmap matching the specified name, screen, foreground, background, and depth. If it finds a matching pixmap, it returns the pixmap.
- If it does not find a matching pixmap in the cache, it searches the image cache for an image matching the specified name. If it finds a matching image, it creates and caches a pixmap of the specified depth on the specified screen, transfers the image to the pixmap, and returns the pixmap.
- If it does not find a matching image in the cache, it uses **XtResolvePathname** to search for a file of the specified name. The search path comes from the environment variable **XBMLANGPATH** or, if **XBMLANGPATH** is not set, from a default search path. Performance is usually better when **XBMLANGPATH** is set to a short list of directories than when the system uses the default search path. If it finds such a file, it assumes that the file is in X bitmap format, reads the file, and creates and caches an image in **XYBitmap** format. It then creates and caches a pixmap of the specified depth on the specified screen, transfers the image to the pixmap, and returns the pixmap. (Since images are cached after they are found, the performance improvement from setting **XBMLANGPATH** only affects the initial search for an image.)
- If it does not find a matching X bitmap file, it returns **XmUNSPECIFIED_PIXMAP**.

Motif preinstalls a number of images in the image cache. The names and characteristics of these images are documented in the **XmInstallImage**(3) reference page. Motif offers two ways for an application to provide its own image as the source for a pixmap to be created by **XmGetPixmapByDepth**:

- The application can create its own image, usually by using **XCreateImage** or **XGetImage**. The image can be of any depth. The application can then call **XmInstallImage** to install the image in the image cache by name.
- The application or user can create a file in X bitmap format and install the file under an appropriate name in a directory that is in the search path used by **XmGetPixmapByDepth**. For a description of the X bitmap format, see *Xlib−−C Language X Interface*.

Both of these mechanisms have advantages and disadvantages. An application using **XmInstallImage** can create an image of any depth. However, if it intends to use the image name in a resource specification, it must be sure to call **XmInstallImage** before creating any widgets that use the image.

An application using an X bitmap file is limited to creating an image of depth 1. However, the image is always available for use by a resource converter, and the application can use the search path mechanism of **XtResolvePathname** for such purposes as supplying different images for different locales.

XmInstallImage does not make a copy of the image when it caches it. The application must not destroy the image until it removes the image from the cache, using **XmUninstallImage**. An application can use **XmDestroyPixmap** to free a pixmap cached by **XmGetPixmapByDepth**. **XmDestroyPixmap** does not actually destroy the pixmap until all references to it are freed.