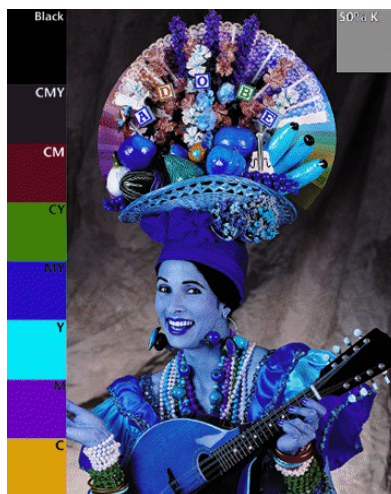


**CCS-ECL Libraries**

**Reference Manual - II**

**for**

**X Panel and Interface Libraries**



## About This Manual

The X Panel and Interface libraries are mainly designed for image generating and controlling. These functions and routines are expected running at real time and full color range handle (display the color as true as possible). Therefore, they are not compatible with XView or Motif GUI. They work only with X internal library

If \*\* symbol appear on any function or sub-routine name line after “—”, this function or sub-routine is not encouraged to be directly used in user’s programming. Most of them are fuzzy layers which are the interface between kernel and user interface. Try to find corresponding interface.

Any function or sub-routine has a \* symbol on its name line after “—” is suggested by using interface call instead of using them directly.

This library is currently not released to public. These functions and sub-routines are available to all HIPS users and LBL users. To be able to use them, contact the responsible people shown in the copyright page in included header files.

**Name:**

BuildColorImage — build a color image in X window

**Synopsis:**

```
BuildColorImage(Image *img,
                 Image *previous_img,
                 char *window_geometry,
                 int icon_factor)
```

**Arguments:**

img                      image structure pointer  
 previous\_img            image pointer to last frame if image is a sequential or  
                          movie.  
 window\_geometry        window geometry description to change window size  
                          different from image size.  
 icon\_factor             icon factor for window icon. Usually, it is generated by  
                          *get\_iconsz*(img, width).

**Description:**

*BuildColorImage*() create color map, dither function, color map function, window, icon window, and X image structure for image displaying. For displaying a single image, the *previous\_img* should pass a NULL pointer.

Note: the *img->name* must be not NULL when calling *BuildColorImage*().

**Program Level:**

User

**Related Functions:**

*create\_windows*(), *choose\_scanline\_converter*(), *eq\_cmap*(),  
*get\_X\_image*(), *get\_dither\_colors*(), *get\_pic*(), *init\_color*()

**Name:**

ButtonPressed — check if a pressbutton is pressed

**Synopsis:**

```
bool  
ButtonPressed( PressButton      *pb,  
               XButtonEvent     *xbutton )
```

**Arguments:**

pb	pressbutton structure pointer
xbutton	X event structure pointer

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

ChangeSliderScale — change slider scale range

**Synopsis:**

```
ChangeSliderScale(  Slider      *s,  
                   int         scale,  
                   bool        active      )
```

**Arguments:**

s	slider structure pointer
scale	new scale factor
active	make change visible now

**Description:****Program Level:**

Panel kernel

**Related Functions:**

CreateSlider()

**Name:**

CreateButton — create a set of rectangle buttons on a control panel

**Synopsis:**

```
CreateButton( Panel    *p,
              int      bx, by,
              int      num_buttons,
              char     *name,
              char     **bnames,
              int      botton_color,
              int      presed_color )
```

**Arguments:**

p	panel on which the button will be set
bx, by	button position on panel
num_buttons	number of buttons in this set
name	button title name
bnames	name list for each button
botton_color	button color
pressed_color	pressed button color

**Description:****Program Level:**

Panel kernel

**Related Functions:**

DrawButton()

**Name:**

CreateCLT — create the Color Lookup Table

**Synopsis:**

```
CreateCLT(  XColor  *cp,
             int      totalevel,
             int      cflag,
             bool     DoQuant,
             bool     rev,
             HistoInfo *hinfo      )
```

**Arguments:**

cp	X color map array pointer (results)
totalevel	number of colors
cflag	DoRed ...
DoQuant	False for doing linear quantization. See Description.
rev	reverse color indices

**Description:**

CreateCLT() create a color lookup table for color mapping. If DoQuant is False, it generates a linear lookup table:

```
cr[0] = cg[0] = cb[0] = 0
cr[1] = cg[1] = cb[1] = 1 * totalevel / 65536
...
cr[n] = cg[n] = cb[n] = n * totalevel / 65536
```

If DoQuant is true, then the value 0 - n on right size will be sorted by image histogram and rearranged by ETA function. This will give a better color mapping result for pseudo-color display

**Program Level:**

X kernel

**Related Functions:**

LinearQuantization()

**Name:**

CreateImage — create image structure with X window

**Synopsis:**

```
CreateImage(  Image          **ip,
              char          *name,
              WinAttribute  *parent,
              int           width, height,
              int           icon_width,
              int           Masks,
              bool          need_buffer  )
```

**Arguments:**

ip	image structure pointer address
name	image name
parent	window attribute structure from parent window
width, height	image dimensions
icon_width	icon width (maximum)
need_buferr	allocate data buffer for Ximage

**Description:****Program Level:**

X kernel

**Related Functions:**

BuildColorImage()



**Name:**

CreatePanel — create a control panel

**Synopsis:**

```
CreatePanel(  int      px, py,
              int      pw, ph,
              char     *name,
              WinAttribute *parent,
              int      mask
              )
```

**Arguments:**

px, py	panel original position. Not important for some window manager.
pw, ph	panel dimensions
name	panel name
parent	parent window attributes structure pointer
mask	X event masks for panel

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

CreatePopupMenu — create a popping menu on a control panel

**Synopsis:**

```
CreatePopupMenu(Panel          *pan,
                 WinAttribute *parent,
                 char          *name,
                 char          *menu[],
                 int            num_item )
```

**Arguments:**

pan	panel on which popping menu inherited
parent	parent window attributes
name	popping menu name
menu	menu item list
num_item	number of items on this menu

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

CreatePressButton — create a press button on a control panel

**Synopsis:**

```
CreatePressButton(      Panel      *pan,  
                      int          x, y,  
                      char         *name,  
                      int          pressed_color )
```

**Arguments:**

an	panel on which press button inherited
x, y	press button position on the panel
name	name for press button
pressed_color	color after button pressed

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

CreateScrollBar — create a scroll bar on any window

**Synopsis:**

```
CreateScrollBar(AnyWindow*awin,
                int      x, y,
                bool     xv, yv,
                int      bar_width,
                int      level_bar_len,
                int      vert_bar_len,
                int      type,
                char     *name,
                int      bar_color,
                int      rail_color )
```

**Arguments:**

x, y	horizontal scroll bar origin
xv, yv	vertical scroll bar origin
bar_width	bar width
level_bar_len	horizontal scroll bar length
vert_bar_len	vertical scroll bar length
type	scroll bar type
name	scroll bar reference name
bar_color	bar color
rail_color	scroll

**Description:****Program Level:**

X kernel

**Related Functions:**

**Name:**

CreateSlider — create a slider on a control panel

**Synopsis:**

```
CreateSlider(  Panel    *pan,
               int      sx, sy,
               int      scale,
               int      v0, v1,
               int      numBars,
               bool     vertical,
               char     *name,
               char     *info,
               int      bar_color,
               int      rail_color )
```

**Arguments:**

pan	panel on which slider is inherited
sx, sy	slider position on the control panel
scale	slider scale factor
v0, v1	sliding range (min - max)
numBars	number of sliding bars
vertical	vertical sliding bar (default is horizontal one)
name	slider title
info	slider description
bar_color	slider bar color
ril_color	slider rail color

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

CreateWindow — create a X window for any purpose

**Synopsis:**

```
CreateWindow(AnyWindow      *awin,  
              WinAttribute   *parent,  
              Cursor         cursor,  
              int            mask,  
              bool          fixedframe,  
              int            wm_flag      )
```

**Arguments:**

awin	any window structure pointer
parent	parent window attributes
cursor	window cursor (No is OK)
mask	X event masks for window
fixedframe	window size is unchangeable
wim_flag	window manager flags

**Description:****Program Level:**

X kernel

**Related Functions:**

**Name:**

Delay\_Clear — \* flushing cursor delay routine

**Synopsis:**

```
Delay_Clear( AnyWindow      *awin,
             int             (*exp_hd)(),
             Image           **imgp,
             int             imgs,
             int             fdelay )
```

**Arguments:**

awin	any window pointer
exp_hd	exposure handle pointer
imgp	image structure array pointer for exp_hd()
imgs	number of images in image structure array
fdelay	delay factor

**Description:****Program Level:**

X kernel

**Related Functions:**

**Name:**

DestroyImage — destroy image content and free memory space

**Synopsis:**

```
DestroyImage(Image      *img)
```

**Arguments:**

img	image structure pointer
-----	-------------------------

**Description:****Program Level:**

kernel

**Related Functions:**



**Name:**

DispInfo — \* display information on any panel or window

**Synopsis:**

```
DispInfo(    Panel    *p,  
            int      xoffset,  
            char     *messages,  
            int      color    )
```

**Arguments:**

p	panel structure pointer
xoffset	start displaying position on left side of the panel
messages	message string
color	message color

**Description:****Program Level:**

assemble

**Related Functions:**

**Name:**

DrawButton — show button on its panel

**Synopsis:**

DrawButton(Button\*        b)

**Arguments:**

      b                    button pointer

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

DrawPixWindow — display pixel information on image window

**Synopsis:**

```
DrawPixWindow(      Image      *img,  
                  int          x, y,  
                  bool         rle_form      )
```

**Arguments:**

img	image structure pointer
x, y	pixel coordinate
rle_form	origin is in RLE conversion

**Description:**

DrawPixWindow() display given pixel information in image display window.

**Program Level:**

X kernel

**Related Functions:**

**Name:**

DrawPressButton — draw pressbutton on its panel

**Synopsis:**

```
DrawPressButton(PressButton*          pb)
```

**Arguments:**

pb                                      press button pointer

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

DrawScrollBars — draw scrollbar on its parent window

**Synopsis:**

```
DrawScrollBars(ScrollBar *sb,  
               int type      )
```

**Arguments:**

sb	scroll bar pointer
type	horizontal or vertical

**Description:****Program Level:**

X kernel

**Related Functions:**

**Name:**

DrawSlider — draw slider on its panel

**Synopsis:**

```
DrawSlider( Slider *s )
```

**Arguments:**

s slider pointer

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

DrawSpeedWindow — display movie speed on image window

**Synopsis:**

```
DrawSpeedWindow(      Image      *img,  
                  int          speed      )
```

**Arguments:**

img	image structure pointer
int	movie speed (frames / sec.) or (1/speed frame / sec. if speed < 0)

**Description:****Program Level:**

assemble

**Related Functions:**

**Name:**

DrawVMark — draw a vertical mark in an image window

**Synopsis:**

```
DrawVMark(  Image    *img,  
            int      nouse,  
            bool     clean      )
```

**Arguments:**

img	image structure pointer
nouse	not used, but for function consistency
clean	clean old before draw new

**Description:****Program Level:**

kernel

**Related Functions:**



**Name:**

Draw\_ImageScrollBars — show scrollbars on an image window

**Synopsis:**

```
Draw_ImageScrollBars(Image          *img)
```

**Arguments:**

img	image structure pointer
-----	-------------------------

**Description:****Program Level:**

X kernel

**Related Functions:**

**Name:**

Draws — draw patterns in an image window by given type

**Synopsis:**

```
Draws(      Image      *img,
           int          arg,
           bool         revs,
           int          type          )
```

**Arguments:**

img	image structure pointer
arg	type argument:
	DrawString: message string
	Others: minimum pattern size for drawing
revs	draw function
type	pattern types: defined in “panel.h”
	DrawsArc
	DrawsLine
	DrawsRect
	-1 DrawString

**Description:**

**Program Level:**

X kernel

**Related Functions:**

**Name:**

DumpScan\_to\_dpy — dump image content to window display

**Synopsis:**

DumpScan\_to\_dpy(Image \*img)

**Arguments:**

img                                  image structure pointer

**Description:**

DumpScan\_to\_dpy() map img->data buffer content to window display. The img must be either set up bychoose\_scanline\_converter() or created by BuildColorImage() routine.

**Program Level:**

kernel

**Related Functions:**

BuildColorImage(), choose\_scanline\_converter()

**Name:**

EraseSlider — hide a slider from its panel

**Synopsis:**

EraseSlider(Slider \*s)

**Arguments:**

s slider pointer

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

FillDGT — \*\* fill Digitizer Table (Quantization Table)

**Synopsis:**

```
FillDGT(      int      n,  
           int      *dgtp      )
```

**Arguments:**

n	number of elements int quantization table
dgtp	quantization table pointer

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

Find\_3\_min\_max — find minimum and maximum value in color histogram

**Synopsis:**

```
Find_3_min_max(      U_IMAGE    *img,  
                     int         *histp,  
                     byte        *datap,  
                     bool        nouse      )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

Find\_min\_max — find minimum and maximum value in image data

**Synopsis:**

```
Find_min_max(U_IMAGE *img,  
             int      *histp,  
             byte     *dp,  
             bool     histocalc,  
             bool     permanent )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

FlushingCursor — flushing cursor generator

**Synopsis:**

```
void
FlushingCursor(AnyWindow      *aw,
               int             (*exp_hd)(),
               Image            **imgp,
               int              imgs,
               int              x0, y0,
               int              cw, ch,
               int              fdelay,
               int              rev
               )
```

**Arguments:**

aw	any window pointer
exp_hd	exposure handle routine pointer
imgp	image array pointer for exposure during Flushing
imgs	number images in image array
x0, y0	flushing cursor origin
cw, ch	cursor size
fdelay	delay factor
rev	reverse cursor color instead clear cursor during flushing

**Description:**

*FlushingCursor()* generates a flushing cursor in a X window for prompt input or noticing.

**Program Level:**

kernel

**Related Functions:**

Delay\_Clear(), TextLine()



**Name:**

GetCloseColor — get close color from image color map

**Synopsis:**

```
PColor
GetCloseColor( Display *dpy,
               Colormap cmap,
               int ncolors,
               XColor *uct,
               int r, g, b )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

GetVctEntry — get the 1st available entry in given color map

**Synopsis:**

```
PColor
GetVctEntry( Display *dpy,
             int screen,
             Colormap cmap,
             bool set_gray )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

LinearQuantization — generate linear color map table

**Synopsis:**

```
void
LinearQuantization(      XColor      *cp,
                        int          totalevel,
                        int          cflag,
                        bool         rev
                        )
```

**Arguments:**

cp	X color map pointer
totalevel	number of colors
cflag	DoRed, DoGreen...
rev	reverse color indices

**Description:**

```
void
LinearQuantization(cp, totalevel, cflag, rev)
register XColor *cp;
register int totalevel, cflag;
{
    register int i, gray, scale = 65536 / totalevel;

    for (i=0; i<totalevel; i++) {
        if (rev)
            gray = totalevel - i - 1;
        else gray = i;
        cp[i].pixel = gray; /* for new map only */
        cp[i].red = cp[i].green = cp[i].blue = scale * gray;
        cp[i].flags = cflag;
    }
}
```

**Program Level:**

kernel

**Related Functions:**

**Name:**

LoadGXImage — load an image from a file and display it on X window

**Synopsis:**

```
LoadGXImage( int      *num_images,
               Image    *parent, **pic[],
               bool     movie_mode,
               char      *win_geom,
               bool     still_win      )
```

**Arguments:**

num_images	a pointer of an image count
parent	any image created earlier. It should contain visual class, display, gamma, default name, input file stream pointer, and pixmap (pixmap_failed) information for new images.
pic	picture array address pointer points to image array address.
movie_mode	True if need to run imaes in a movie mode.
win_geom	window geometry string to change display window size. NULL is for default.
still_win	when a new image is required, it will be displaied in the same window as the previous image. The window will be resized.

**Description:**

*LoadGXImage()* loads an image or a sequence of images from one file and displays them on a X terminal or monitor. It requires a non-empty parent to pass many information to new images; otherwise, the new image may not be mapped correctly onto the display. For usage, see getx.c in program guide.

**Return Value:** Positive value if loaded image is a multi-frame image, and zero for a single fram image. The negtive value indicates nothing is loaded. If fatal error happens, it will exit to OS.

**Program Level:**

kernel

**Related Functions:**

get\_pic(), RLE library

**Name:**

LoadIcon — load image icon X window data buffer

**Synopsis:**

```
LoadIcon(      Image      *ip      )
```

**Arguments:**

ip                      image structure pointer

**Description:**

*LoadIcon()* maps an image data content into its icon data buffer for icon display.

**Program Level:**

X kernel

**Related Functions:**

**Name:**

Maintain\_Flush — \* keep all image dumped to screen to be flushed

**Synopsis:**

```
Maintain_Flush(int      cur_img,  
                Image    *previous,  
                Image    **imgp,  
                int      image_y      )
```

**Arguments:**

cur_img	current image index to image array
previous	previous image pointer for movie mode
imgp	image array pointer
image_y	current image display line position

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

MapPixWindow — show pixel information window

**Synopsis:**

```
void
MapPixWindow(Image      *img,
               bool      use_top,
               int        win_bg      )
```

**Arguments:**

img	image structure pointer
use_top	force to use top of the window; otherwise, the window will be opposite the cursor position in the window. e.g. use bottom part if cursor is at top half of the window.
win_bg	pixel window background

**Description:****Program Level:**

X user

**Related Functions:**

**Name:**

MapRGB — map an image line into X window data buffer and maintain the display

**Synopsis:**

```
MapRGB(    int      cur_img,
           Image     *previous,
           Image     **imgp,
           byte      *data_buf[],
           byte      *save_scan[],
           int       y, w,
           int       image_y,
           int       icon_factor    )
```

**Arguments:****Description:****Program Level:**

X interface

**Related Functions:**

Map\_Scanline()



**Name:**

Map\_Scanline — map ILL scan line into X window data buffer

**Synopsis:**

```
Map_Scanline( Image      *img,  
              byte       *data_buf[],  
              byte       *save_scan[],  
              int        x0,  
              int        y,  
              int        w,  
              int        icon_factor    )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

OnButton — check if mouse is clicked on a button

**Synopsis:**

```
OnButton(      Button      *b,  
           XButtonEvent *xbutton  )
```

**Arguments:**

b	any button pointer
xbutton	any X event pointer

**Description:****Program Level:**

Panel kernel

**Related Functions:**

**Name:**

OnScrollBar — if mouse is clicked on a scrollbar

**Synopsis:**

```
OnScrollBar( ScrollBar *sb,  
             XButtonEvent *xbutton)
```

**Arguments:**

sb	scrollbar pointer
xbutton	any X event pointer

**Description:****Program Level:**

panel kernel

**Related Functions:**

**Name:**

OnSliderBar — if mouse is clicked on a slider

**Synopsis:**

```
OnSliderBar(  Slider      *s,  
             XButtonEvent *xbutton)
```

**Arguments:**

s	slider pointer
xbutton	any X event pointer

**Description:****Program Level:**

panel kernel

**Related Functions:**

**Name:**

PanelMessage — display message onto right area of panel parts or onto given place

**Synopsis:**

```
PanelMessage( AnyParts *pp,  
              char      *pmsg,  
              int        x0, y0,  
              int        len      )
```

**Arguments:**

pp	any panel part pointer
pmsg	message string
x0, y0, len	display origin and area length. If one of them is passed by zero, then, that value will be set to parts->x, parts->y or length of the right area of parts, respectively

**Description:****Program Level:**

panel kernel

**Related Functions:**

**Name:**

ParameterWin — display pixel information in an image window

**Synopsis:**

```
void  
ParameterWin( Image      *ip,  
              HistoInfo *hinf,  
              int        x, y,  
              int        y0,  
              bool       is_hist      )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

SetParameterWin()

**Name:**

PopingMenu — pop a pop menu

**Synopsis:**

```
PopingMenu( PopMenu *pm,  
            int      n_th,  
            void      (*expos)(),  
            Image     **imgp,  
            int        nimg      )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

ReadButton — get button name

**Synopsis:**

```
char*  
ReadButton( Button *b )
```

**Arguments:**

b any button pointer

**Description:****Program Level:**

panel kernel

**Related Functions:**



**Name:**

ReadSlider — read slider bar position (value)

**Synopsis:**

```
ReadSlider(  Slider  *s,  
            int    who      )
```

**Arguments:**

s	slider pointer
who	which bar (bar ID)

**Description:****Program Level:**

panel kernel

**Related Functions:**

**Name:**

ResetPressButton — make pressed pressbutton unpressed

**Synopsis:**

```
void  
ResetPressButton(      PressButton    *pb      )
```

**Arguments:**

pb                    press button pointer

**Description:****Program Level:**

panel kernel

**Related Functions:**

**Name:**

ResizeWindow — resize window size

**Synopsis:**

```
ResizeWindow(Image *img,  
              XConfigureEvent *cevent )
```

**Arguments:**

img	image structure pointer
cevent	any X event structure pointer

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

SetFontGetSize — set font in a window, get size into img structure and return font ID

**Synopsis:**

```
Font  
SetFontGetSize(AnyWindow*aw,  
               char      *fontname    )
```

**Arguments:**

aw	any window pointer
fontname	font name string

**Description:****Program Level:**

X kernel

**Related Functions:**

**Name:**

SetParameterWin — set pixel information window

**Synopsis:**

```
SetParameterWin(Image    *img,  
                 XButtonEvent *xbutton,  
                 int      Font_Height,  
                 bool      use_top    )
```

**Arguments:****Description:****Program Level:**

X user

**Related Functions:**

**Name:**

SetSBarPos — set Slider Bar position by point on panel

**Synopsis:**

```
SetSBarPos(  Slider      *s,  
             int         x, y,  
             int         bar_id      )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

SetSBarRPos — set Slider Bar in value (p) position

**Synopsis:**

```
SetSBarRPos( Slider    *s,  
             int       p,  
             int       bar_id      )
```

**Arguments:**

s	slider pointer
p	bar position in value (not x or y)
bar_ib	bar ID 0, 1, 2, ...

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

SetScrollBar — set scrollbar position

**Synopsis:**

```
void
SetScrollBar( ScrollBar *sb,
              int      x, y,
              int      type      )
```

**Arguments:**

sb	scroll bar pointer
x, y	position (in pixels from 0) the bar to be set
type	vertical or horizontal bar

**Description:****Program Level:**

kernel

**Related Functions:**



**Name:**

SetScrollBarLength — set scrollbar length

**Synopsis:**

```
void
SetScrollBarLength(      ScrollBar      *sb,
                        int              vert,
                        int              len
                        )
```

**Arguments:**

sb	scrollbar pointer
vert	vertical bar (False for horizontal)
len	the new scroll length

**Description:**

SetScrollBarLength() changes the scrollbar length to fit window changes.

**Program Level:**

kernel

**Related Functions:**

**Name:**

Set\_Panel — set panel part properties

**Synopsis:**

```
void
Set_Panel(    VType    *ip,
             int       item,
             int       op1, op2, op3  )
```

**Arguments:**

ip	any panel item pointer
item	what to be set
	BUTTON_COLOR
	PRESS_BUTTON_COLOR
	SLIDER_COLOR
op1, op2, op3	operands

**Description:**

see set\_button\_color(), set\_pressbutton\_color(), set\_slider\_color()

**Program Level:**

kernel

**Related Functions:**

set\_button\_color(), set\_pressbutton\_color(), set\_slider\_color()

**Name:**

SliderInfo — draw slider information message on the slider

**Synopsis:**

```
SliderInfo(  Slider  *s,  
            int      color      )
```

**Arguments:**

s	slider pointer
color	message color

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

TextLine — text line editor for panel input

**Synopsis:**

```
TextLine(      Button      *b,
              char        *buf,
              int         len,
              int         x0,
              int         (*exp_hd)(),
              Image       **imgp,
              int         imgs)
```

**Arguments:**

b	button on this panel
buf	input buffer to hold input string
len	maximum buffer length. Once the input string length exceeding this length, then return.
x0	x offset to panel information area which is at the bottom of the panel. If x0 is zero, then, use the area at right side of the button (b).
exp_hd	exposure handler for waiting input
imgp	image array pointer
imgs	number of images in image array to be handled

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

TopWindow — put window on the top of the display stack

**Synopsis:**

```
TopWindow( AnyWindow *aw,  
            bool      use_unmap )
```

**Arguments:**

aw	any image or window pointer
use_unmap	use unmap and map to make window top. The default is to clear window and raise it.

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

TrackSubWin — Crop a sub-window in different sharp

**Synopsis:**

```
TrackSubWin( Image      *img,
              HistoInfo *hinf,
              int        ox, oy,
              int        shp,
              int        cropButton )
```

**Arguments:**

img	image structure pointer
hinf	histogram information (NULL is OK)
ox, oy	origin to trace
shp	sub-window sharp; see Draws()
cropButton	which button for crop

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

WhichImage — see cursor is in which image window

**Synopsis:**

```
WhichImage( Window win,
             Image **imgp,
             int    imgs      )
```

**Arguments:**

win	X window ID
imgp	image structure pointer array
imgs	number of images in pointer array

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

XCopyImage — copy a X image region to another region

**Synopsis:**

```
XCopyImage( XImage  *img,  
            int      src_x, src_y,  
            int      width, height,  
            int      dst_x, dst_y  )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**



**Name:**

check\_pixmap\_allocation — set error handler

**Synopsis:**

```
void  
check_pixmap_allocation( Image      *img      )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

choose\_scanline\_converter — find appropriate scan line routine to use

**Synopsis:**

```
void  
choose_scanline_converter( Image      *img      )
```

**Arguments:**

img                      image structure pointer

**Description:****Program Level:**

X assemble

**Related Functions:**

BuildColorImage(), CreateImage()

**Name:**

create\_windows — create X window for image display

**Synopsis:**

```
void  
create_windows(Image      *img,  
                char      *window_geometry,  
                bool      stingy  
                )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

eq\_cmap — compare two color maps

**Synopsis:**

```
eq_cmap(    rle_pixel  **cmap1,
            int         c1_len,
            rle_pixel  **cmap2,
            int         c2_len    )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

exposure\_r — exposure event handler

**Synopsis:**

```
exposure_r(  Image    *img,
             int      (*sub_win)(),
             int      x, y,
             int      w, h,
             int      cdp      )
```

**Arguments:**

img	image structure pointer
sub_win	the routine pointer to handle sub-window and superimposed elements and draw patterns
x, y	exposure area origin
w, h	exposure area size
cdp	copy from pixel map if it exists

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

find\_appropriate\_visual — find appropriate visual type

**Synopsis:**

```
void  
find_appropriate_visual( Image *img )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

find\_min\_max — find minimum and maximum value in a region

**Synopsis:**

```
find_min_max( byte    *bufp,  
              int      *hist,  
              Mregister *mmm,  
              int      r_width, r_height,  
              int      width      )
```

**Arguments:**

bufp	data buffer
hist	histogram buffer pointer
mmm	min, max, and max count register
r_width, r_height	region width and height
width	line (image) width

**Description:**

find\_min\_max() computes the min & max value in a region of data buffer.

**Program Level:**

kernel

**Related Functions:**

**Name:**

free\_unique\_colors — \* free system color map space

**Synopsis:**

```
free_unique_colors(      Image      *img,  
                       Pixel      *pixels,  
                       int         npixels      )
```

**Arguments:****Description:**

free\_unique\_colors() frees system color map space allocated by XAllocColorCells(), XAllocColor(), or XAllocColors().

**Program Level:**

kernel

**Related Functions:**



**Name:**

get\_X\_image — create X image for image display

**Synopsis:**

```
XImage*
get_X_image( Image    *img,
              int      width, height,
              bool      buf_need    )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

get\_cursors — create cursors for a window

**Synopsis:**

```
void  
get_cursors( Window window )
```

**Arguments:**

window                      X window ID

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

get\_dither\_arrays — allocate image dither arrays

**Synopsis:**

```
void  
get_dither_arrays( Image *img )
```

**Arguments:**

img	image structure pointer
-----	-------------------------

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

get\_dither\_colors — sets in\_cmap, cmlen, ncmap and mono\_color in an image structure

**Synopsis:**

```
void  
get_dither_colors( Image *img )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

get\_iconsize — get appropriate icon size

**Synopsis:**

```
get_iconsize(  Image      *img,  
               int        width      )
```

**Arguments:**

img	image structure pointer
width	icon width

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

`get_pic` — build a image and its display window from open files

**Synopsis:**

```
get_pic(      int      cur_img,
              char      *win_geom,
              Image     *prev_img,
              Image     **imgp,
              int      img_win_bg  )
```

**Arguments:**

<code>cur_img</code>	current image index to image structure array
<code>win_geom</code>	window geometry string for changing window size (NULL for no change)
<code>prev_img</code>	movie pointer point to previous frame
<code>imgp</code>	image structure array pointer
<code>img_win_bg</code>	image window background color

**Description:**

`get_pic()` loads different images through CCS interface, converts the loaded image into ILC format, and creates X window to display it. `get_pic()` contains two global variables: *shared\_cmap* and *multi\_hd*. On pseudo color devices, if *shared\_cmap* is set to True, then most images will converted to pseudo color formats and recalibrated to share same color with the image first loaded. The *multi\_hd* variable is used for RLE like images that multi-frame images are constructed by concatenating all single images together.

**Return Value:** It returns 0 when an image is successfully loaded, or negative value on failures.

**Program Level:**

X interface

**Related Functions:**

`BuildColorImage()`, `bool shared_cmap`, `bool multi_hd`

**Name:**

get\_x\_colormap — create X color map

**Synopsis:**

```
void  
get_x_colormap( Image *img )
```

**Arguments:****Description:**

equivalent to XCreateColormap()

**Program Level:**

kernel

**Related Functions:**

**Name:**

handle\_exposure — image exposure handler

**Synopsis:**

```
void  
handle_exposure(Image *img,  
                 int (*sub_win)(),  
                 int x, y,  
                 int width, height,  
                 int img_h,  
                 bool tuner_flag )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**



**Name:**

init\_color — initial color map

**Synopsis:**

```
void  
init_color(      Image      *img      )
```

**Arguments:**

img                      image structure pointer

**Description:****Program Level:**

X assemble

**Related Functions:**

**Name:**

init\_img\_info — initial image structure content

**Synopsis:**

```
void
init_img_info( Image      *i,
               Display    *dpy,
               int         mid_type,
               bool        color_dpy )
```

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

interpolation — interpolate an image by regions

**Synopsis:**

```
interpolation(  byte      *in, *out,
                int       x_regions, y_regions,
                InterpMap *imp,
                Image      *img,
                HistoInfo *hinf          )
```

**Arguments:**

in	original data buffer
out	interpolated data buffer
x_regions	divisor in X dimension
y_regions	divisor in Y dimension
imp	interpolation information structure
<pre>typedef struct {     int min, max, diff; /* union with Mregister */     LKT *lkt; } InterpMap;</pre>	
img	image to be interpolated
hinf	image histogram information

**Description:****Program Level:**

kernel

**Related Functions:**

histogram(), new\_curve()

**Name:**

mag\_pan — zoom an image

**Synopsis:**

```
void
mag_pan(      Image      *img,
             int          job,
             int          bx, by,
             int          mag_factor,
             bool         stingy      )
```

**Arguments:**

img	image structure pointer
job	zooming, shifting, or switching mode
bx, by	position the button press event happened
mag_factor	zooming factor (negative is zoom down)
stingy	slow but use less memory

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

map\_1\_to\_3 — map 8-bit color image to a ILL (RLE) image

**Synopsis:**

```
VType*
map_1_to_3(  char    *src, *dst,
              cmap_t **cmap,
              int     w, h      )
```

**Arguments:**

src	source data buffer
dst	result data buffer
cmap	color map pointer
w, h	image size

**Description:****Program Level:**

kernel

**Related Functions:**

snf\_to\_rle()

**Name:**

map\_rgb\_to\_bw — map RGB to black-and-white through NTSC transform

**Synopsis:**

```
map_rgb_to_bw(Image      *img,
               rle_pixel **color_row,
               VType      *bw_row,
               int         rowlen )
```

**Arguments:**

img	image structure pointer
color_row	rgb data row buffer
bw_row	black-and-white row buffer
rowlen	scanline width

**Description:****Program Level:**

kernel

**Related Functions:**

map\_rgb\_to\_rgb()

**Name:**

map\_rgb\_to\_rgb — convert RGB to RGB through a colormap

**Synopsis:**

```
void
map_rgb_to_rgb(Image      *img,
               rle_pixel **in_row, **out_row,
               int         rowlen
               )
```

**Arguments:**

img	image structure pointer
in_row	original row buffer
out_row	out going row buffer
rowlen	buffer length (scanline width)

**Description:****Program Level:**

kernel

**Related Functions:**

map\_rgb\_to\_bw()

**Name:**

new\_curve — ETA curve generator for X program

**Synopsis:**

```
new_curve(  LKT      *lkt,
            HistoInfo *hinf,
            Mregister *mmm,
            int       type,
            int       Maxout,
            Image     *img,
            int       hsize
            )
```

**Arguments:**

lkt	lookup table to hold new curve map
hinf	histogram information pointer
mmm	min, max, pixel value & max count register
type	curve type
Maxout	maximum output value for pixel mapping
img	image structure pointer
hsize	histogram buffer size

**Description:****Program Level:**

kernel

**Related Functions:**

find\_min\_max()



**Name:**

on\_superimpose\_elem — if mouse clicked on a superimposed element

**Synopsis:**

```
on_superimpose_elem(    Image    *img,  
                       int        x0, y0    )
```

**Arguments:**

img	image structure pointer
x0, y0	mouse clicked position

**Description:**

If place mouse clicked contains some superimposed elements, on\_superimpose\_elem() returns the top one's ID; otherwise, returns 0.

**Program Level:**

kernel

**Related Functions:**

superimpose\_handle()

**Name:**

set\_button\_color — set button colors

**Synopsis:**

```
set_button_color(      Button      *b,
                      int          button_color,
                      int          pressed_color,
                      int          title_color      )
```

**Arguments:**

b	button pointer
button_color	button color
pressed_color	button pressed color
title_color	button name color

**Description:**

**Program Level:**

kernel

**Related Functions:**

**Name:**

set\_circle\_cursor — set window cursor to circle cursor

**Synopsis:**

```
void  
set_circle_cursor( Window win )
```

**Arguments:**

win                                  X Window ID

**Description:**

give window a circle cursor.

**Program Level:**

kernel

**Related Functions:**

set\_left\_ptr\_cursor(), set\_watch\_cursor()

**Name:**

set\_left\_ptr\_cursor — set window cursor to left pointer cursor

**Synopsis:**

```
void  
set_left_ptr_cursor( Window win )
```

**Arguments:**

win                      X Window ID

**Description:**

give window a left pointer cursor

**Program Level:**

kernel

**Related Functions:**

set\_circle\_cursor(), set\_watch\_cursor()

**Name:**

set\_pressbutton\_color — set pressbutton color

**Synopsis:**

set\_pressbutton\_color

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

set\_slider\_color — set slider colors

**Synopsis:**

set\_slider\_color

**Arguments:****Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

set\_timer — set timer for wait\_timer()

**Synopsis:**

```
void  
set_timer(    unsigned    microseconds    )
```

**Arguments:**

microseconds                  time to be wake up

**Description:****Program Level:**

kernel

**Related Functions:**

wait\_timer()

**Name:**

set\_watch\_cursor — set window cursor to watch cursor

**Synopsis:**

```
set_watch_cursor( Window win )
```

**Arguments:**

win                                  X window ID

**Description:**

give window a watch cursor.

**Program Level:**

kernel

**Related Functions:**

set\_circle\_cursor(), set\_left\_ptr\_cursor()



**Name:**

set\_window\_cursor — set window cursor

**Synopsis:**

```
void  
set_window_cursor(      Window      window,  
                      Cursor      cursor_type      )
```

**Arguments:**

window	X Window ID
cursor_type	X Cursor type

**Description:**

give window a cursor in X Cursor domain.

**Program Level:**

kernel

**Related Functions:**

set\_circle\_cursor(), set\_left\_ptr\_cursor(), set\_watch\_cursor()

**Name:**

superimpose\_add\_elem — add draw element to an image

**Synopsis:**

```
superimpose_add_elem(  U_IMAGE    *img,
                       int         line_w,
                       int         color,
                       int         shape,
                       int         y, x,
                       int         rows, cols    )
```

**Arguments:**

img	image structure pointer
line_w	pencil line wideness
color	pencil color
sharp	pattern sharp
y, x	pattern origin (center for ARCs)
rows, cols	pattern box size (quart for ARCs)

**Description:****Program Level:**

kernel

**Related Functions:**

**Name:**

superimpose\_handle — superimposed element moving handler

**Synopsis:**

```
superimpose_handle(    Image    *img,  
                      int        who,  
                      int        x, y    )
```

**Arguments:**

img	image structure pointer
who	which element need to be moved in superimposing stacks. It should be gotten from on_superimpose_elem(), and it must be non-zero. Tips: negative for drawing string, and positive for drawing patterns.
x, y	position the mouse button pressed

**Description:**

superimpose\_handle() used to move a

**Program Level:**

kernel

**Related Functions:**

on\_superimpose\_elem(), superimpose\_images()

**Name:**

superimpose\_images — draws pattern onto an image

**Synopsis:**

```
void
superimpose_images(    U_IMAGE    *img,
                       int          who,
                       int          x0, y0,
                       int          cw, ch          )
```

**Arguments:**

img	image structure pointer
who	pattern type; see Draws()
x0, y0	drawing origin
cw, ch	pattern box width and height

**Description:**

superimpose\_images() does similar pattern as Draws(), but does not draw the patterns into the image (onto).

**Program Level:**

kernel

**Related Functions:**

Draws()

**Name:**

wait\_timer — wait time out set by set\_timer)(

**Synopsis:**

```
void  
wait_timer(    void    )
```

**Arguments:**

None

**Description:**

sleep until time out

**Program Level:**

kernel

**Related Functions:**

set\_timer()

**Name:**

win\_exposure — image window exposure handler

**Synopsis:**

```
win_exposure( XExposeEvent      *expose,
              Image             *img,
              void              (*sub_win)() )
```

**Arguments:**

expose	X exposure event pointer
img	image whose window generates exposure event
sub_win	routine to handle pattern draws on the image

**Description:****Program Level:**

X user

**Related Functions:**

exposure\_r()

# INDEX

## **B**

### **button**

ButtonPressed 4  
 CreateButton 6  
 CreatePressButton 11  
 DrawButton 18  
 DrawPressButton 20  
 OnButton 42  
 ReadButton 48  
 ResetPressButton 50  
 set\_button\_color 90  
 Set\_Panel 58  
 set\_pressbutton\_color 93

## **C**

check\_pixmap\_allocation 65  
 choose\_scanline\_converter 66

### **color**

BuildColorImage 3  
 CreateCLT 7  
 DumpScan\_to\_dpy 27  
 eq\_cmap 68  
 FillDGT 29  
 free\_unique\_colors 72  
 get\_dither\_arrays 75  
 get\_dither\_colors 76  
 get\_x\_colormap 79  
 GetCloseColor 33  
 init\_color 81  
 LinearQuantization 35  
 map\_1\_to\_3 85  
 map\_rgb\_to\_bw 86  
 map\_rgb\_to\_rgb 87  
 Map\_Scanline 41  
 MapRGB 40  
 create\_windows 67  
 CreateCLT 7  
 CreatePanel 9  
 CreateWindow 14

### **cursor**

Delay\_Clear 15  
 FlushingCursor 32

get\_cursors 74  
 set\_circle\_cursor 91  
 set\_left\_ptr\_cursor 92  
 set\_watch\_cursor 96  
 set\_window\_cursor 97

## **D**

Delay\_Clear 15  
 DestroyImage 16  
 DispInfo 17  
 SetParameterWin 53  
 TrackSubWin 62  
 DrawPixWindow 19  
 Draws 26  
 DrawVMark 24  
 DumpScan\_to\_dpy 27

## **E**

### **editor**

TextLine 60  
 eq\_cmap 68  
 exposure\_r 69

## **F**

FillDGT 29  
 Find\_3\_min\_max 30  
 find\_appropriate\_visual 70  
 Find\_min\_max 31  
 find\_min\_max 71  
 Maintain\_Flush 38  
 FlushingCursor 32  
 SetFontGetSize 52  
 free\_unique\_colors 72

## **G**

get\_cursors 74  
 get\_dither\_arrays 75  
 get\_dither\_colors 76  
 get\_ysize 77  
 get\_pic 78  
 get\_x\_colormap 79  
 get\_X\_image 73  
 GetCloseColor 33

GetVctEntry 34

## **H**

### **histogram**

handle\_exposure 80  
Find\_3\_min\_max 30  
Find\_min\_max 31  
find\_min\_max 71

## **I**

get\_iconsize 77  
LoadIcon 37  
BuildColorImage 3  
ChangeSliderScale 5  
CreateCLT 7  
CreateImage 8  
CreateScrollBar 12  
DestroyImage 16  
Draw\_ImageScrollBars 25  
DrawPixWindow 19  
Draws 26  
DrawSpeedWindow 23  
DrawVMark 24  
DumpScan\_to\_dpy 27  
get\_pic 78  
get\_X\_image 73  
GetCloseColor 33

### **image**

init\_img\_info 82  
interpolation 83  
LoadGXImage 36  
LoadIcon 37  
mag\_pan 84  
Maintain\_Flush 38  
map\_1\_to\_3 85  
MapPixWindow 39  
on\_superimpose\_elem 89  
ParameterWin 46  
SetFontGetSize 52  
SetParameterWin 53  
superimpose\_add\_elem 98  
superimpose\_handle 99  
superimpose\_images 100  
WhichImage 63

XCopyImage 64  
init\_color 81  
init\_img\_info 82  
interpolation 83

## **L**

LinearQuantization 35  
LoadGXImage 36  
LoadIcon 37

## **M**

mag\_pan 84  
Maintain\_Flush 38  
choose\_scanline\_converter 66  
DumpScan\_to\_dpy 27  
eq\_cmap 68  
free\_unique\_colors 72  
get\_dither\_colors 76  
get\_x\_colormap 79  
GetCloseColor 33  
map\_1\_to\_3 85  
map\_rgb\_to\_bw 86  
map\_rgb\_to\_rgb 87  
Map\_Scanline 41  
MapRGB 40  
new\_curve 88  
map\_1\_to\_3 85  
map\_rgb\_to\_rgb 87  
MapPixWindow 39  
MapRGB 40  
CreatePopupMenu 10  
menu 2  
PopingMenu 47  
PanelMessage 45

### **movie**

DrawSpeedWindow 23

## **N**

new\_curve 88

## **O**

on\_superimpose\_elem 89  
OnButton 42



OnScrollBar 43  
OnSliderBar 44

## ***P***

### **panel**

ButtonPressed 4  
ChangeSliderScale 5  
CreateButton 6  
CreatePanel 9  
CreatePopupMenu 10  
CreatePressButton 11  
CreateSlider 13  
DispInfo 17  
DrawButton 18  
DrawPressButton 20  
DrawSlider 22  
EraseSlider 28  
OnButton 42  
OnSliderBar 44  
PanelMessage 45  
PoppingMenu 47  
ReadButton 48  
ReadSlider 49  
ResetPressButton 50  
set\_button\_color 90  
Set\_Panel 58  
set\_pressbutton\_color 93  
set\_slider\_color 94  
SetSBarPos 54  
SetSBarRPos 55  
SliderInfo 59  
PanelMessage 45  
ParameterWin 46  
PoppingMenu 47

## ***Q***

FillDGT 29  
get\_dither\_arrays 75  
get\_dither\_colors 76  
interpolation 83  
LinearQuantization 35  
new\_curve 88

## ***R***

ReadButton 48  
ReadSlider 49  
ResizeWindow 51

## ***S***

### **scrollbar**

CreateScrollBar 12  
Draw\_ImageScrollBars 25  
DrawScrollBars 21  
OnScrollBar 43  
SetScrollBar 56  
SetScrollBarLength 57  
set\_button\_color 90  
set\_circle\_cursor 91  
set\_left\_ptr\_cursor 92  
set\_pressbutton\_color 93  
set\_slider\_color 94  
set\_timer 95  
set\_watch\_cursor 96  
SetFontGetSize 52  
SetParameterWin 53

### **slider**

SetSBarPos 54  
SetSBarRPos 55  
SetScrollBar 56  
SetScrollBarLength 57  
ChangeSliderScale 5  
CreateSlider 13  
DrawSlider 22  
EraseSlider 28  
OnSliderBar 44  
ReadSlider 49  
Set\_Panel 58  
set\_slider\_color 94  
SetSBarPos 54  
SetSBarRPos 55  
SliderInfo 59  
superimpose\_add\_elem 98  
superimpose\_handle 99  
superimpose\_images 100

**T**

TextLine 60  
 set\_timer 95  
 wait\_timer 101  
 TopWindow 61  
 TrackSubWin 62

**V****value**

ReadButton 48  
 ReadSlider 49  
 SetSBarPos 54  
 SetSBarRPos 55  
 TextLine 60

**W**

wait\_timer 101  
 WhichImage 63

**window**

win\_exposure 102  
 create\_windows 67  
 CreateImage 8  
 CreateWindow 14  
 Delay\_Clear 15  
 DispInfo 17  
 Draw\_ImageScrollBars 25  
 DrawPixWindow 19  
 DrawScrollBars 21  
 DrawSpeedWindow 23  
 DrawVMark 24  
 exposure\_r 69  
 FlushingCursor 32  
 get\_cursors 74  
 LoadGXImage 36  
 LoadIcon 37  
 MapPixWindow 39  
 OnScrollBar 43  
 ParameterWin 46  
 ResizeWindow 51  
 set\_circle\_cursor 91  
 set\_left\_ptr\_cursor 92  
 set\_watch\_cursor 96  
 set\_window\_cursor 97

SetFontGetSize 52  
 SetParameterWin 53  
 SetScrollBar 56  
 SetScrollBarLength 57  
 TextLine 60  
 TopWindow 61  
 TrackSubWin 62  
 win\_exposure 102

**X**

check\_pixmap\_allocation 65  
 create\_windows 67  
 CreateWindow 14

**X**

exposure\_r 69  
 find\_appropriate\_visual 70  
 get\_pic 78  
 get\_x\_colormap 79  
 get\_X\_image 73  
 handle\_exposure 80  
 init\_color 81  
 LoadGXImage 36  
 LoadIcon 37  
 Map\_Scanline 41  
 MapRGB 40  
 SetFontGetSize 52  
 XCopyImage 64  
 GetVctEntry 34  
 XCopyImage 64

**Z**

mag\_pan 84