Chapter 5

CMX11 Man Pages

This chapter contains UNIX-style manual pages for the CMX11 library routines.

NOTE

If you are not using display functions, you should replace the C* header file

```
#include <cm/cmxdisplay.h>
```

with

```
#include <cm/cmx.h>
```


Copyright © 1993 Thinking Machines Corporation

CMXAllocateMousePoint

Allocates a mouse point.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXAllocateMousePoint()
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
CMXMousePoint_t
CMXAllocateMousePoint(void);
```

ARGUMENTS

None.

DESCRIPTION

`CMXAllocateMousePoint` allocates a mouse point. A mouse point is a structure in which information about the mouse's status is stored, including its coordinates and which button or key was last pressed.

SEE ALSO

`CMXDeallocateMousePoint`

Copyright © 1993 Thinking Machines Corporation

CMXAllocMBuffers

Allocates image buffers for double buffering.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXAllocMBuffers(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Multibuffer*
CMXAllocMBuffers(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXAllocMBuffers makes a double buffer for the display associated with *widget* and returns an identifier for the multibuffer. You can pass the multibuffer return value as the *buffers* argument to **CMXGetDrawMBuffer**, **CMXGetDisplayMBuffer**, and **CMXSwitchMBuffers**, to identify and control the double buffers.

The display's workstation must have special hardware to support double buffering. You can determine whether a display supports double buffering by calling **CMXHasMBuffers**. If you write your program to use the multibuffer routines and the display does not support true double buffering, CMX11 redraws the image into the display window as if no double buffering had been attempted.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXFreeMBuffers
CMXGetDisplayMBuffer
CMXGetDrawMBuffer
CMXHasMBuffers
CMXSwitchMBuffers

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXAllocateColors

Allocates cells for your application in the display colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXAllocateColors(display, cmap, ncolors, base)
   INTEGER display
   INTEGER cmap
   INTEGER ncolors   ! IN/OUT
   INTEGER base      ! OUT
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXAllocateColors(Display *display, Colormap cmap,
                  int *ncolors, int *base);
```

ARGUMENTS

display	An integer identifying the X11 display for which you wish to allocate colors. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.
cmap	An integer identifying the X11 **Colormap** on *display* for which you wish to allocate colors. The value for this argument is returned for a CMX11 display by **CMXGetColormap**.
ncolors	As input, the number of colors you wish to allocate. On return, the number of color cells actually allocated.
base	An integer set by the function to the entry number of the first color cell allocated to you. Use this number as an offset when specifying your colors.

Copyright © 1993 Thinking Machines Corporation

DESCRIPTION

CMXAllocateColors attempts to reserve *ncolors* consecutive color cells in the display colormap for your use. On return, the routine sets the *base* parameter to the first color cell you are allocated and sets *ncolors* to the number of color cells actually allocated to you. If no color cells are available, the routine sets *ncolors* to 0. Once the locations are allocated you may assign color values to them with **CMXStoreColors** or **CMXStoreColor**.

Note that the colormap is shared among all the windows using the display. The cells you request are allocated from the unused cells and may begin anywhere in the colormap. Remember to offset your color values by *base*. To refer to your first color, use *base*; to refer to the second use *base* + 1; etc.

SEE ALSO

CMXStoreColors
CMXStoreColor
CMXLoadColorArrays
CMXFreeColors

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXBitsOfBlue

Returns the number of bits used for blue in RGB colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXBitsOfBlue(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXBitsOfBlue(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXBitsOfBlue returns the number of bits supported for the blue component of the RGB colormap.

This routine is useful for displays that do not allocate an equal number of bits to each color component. For example, an 8-bit display with a TrueColor or Direct-Color visual class may allocate 3 bits for the red component, 3 bits for the green, and 2 bits for the blue component.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXBitsOfRed
CMXBitsOfGreen
CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapSize
CMXColormapBlueSize
CMXColormapGreenSize
CMXColormapRedSize
CMXCreateDisplayClass

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXBitsOfGreen

Returns the number of bits used for green in RGB colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXBitsOfGreen(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXBitsOfGreen(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXBitsOfGreen returns the number of bits supported for the green component of the RGB colormap.

This routine is useful for displays that do not allocate an equal number of bits to each color component. For example, an 8-bit display with a TrueColor or Direct-Color visual class may allocate 3 bits for the red component, 3 bits for the green, and 2 bits for the blue component.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXBitsOfRed
CMXBitsOfBlue
CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapSize
CMXColormapGreenSize
CMXColormapBlueSize
CMXColormapRedSize
CMXCreateDisplayClass

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXBitsOfRed

Returns the number of bits used for red in RGB colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXBitsOfRed(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXBitsOfRed(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXBitsOfRed returns the number of bits supported for the red component of the RGB colormap.

This routine is useful for displays that do not allocate an equal number of bits to each color component. For example, an 8-bit display with a TrueColor or Direct-Color visual class may allocate 3 bits for the red component, 3 bits for the green, and 2 bits for the blue component.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXBitsOfGreen
CMXBitsOfBlue
CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapSize
CMXColormapRedSize
CMXColormapGreenSize
CMXColormapBlueSize
CMXCreateDisplayClass

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXClearDisplay

Fills display with background color or pixmap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXClearDisplay(display, drawable)
   INTEGER display
   INTEGER drawable
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXClearDisplay(Display *display, Drawable drawable);
```

ARGUMENTS

display	An integer identifying the X11 display which you wish to clear. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.
drawable	An integer identifying the window or pixmap on *display* which you wish to clear. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.

DESCRIPTION

CMXClearDisplay clears the window on *display* specified by *drawable*. If the drawable is a window, it is filled with the drawable's background color or pixmap. If the drawable is a pixmap, it is filled with color 0. This routine overwrites all previous contents of the drawable. Note that the drawable's background color is set by **XSetWindowBackground**, not **CMXSetBackground**.

Copyright © 1993 Thinking Machines Corporation

NOTE: When *drawable* is a window, **`CMXClearDisplay`** is equivalent to the X11 routine **`XClearWindow`**. C* programmers can just call the X11 routine.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXColormapBlueSize

Returns the number of entries of blue values in RGB colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXColormapBlueSize(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXColormapBlueSize(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXColormapBlueSize returns the number of entries, or color cells, supported for the blue component of the RGB colormap.

This routine is useful for displays that do not allocate an equal number of bits to each color component. For example, an 8-bit display with a TrueColor or DirectColor visual class may allocate 3 bits for the red component, 3 bits for the green, and 2 bits for the blue component. In this case the red and green maps would have 8 entries while the blue map would have only 4 entries. In addition, some entries may be reserved for the system's use. For example, one entry in the red map might be reserved for the cursor color, leaving only 7 entries available in the red map.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXColormapGreenSize
CMXColormapRedSize
CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapSize
CMXBitsOfRed
CMXBitsOfBlue
CMXBitsOfGreen

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXColormapGreenSize

Returns the number of entries of green values in RGB colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXColormapGreenSize(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXColormapGreenSize(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXColormapGreenSize returns the number of entries, or color cells, supported for the green component of the RGB colormap.

This routine is useful for displays that do not allocate an equal number of bits to each color component. For example, an 8-bit display with a TrueColor or DirectColor visual class may allocate 3 bits for the red component, 3 bits for the green, and 2 bits for the blue component. In this case the red and green maps would have 8 entries while the blue map would have only 4 entries. In addition, some entries may be reserved for the system's use. For example, one entry in the red map might be reserved for the cursor color, leaving only 7 entries available in the red map.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXColormapBlueSize
CMXColormapRedSize
CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapSize
CMXBitsOfRed
CMXBitsOfBlue
CMXBitsOfGreen

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXColormapRedSize

Returns the number of entries of red values in RGB colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXColormapRedSize(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXColormapRedSize(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXColormapRedSize returns the number of entries, or color cells, supported for the red component of the RGB colormap.

This routine is useful for displays that do not allocate an equal number of bits to each color component. For example, an 8-bit display with a TrueColor or DirectColor visual class may allocate 3 bits for the red component, 3 bits for the green, and 2 bits for the blue component. In this case the red and green maps would have 8 entries while the blue map would have only 4 entries. In addition, some entries may be reserved for the system's use. For example, one entry in the red map might be reserved for the cursor color, leaving only 7 entries available in the red map.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXColormapBlueSize
CMXColormapGreenSize
CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapSize
CMXBitsOfRed
CMXBitsOfBlue
CMXBitsOfGreen

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXColormapSize

Returns the number of entries in the colormap.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXColormapSize(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXColormapSize(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXColormapSize returns the number of entries, or color cells, in the display colormap for PseudoColor, StaticColor, GrayScale, or StaticGray visual classes.

This function returns the size of the largest map (red, green, or blue) for TrueColor or DirectColor visual classes.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXIsColor
CMXIsRGB
CMXHasColormap
CMXColormapRedSize
CMXColormapGreenSize
CMXColormapBlueSize
CMXBitsOfRed
CMXBitsOfGreen
CMXBitsOfBlue

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXCreateDisplay

Creates a CMX11 display window connected to a specified X11 display and returns a CMX11 display identifier.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXCreateDisplay(display, vinfo, cmap, width, height)
   INTEGER display
   INTEGER vinfo
   INTEGER cmap
   INTEGER width
   INTEGER height
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
CMX_display_t
CMXCreateDisplay(Display *display, XVisualInfo *vinfo,
    Colormap cmap, int width, int height);
```

ARGUMENTS

display	A pointer to an X11 **Display** structure specifying the X server to be used to support the CMX11 display.
vinfo	A pointer to an X11 **XVisualInfo** structure. The **XVisual-Info** structure is used to define the display visual. It includes specifications for the display's visual class, depth, colormap size, and, for DirectColor and TrueColor displays, the number of bits for each color component.
cmap	A pointer to an X11 **Colormap** structure containing the colormap to be installed for the display.
width	The width of the display window, that is, the number of pixels in the horizontal (x) dimension.

Copyright © 1993 Thinking Machines Corporation

height The height of the display window, that is, the number of pixels in the vertical (y) dimension.

DESCRIPTION

CMXCreateDisplay establishes the X server specified by *display* as a CMX11 display and opens a window sized *width* by *height*, configured according to *vinfo*, and with *cmap* installed as the colormap. **CMXCreateDisplay** returns a CMX11 display identifier.

NOTE: The CMX11 routines **CMXCreateDisplayClass** and **CMXCreateSimpleDisplay** provide an easier interface for creating a CMX11 display. **CMXCreateDisplay** should only be used to establish an existing X11 display as a CMX11 display.

SEE ALSO

CMXCreateDisplayClass
CMXCreateSimpleDisplay

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXCreateDisplayClass

Creates CMX11 X Window System display of a specified visual class and returns a CMX11 display identifier.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXCreateDisplayClass(class, ncolors, x, y)
   INTEGER class
   INTEGER ncolors
   INTEGER x, y
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
CMX_display_t
CMXCreateDisplayClass(int class, int ncolors, int x, int y);
```

ARGUMENTS

class	The visual class you wish for the window created by this routine. *class* may be any of the standard X11 visual classes: **PseudoColor**, **StaticColor**, **DirectColor**, **TrueColor**, **GrayScale**, or **StaticGray**.
ncolors	An integer specifying the number of colors you wish the window to support.
x, *y*	Integers specifying the dimensions, in pixels, of the window you wish to open.

DESCRIPTION

CMXCreateDisplayClass establishes a connection to an X Window System display. The routine matches the value you give for *ncolors* and the visual class specified in *class* to the capabilities of the display to assign an appropriate visual.

Copyright © 1993 Thinking Machines Corporation

The display can be specified in your current **DISPLAY** environment variable or by using the **-display** command line option when you invoke your program. The command line option overrides the environment variable. To use the command line option from a CM Fortran program you must first call **CMXSetArgcArgvF**, and from C* call **CMXSetArgcArgv**.

The function also

- opens a window on the display sized *x* by *y* pixels with the visual class specified by *class*. *x* is the horizontal dimension, and *y* is the vertical dimension.
- creates a graphics context for the display set to a background color of black and the other standard X11 defaults.
- returns an integer that is a CMX11 display identifier.

  You pass this returned value to CMX11 accessor routines to get X11 identifiers used to manage the display. The CMX11 routines that require the CMX11 display identifier are:

  CMXGetDisplay
  CMXGetDrawable
  CMXGetGC
  CMXGetColormap
  CMXGetWidget
  CMXGetVisualInfo

See the man pages for these routines for more information.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXCreateSimpleDisplay

Creates a CMX11 X Window System display and returns a CMX11 display identifier.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXCreateSimpleDisplay(ncolors, x, y)
   INTEGER ncolors
   INTEGER x, y
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
CMX_display_t
CMXCreateSimpleDisplay(int ncolors, int x, int y);
```

ARGUMENTS

ncolors	An integer specifying the number of colors you wish the display to support.
x, *y*	Integers specifying the dimensions, in pixels, of the window you wish to open.

DESCRIPTION

CMXCreateSimpleDisplay establishes a connection to an X Window System display. The routine matches the number of colors specified in *ncolors* to the capabilities of the display to assign an appropriate visual.

The display you want to connect to can be specified in your current **DISPLAY** environment variable or by using the **-display** command line option when you invoke your program. The command line option overrides the environment variable. To use the command line option from a CM Fortran program you must first call **CMXSetArgcArgvF**, and from C* call **CMXSetArgcArgv**.

Copyright © 1993 Thinking Machines Corporation

The function also

- opens a window on the display sized *x* by *y* pixels, where *x* is the horizontal dimension and *y* is the vertical dimension. The system determines an appropriate visual class by matching the number of colors you request in *ncolors* with the capabilities of the display.
- creates a graphics context for the display set to a background color of black and the other standard X11 defaults.
- returns an integer that is a CMX11 display identifier.

  You pass this returned value to CMX11 accessor routines to get X11 identifiers used to manage the display. The CMX11 routines that require the CMX11 display identifier are:

  CMXGetDisplay
  CMXGetDrawable
  CMXGetGC
  CMXGetColormap
  CMXGetWidget
  CMXGetVisualInfo

See the man pages for these routines for more information.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXCurrentMousePoint

Gets the current state of the mouse.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXCurrentMousePoint(widget, point)
   INTEGER widget
   INTEGER point
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXCurrentMousePoint(Widget widget, CMXMousePoint_t point);
```

ARGUMENTS

widget	Specifies a widget.
point	Specifies the point's *x* and *y* coordinates, and its button member.

DESCRIPTION

CMXCurrentMousePoint gets the current state of the mouse; *x* and *y* (in *point*) are the current coordinates, relative to the window, which means they might be negative or larger than the window size if the pointer is not within the window. The button member is filled in with the current state of the mouse, which is the OR of the following constants: **ShiftMask**, **LockMask**, **ControlMask**, **Mod1Mask**, **Mod2Mask**, **Mod3Mask**, **Mod4Mask**, **Mod5Mask**, **Button1Mask**, **Button2Mask**, **Button3Mask**, **Button4Mask**, **Button5Mask**, depending on which button or key is pressed.

Copyright © 1993 Thinking Machines Corporation

This routine returns immediately with the current state; it does not wait for any buttons to be pressed.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXDeallocateMousePoint

Deallocates a mouse point.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXDeallocateMousePoint(point)
   INTEGER point
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDeallocateMousePoint(CMXMousePoint_t point);
```

ARGUMENT

point	Specifies the point's *x* and *y* coordinates, and its button member.

DESCRIPTION

CMXDeallocateMousePoint deallocates a mouse point allocated with **CMXAllocateMousePoint**.

SEE ALSO

CMXAllocateMousePoint

Copyright © 1993 Thinking Machines Corporation

CMXDisplayAlloc

Allocates a CMX11 display structure.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXDisplayAlloc( )
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
CMX_display_t
CMXDisplayAlloc(void);
```

ARGUMENTS

None.

DESCRIPTION

CMXDisplayAlloc allocates a **cmx_display_t** data structure and returns an identifier for it.

This structure is used by the CMX11 library to identify an X11 server and to carry information about it. A set of routines allow you to set and retrieve fields in the **cmx_display_t** structure:

CMXSetDisplay	**CMXGetDisplay**
CMXSetDrawable	**CMXGetDrawable**
CMXSetGC	**CMXGetGC**
CMXSetVisualInfo	**CMXGetVisualInfo**
CMXSetColormap	**CMXGetColormap**
CMXSetWidget	**CMXGetWidget**

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXDisplayFree

Copyright © 1993 Thinking Machines Corporation

CMXDisplayFree

Frees a CMX11 display structure.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXDisplayFree(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDisplayFree(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t A CMX11 display identifier as allocated by **CMXDisplayAlloc**.

DESCRIPTION

CMXDisplayFree frees all resources associated with the *cmx_display_t* display structure.

This structure is created by **CMXDisplayAlloc** and by the CMX11 create routines **CMXCreateDisplayClass**, **CMXCreateDisplayClass**, and **CMXCreateDisplay**. It is used by the CMX11 library to identify an X11 server and to carry information about it. A set of routines allow you to set and retrieve fields in the **cmx_display_t** structure:

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXSetDisplay	**CMXGetDisplay**
CMXSetDrawable	**CMXGetDrawable**
CMXSetGC	**CMXGetGC**
CMXSetVisualInfo	**CMXGetVisualInfo**
CMXSetColormap	**CMXGetColormap**
CMXSetWidget	**CMXGetWidget**

NOTE: **CMXDisplayFree** does *not* free any colors allocated with **CMXAllocate-Colors**. You must free those by using **CMXFreeColors**.

SEE ALSO

CMXDisplayAlloc

Copyright © 1993 Thinking Machines Corporation

CMXDrawArc, CMXDrawCArc, CMXFillArc, CMXFillCArc

Draws and fills arcs, circles, and ellipses.

CM FORTRAN SYNTAX

```
 INCLUDE '/usr/include/cm/cmx-cmf.h'

 SUBROUTINE CMXDrawArc(dpy, drawable, gc, x, y, width, height,
&                      angle1, angle2)

 SUBROUTINE CMXDrawCArc(dpy, drawable, gc, x, y, width, height,
&                       angle1, angle2, color)

 SUBROUTINE CMXFillArc(dpy, drawable, gc, x, y, width, height,
&                      angle1, angle2)

 SUBROUTINE CMXFillCArc(dpy, drawable, gc, x, y, width, height,
&                       angle1, angle2, color)

   INTEGER dpy
   INTEGER drawable
   INTEGER gc
   INTEGER x(array_size), y(array_size)
   INTEGER width(array_size), height(array_size)
   INTEGER angle1(array_size), angle2(array_size)
   INTEGER color(array_size)
```

NOTE: Because the array arguments must be declared with an explicit rank, these routines cannot be defined in the CMX11 CM Fortran include files. If you want parameter type checking when the program is compiled, you must declare these routines with the desired dimensions in an **INTERFACE** block in your program. Sample **INTERFACE** blocks that may be copied into your program are available as comments in the include file **/usr/include/cm/cmx-cmf.h**.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawArc(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height,
    int:current angle1, int:current angle2);

#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawCArc(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height,
    int:current angle1, int:current angle2,
    int:current color);

#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFillArc(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height,
    int:current angle1, int:current angle2);

#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFillCArc(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height,
    int:current angle1, int:current angle2,
    int:current color);
```

ARGUMENTS

dpy — An integer identifying the X11 display in which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.

Copyright © 1993 Thinking Machines Corporation

drawable	An integer identifying the X11 drawable in which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.
gc	An integer identifying the X11 graphics context to be applied to drawing the arcs. The value for this argument is returned for a CMX11 display by **CMXGetGC**.
x, *y*	Parallel integer arrays containing the display coordinates for the upper left corner of each arc's bounding rectangle.
width, *height*	Parallel integer arrays containing the width and height of each arc's bounding rectangle. *width* is the distance, in pixels, to the right from the point (*x*, *y*). *height* is the distance, in pixels, downward.
angle1	A parallel array containing the starting point of each arc relative to the 3:00 position from the center of the bounding rectangle. The position is given in sixty-fourths of a degree. Positive values move counterclockwise; negative values move clockwise.
angle2	A parallel array containing the ending position of each arc relative to *angle1*.
color	For **CMXDrawCArc** and **CMXFillCArc**, a parallel array containing the color in which each arc is to be drawn.

DESCRIPTION

CMXDrawArc, **CMXDrawCArc**, **CMXFillArc**, and **CMXFillCArc** draw or fill a circular or elliptical arc for each data element in the *x*, *y*, *width*, *height*, *angle1*, and *angle2* parallel arrays. All these arrays must be conformable.

CMXDrawArc and **CMXFillArc** draw in the foreground color set in the display's graphics context. **CMXDrawCArc** and **CMXFillCArc** take an additional *color* parallel array argument with which you can specify an individual color for each arc.

The arguments *x*, *y*, *width*, and *height* define the bounding box of the arc, that is, the rectangle that contains the arc. The upper left corner of the rectangle is defined by (*x*, *y*), and the rectangle extends to the right *width* pixels and downward *height* pixels. The center of the arc is at the center of this rectangle, and the curve of the arc is drawn to intersect with the sides of the rectangle.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

The length of the arc is defined by *angle1* and *angle2*. *angle1* specifies the start of the arc relative to the 3:00 position from the center of the bounding rectangle. *angle2* specifies the end of the arc relative to *angle1*. The angles are given in sixty-fourths of a degree. Positive values move counterclockwise. 360 * 64 is a complete circle.

CMXFillArc and **CMXFillCArc** fill the area enclosed by the arc. The exact area is determined by the **arc_mode** set in the graphics context. If the **arc_mode** in the graphics context *gc* is **ArcChord**, the arc is filled to a line joining the endpoints of the arc. If **arc_mode** is **ArcPieSlice**, the boundary of the arc is defined by line segments from the endpoints of the arc to its center point. The **arc_mode** is set with the X Window System routine **XSetArcMode**.

CMXDrawArc and **CMXDrawCArc** draw arcs using the values for the following graphics context (*gc*) data elements: **function**, **plane_mask**, **line_width**, **line_style**, **cap_style**, **join_style**, **fill_style**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, **clip_mask**, **foreground**, **background**, **tile**, **stipple**, **ts_x_origin**, **ts_y_origin**, **dash_offset**, **dash_list**.

CMXFillArc and **CMXFillCArc** fill arcs using values for the following graphics context (*gc*) data elements: **function**, **plane_mask**, **fill_style**, **arc_mode**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, **clip_mask**, **foreground**, **background**, **tile**, **stipple**, **ts_x_origin**, **ts_y_origin**.

See your X Window System documentation for more information on the graphic context parameters.

Copyright © 1993 Thinking Machines Corporation

CMXDrawLine, CMXDrawCLine

Draws lines.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXDrawLine(dpy, drawable, gc, x1, y1, x2, y2)

SUBROUTINE CMXDrawCLine(dpy, drawable, gc, x1, y1, x2, y2, color)

   INTEGER dpy
   INTEGER drawable
   INTEGER gc
   INTEGER x1(array_size), y1(array_size)
   INTEGER x2(array_size), y2(array_size)
   INTEGER color(array_size)
```

NOTE: Because the array arguments must be declared with an explicit rank, these routines cannot be defined in the CMX11 CM Fortran include files. If you want parameter type checking when the program is compiled, you must declare these routines with the desired dimensions in an **INTERFACE** block in your program. Sample **INTERFACE** blocks that may be copied into your program are available as comments in the include file **/usr/include/cm/cmx-cmf.h**.

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawLine(Display *dpy, Drawable drawable,
    GC gc, int:current x1, int:current y1,
    int:current x2, int:current y2);
```


Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawCLine(Display *dpy, Drawable drawable,
    GC gc, int:current x1, int:current y1,
    int:current x2, int:current y2,
    int:current color);
```

ARGUMENTS

dpy	An integer identifying the X11 display to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.
drawable	An integer identifying the window on *dpy* in which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.
gc	An integer identifying the X11 graphics context to be applied to drawing the lines. The value for this argument is returned for a CMX11 display by **CMXGetGC**.
x1, *y1*	Parallel arrays containing the start coordinates of the lines to be drawn.
x2, *y2*	Parallel arrays containing the end coordinates of the lines to be drawn.
color	For **CMXDrawCLine**, a parallel array containing a color value for each line to be drawn.

Copyright © 1993 Thinking Machines Corporation

DESCRIPTION

CMXDrawLine and **CMXDrawCLine** draw a line for each data element in the coordinate arrays *x1*, *y1*, *x2*, and *y2*. All these arrays must be conformable. The corresponding elements of the *x1* and *y1* arrays supply the beginning points of the lines, and the *x2* and *y2* arrays supply the endpoints. So, the first line is drawn from (*x1*(1), *y1*(1)) to ((*x2*(1), *y2*(1)), the second from (*x1*(2), *y1*(2)) to ((*x2*(2), *y2*(2)), etc. Unlike **XDrawLines**, the lines defined in **CMXDrawLine** and **CMXDrawCLine** need not be consecutive, that is, they need not be joined. No pixel is drawn more than once.

CMXDrawLine draws in the window foreground color. **CMXDrawCLine** allows each line to have its own color; each element of the *color* array provides the color for the line defined by the corresponding data elements in the coordinate arrays.

Lines are drawn using the values for the following graphics context (*gc*) data elements: **function**, **plane_mask**, **line_width**, **line_style**, **cap_style**, **fill_style**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, **clip_mask**, **foreground**, **background**, **tile**, **stipple**, **ts_x_origin**, **ts_y_origin**, **dash_offset**, **dash_list**.

See your X Window System documentation for more information on these graphics context parameters.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXDrawPoint, CMXDrawCPoint

Draws points.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXDrawPoint(dpy, drawable, gc, x, y)

SUBROUTINE CMXDrawCPoint(dpy, drawable, gc, x, y, color)

   INTEGER dpy
   INTEGER drawable
   INTEGER gc
   INTEGER x(array_size), y(array_size)
   INTEGER color(array_size)
```

NOTE: Because these array arguments must be declared with an explicit rank, these routines cannot be defined in the CMX11 CM Fortran include files. If you want parameter type checking when the program is compiled, you must declare these routines with the desired dimensions in an **INTERFACE** block in your program. Sample **INTERFACE** blocks that may be copied into your program are available as comments in the include file **/usr/include/cm/cmx-cmf.h**.

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawPoint(Display *dpy, Drawable drawable, GC gc,
             int:current x, int:current y);
```


Copyright © 1993 Thinking Machines Corporation

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawCPoint(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current color);
```

ARGUMENTS

dpy	An integer identifying the X11 display to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.
drawable	An integer identifying the window on *dpy* to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.
gc	An integer identifying the X11 graphics context to be applied when drawing the points. The value for this argument is returned for a CMX11 display by **CMXGetGC**.
x, *y*	Parallel arrays containing the coordinates of the points to be drawn.
color	A parallel array containing a color value for each point to be drawn.

DESCRIPTION

CMXDrawPoint and **CMXDrawCPoint** draw a point for each data element in the coordinate arrays *x* and *y*. The arrays must be conformable, that is they must be the same size and shape. Each point corresponds to a pixel in the display drawable. The first point is drawn at $(x(1), y(1))$, the second at $(x(2), y(2))$, etc.

CMXDrawPoint draws the points in the foreground color specified in the display's graphics context. For **CMXDrawCPoint**, the *color* array is an array of color values conforming to the *x* and *y* coordinate arrays. Each point is drawn in the color specified in the corresponding element of the *color* array.

Copyright © 1993 Thinking Machines Corporation

Points are drawn using the graphics context (*gc*) values for **function**, **plane_mask**, **foreground**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, and **clip_mask**. See your X Window System documentation for more information on the graphics context parameters.

Copyright © 1993 Thinking Machines Corporation

CMXDrawRectangle, CMXDrawCRectangle, CMXFillRectangle, CMXFillCRectangle

Draws and fills rectangles.

CM FORTRAN SYNTAX

```
 INCLUDE '/usr/include/cm/cmx-cmf.h'

 SUBROUTINE CMXDrawRectangle(dpy, drawable, gc, x, y, width, height)

 SUBROUTINE CMXDrawCRectangle(dpy, drawable, gc, x, y, width, height,
&                             color)

 SUBROUTINE CMXFillRectangle(dpy, drawable, gc, x, y, width, height)

 SUBROUTINE CMXFillCRectangle(dpy, drawable, gc, x, y, width, height,
&                             color)

   INTEGER dpy
   INTEGER drawable
   INTEGER gc
   INTEGER x(array_size), y(array_size)
   INTEGER width(array_size), height(array_size)
   INTEGER color(array_size)
```

NOTE: Because the array arguments must be declared with an explicit rank, these routines cannot be defined in the CMX11 CM Fortran include files. If you want parameter type checking when the program is compiled, you must declare these routines with the desired dimensions in an **INTERFACE** block in your program. Sample **INTERFACE** blocks that may be copied into your program are available as comments in the include file **/usr/include/cm/cmx-cmf.h**.

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawRectangle(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height);
#include <cm/cmxdisplay.h>
```


Copyright © 1993 Thinking Machines Corporation

```
#include <cm/cmx-cs.h>
void
CMXDrawCRectangle(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height,
    int:current color);

#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFillRectangle(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height);

#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFillCRectangle(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    int:current width, int:current height,
    int:current color);
```

ARGUMENTS

dpy	An integer identifying the X11 display to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.
drawable	An integer identifying the window on *dpy* to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.
gc	An integer identifying the X11 graphics context to be applied to drawing the lines. The value for this argument is returned for a CMX11 display by **CMXGetGC**.
x, *y*	Parallel arrays containing the coordinates for the location of the upper left corner of each rectangle to be drawn.
width, *height*	Parallel arrays containing the dimensions, in pixels, of the rectangles to be drawn.

Copyright © 1993 Thinking Machines Corporation

color For **CMXDrawCRectangle** and **CMXFillCRectangle**, a parallel array containing a color for each rectangle to be drawn.

DESCRIPTION

CMXDrawRectangle, **CMXFillRectangle**, **CMXDrawCRectangle**, and **CMXFillCRectangle** draw or draw and fill a rectangle for each data element in the *x*, *y*, *width*, and *height* arrays. All these arrays must be conformable.

The *x* and *y* arrays define the upper left corner of each rectangle to be drawn in relation to the origin of the window. The corresponding elements of the *width* and *height* arrays specify the size of the rectangle from the point (x, y). *width* is the number of pixels the rectangle extends to the right, and *height* is the number of pixels downward.

CMXDrawRectangle draws the outline of a rectangle for each data element in the parallel arrays in the foreground color. **CMXFillRectangle** draws a filled rectangle. Note that **CMXFillRectangle** does not light pixels along the right and bottom edge of the area lit by **CMXDrawRectangle**. A drawn rectangle outlines a rectangle from (x, y) at the upper left corner to $(x + width, y + height)$ at the lower right corner. A filled rectangle creates a solid rectangle from (x, y) to $(x + (width - 1), y + (height - 1))$.

CMXDrawCRectangle and **CMXFillCRectangle** allow you to specify a color for each rectangle. Each element of the *color* array supplies the color for the rectangle defined by the corresponding elements of the other arrays.

Rectangles are drawn and filled using the values for the following graphics context (*gc*) data elements: **function**, **plane_mask**, **line_width**, **line_style**, **cap_style**, **join_style**, **fill_style**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, **clip_mask**, **foreground**, **background**, **tile**, **stipple**, **ts_x_origin**, **ts_y_origin**, **dash_offset**, and **dash_list**.

See your X Window System documentation for more information on the graphics context parameters.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXDrawString, CMXDrawCString

Draws text strings.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXDrawString(dpy, drawable, gc, x, y, string, slen)

SUBROUTINE CMXDrawCString(dpy, drawable, gc, x, y, string, slen, color)

  INTEGER dpy
  INTEGER drawable
  INTEGER gc
  INTEGER x(array_size), y(array_size)
  INTEGER string(array_size)
  INTEGER slen
  INTEGER color(array_size)
```

NOTE: Because the array arguments must be declared with an explicit rank, these routines cannot be defined in the CMX11 CM Fortran include files. If you want parameter type checking when the program is compiled, you must declare these routines with the desired dimensions in an **INTERFACE** block in your program. Sample **INTERFACE** blocks that may be copied into your program are available as comments in the include file **/usr/include/cm/cmx-cmf.h**.

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawString(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    char:current *string, int slen);
```


Copyright © 1993 Thinking Machines Corporation

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXDrawCString(Display *dpy, Drawable drawable,
    GC gc, int:current x, int:current y,
    char:current *string, int slen,
    int:current color);
```

ARGUMENTS

dpy — An integer identifying the X11 display to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.

drawable — An integer identifying the window on *dpy* to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.

gc — An integer identifying the X11 graphics context to be applied to drawing the text strings. The value for this argument is returned for a CMX11 display by **CMXGetGC**.

x, *y* — Parallel arrays containing the coordinates at which the text string is to be drawn.

string — A parallel array containing the character strings to be drawn. The programmer is responsible for packing the characters into the *string* array. For example, the following lines of code pack the character string “abcd” into all the elements of the *string* array:

```
    INTEGER string(1,100)
 cmf$layout string(:serial,:news)
    string(1,:) = ICHAR(”a”) * 65536 * 256
   &            + ICHAR(”b”) * 65536
   &            + ICHAR(”c”) * 256
   &            + ICHAR(”d”)
```

slen — The length, in characters, of each string to be drawn. All strings must be the same length.

color — For **CMXDrawCString**, a parallel array containing a color for each string to be drawn.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

DESCRIPTION

CMXDrawString writes a string to the specified drawable in the foreground color. **CMXDrawCString** draws each string in the color defined by the corresponding element of the *color* array.

The coordinates (*x*, *y)* define the origin of the text string; *y* is the baseline row of pixels and *x* is the point from which the text string's lbearing, rbearing, and width are measured.

CMXDrawString draws the given string into the drawable using the **foreground** color set in the graphics context (*gc*) to draw only the bits set in the font defined in *gc*. **CMXDrawCString** draws each string in the color defined by the corresponding element of the *color* array. The values of the following graphics context data elements are also used by **CMXDrawString** and **CMXDrawCString**: **function**, **plane_mask**, **fill_style**, **font**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, **clip_mask**, **foreground**, **tile**, **stipple**, **ts_x_origin**, and **ts_y_origin**.

Copyright © 1993 Thinking Machines Corporation

CMXFillPolygon, CMXFillCPolygon

Draws and fills polygons.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXFillPolygon(dpy, drawable, gc, x, y, shape, mode)

SUBROUTINE CMXFillCPolygon(dpy, drawable, gc, x, y, shape, mode, color)

      INTEGER dpy
      INTEGER drawable
      INTEGER gc
      INTEGER x(num_points, num_polys)
CMF$  LAYOUT x(:SERIAL, )
      INTEGER y(num_points, num_polys)
CMF$  LAYOUT x(:SERIAL, )
      INTEGER shape
      INTEGER mode
      INTEGER color(num_polys)
```

NOTE: Because the array arguments must be declared with an explicit rank, these routines cannot be defined in the CMX11 CM Fortran include files. If you want parameter type checking when the program is compiled, you must declare these routines with the desired dimensions in an **INTERFACE** block in your program. Sample **INTERFACE** blocks that may be copied into your program are available as comments in the include file **/usr/include/cm/cmx-cmf.h**.

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFillPolygon(Display *dpy, Drawable drawable,
    GC gc, Xpoint:current points[], int npoints,
    int _shape, int mode);
```


Copyright © 1993 Thinking Machines Corporation

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFillCPolygon(Display *dpy, Drawable drawable,
    GC gc, Xpoint:current points[], int npoints,
    int _shape, int mode, int:current color);
```

ARGUMENTS

dpy — An integer identifying the X11 display to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.

drawable — An integer identifying the window on *dpy* to which you wish to draw. The value for this argument is returned for a CMX11 display by **CMXGetDrawable**.

gc — An integer identifying the X11 graphics context to be applied to drawing the polygons. The value for this argument is returned for a CMX11 display by **CMXGetGC**.

x, *y* or *points[]*, *npoints* — 2D parallel arrays containing the coordinates of the polygons to be drawn. In C*, an Xpoint is defined as:

```
typedef struct Xpoint {
                short x,y;
        }
```

shape — The shape class of the polygons. Valid values are **Complex**, **Nonconvex**, and **Convex**. See Description below for more information.

mode — The coordinate mode of the polygons. Valid values are **CoordModeOrigin** and **CoordModePrevious**. See Description below for more information.

color — A parallel array containing a color value for each polygon to be drawn.

Copyright © 1993 Thinking Machines Corporation

DESCRIPTION

CMXFillPolygon and **CMXFillCPolygon** draw and fill a polygon for each data element in the coordinate arrays *x* and *y* or *points[]* and *npoints*. All polygons must contain the same number of points.

The arguments *x*, *y*, and *color* must be conformable arrays. The corresponding elements of *x* and *y* define the coordinates of the points of the polygons. So, for example, if each polygon is defined by three points, the first polygon is drawn between $(x(1,1), y(1,1))$, $(x(1,2), y(1,2))$, and $(x(1,3), y(1,3))$. And, for **CMXFillCPolygon**, the *color* array element (1) supplies the color value. **CMXFillPolygon** draws in the current foreground color for the window.

The *shape* and *mode* arguments are the same as in the Xlib routine **XFillPolygon**. The *shape* argument classes the shape of the polygons, allowing the system to use faster algorithms to draw and fill polygons with a simpler shape. Valid values for *shape* are:

Complex The path may self-intersect. The **fill_rule** (**EvenOddRule** or **WindingRule**) defined in the graphics context *gc* determines how the shape is filled. See your X Window System documentation for more information.

Nonconvex The path does not self-intersect, but the shape is not wholly convex. Specifying **Nonconvex** instead of **Complex** may improve performance. However, if the path is self-intersecting, the graphics results are undefined.

Convex The path is wholly convex. Specifying **Convex** may improve performance. However, if the path is not convex, the graphics results are undefined.

The *mode* argument specifies a coordinate mode, that is, how the sequence of coordinates is to be interpreted. Valid values for *mode* are

CoordModeOrigin
All coordinate values are interpreted in relation to the origin of the window. That is, the coordinates define positions as the number of pixels from the upper left corner of the window.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CoordModePrevious

The first point is defined in relation to the origin of the window, but each succeeding point is defined in relation to the one that precedes it. That is, the point coordinates are offsets from the position of the previous point. For example, if the first point is located at the window coordinates (20, 30) and the second point coordinates are (10, 15), the second point is drawn at (30, 45).

Polygons are drawn and filled using the values for the following graphics context (*gc*) data elements for **function**, **plane_mask**, **fill_style**, **fill_rule**, **subwindow_mode**, **clip_x_origin**, **clip_y_origin**, **clip_mask**, **foreground**, **background**, **tile**, **stipple**, **ts_x_origin**, and **ts_y_origin**.

See your X Window System documentation for more information on the graphics context parameters.

Copyright © 1993 Thinking Machines Corporation

CMXFreeColors

Frees colors allocated by **CMXAllocateColors**.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXFreeColors(display, cmap, ncolors, base)
   INTEGER display
   INTEGER cmap
   INTEGER ncolors
   INTEGER base
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFreeColors(Display *display, Colormap cmap, int ncolors, int base);
```

ARGUMENTS

display	An integer identifying the X11 display for which you wish to deallocate colors. The value for this argument is returned for a CMX11 display by **CMXGetDisplay**.
cmap	An integer identifying the X11 **Colormap** on *display* for which you wish to deallocate colors. The value for this argument is returned for a CMX11 display by **CMXGetColormap**.
ncolors	The number of colors you wish to deallocate.
base	The entry number of the first color cell to deallocate.

DESCRIPTION

CMXFreeColors frees colors allocated by **CMXAllocateColors**. It deallocates *ncolors* consecutive colors in the display colormap, beginning with *base*.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXAllocateColors

Copyright © 1993 Thinking Machines Corporation

CMXFreeMBuffers

Deallocates display multibuffers.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXFreeMBuffers(widget, buffers)
   INTEGER widget
   INTEGER buffers
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXFreeMBuffers(Widget widget, Multibuffer *buffers);
```

ARGUMENTS

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a display.
buffers	An integer identifying the multibuffer pair to be deallocated. This value is returned by **CMXAllocMBuffers**.

DESCRIPTION

CMXFreeMBuffers frees all resources associated with the multibuffers, making them available to other applications.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXAllocMBuffers
CMXGetDrawMBuffer
CMXGetDisplayMBuffer
CMXSwitchMBuffers

Copyright © 1993 Thinking Machines Corporation

CMXGetColormap

Returns X11 **Colormap** identifier for a CMX display.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetColormap(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Colormap
CMXGetColormap(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t	A CMX11 display identifier as returned by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**.

DESCRIPTION

CMXGetColormap returns the X11 **Colormap** identifier for the X Window System display to which *cmx_display_t* is connected.

The colormap identifier is passed to the color management routines **CMXAllocateColors**, **CMXStoreColors**, and **CMXStoreColor**.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXGetDisplay
CMXGetDrawable
CMXGetGC
CMXGetWidget

Copyright © 1993 Thinking Machines Corporation

CMXGetDepth

Returns the depth (bits-per-pixel) of a CMX11 display window.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetDepth(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXGetDepth(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXGetDepth returns the depth of the display window identified by *widget*. The depth of a display window is the number of bits of color information maintained per pixel.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXHasColormap
CMXIsColormap
CMXIsRGB
CMXColormapSize
CMXColormapRedSize
CMXColormapGreenSize
CMXColormapBlueSize
CMXBitsOfRed
CMXBitsOfGreen
CMXBitsOfBlue

Copyright © 1993 Thinking Machines Corporation

CMXGetDisplay

Gets X11 display identifier for a CMX11 display.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetDisplay(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Display*
CMXGetDisplay(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t	A CMX11 display identifier as returned by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**.

DESCRIPTION

CMXGetDisplay returns the X11 **Display** identifier for the X Window System display to which *cmx_display_t* is connected.

This returned value is passed to all CMX11 drawing routines; to the color management routines **CMXAllocateColors**, **CMXStoreColors**, and **CMXStoreColor**; and to the double buffering routine **CMXHasMBuffers**.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXGetDrawable
CMXGetGC
CMXGetColormap
CMXGetWidget

Copyright © 1993 Thinking Machines Corporation

CMXGetDisplayMBuffer

Returns identifier for buffer being displayed.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetDisplayMBuffer(buffers)
   INTEGER buffers
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Multibuffer
CMXGetDisplayMBuffer(Multibuffer *buffers);
```

ARGUMENT

buffers	An integer identifying the multibuffer pair to be used for double buffering. This value is returned by **CMXAllocMBuffers**.

DESCRIPTION

CMXGetDisplayMBuffer returns an identifier for the multibuffer that is currently being displayed. This return value can be passed to any of the drawing routines as the *drawable* argument. Changes made to the display buffer immediately appear on the screen.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXAllocMBuffers
CMXGetDrawMBuffer
CMXSwitchMBuffers

Copyright © 1993 Thinking Machines Corporation

CMXGetDrawMBuffer

Returns identifier for buffer being drawn to.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetDrawMBuffer(buffers)
   INTEGER buffers
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Multibuffer
CMXGetDrawMBuffer(Multibuffer *buffers);
```

ARGUMENT

buffers	An integer identifying the multibuffer pair to be used for double buffering. This value is returned by **CMXAllocMBuffers**.

DESCRIPTION

CMXGetDrawMBuffer returns an identifier for the multibuffer that is currently being drawn to. This return value can be passed to any of the drawing routines as the *drawable* argument. Changes made to the draw buffer will not be displayed until **CMXSwitchMBuffers** is called.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXAllocMBuffers
CMXGetDisplayMBuffer
CMXSwitchMBuffers

Copyright © 1993 Thinking Machines Corporation

CMXGetDrawable

Gets X11 **Drawable** identifier for a CMX11 display.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetDrawable(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Drawable
CMXGetDrawable(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t	A CMX11 display identifier as returned by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**.

DESCRIPTION

CMXGetDrawable returns the X11 **Drawable** identifier for the X Window System display to which *cmx_display_t* is connected.

The **Drawable** identifier is passed to all the CMX11 drawing routines.

SEE ALSO

CMXGetDisplay
CMXGetGC
CMXGetColormap
CMXGetWidget

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXGetGC

Gets X11 graphics context identifier for a CMX11 display.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetGC(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
GC
CMXGetGC(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t	A CMX11 display identifier as returned by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**.

DESCRIPTION

CMXGetGC returns the X11 graphics context (**GC**) identifier for the X Window System display to which *cmx_display_t* is connected.

The graphics context identifier is passed to all the CMX11 drawing routines, and to the color convenience routines **CMXSetForeground** and **CMXSetBackground**.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXGetDisplay
CMXGetDrawable
CMXGetColormap
CMXGetWidget

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXGetHeight

Returns the height, in pixels, of a CMX11 display window.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetHeight(widget)
   INTEGER widget
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
int
CMXGetHeight(Widget widget);
```

ARGUMENT

widget	A CMX11 widget identifier as returned by **CMXGetWidget**. The widget identifies the drawing surface of a CMX11 display.

DESCRIPTION

CMXGetHeight returns the height, in pixels, of the display window identified by *widget*. The height is the vertical dimension of the window inside any window border that may be used.

SEE ALSO

CMXSetHeight
CMXGetWidth
CMXSetWidth

Copyright © 1993 Thinking Machines Corporation

CMXGetMouseLine

Draws a straight “rubber band” line.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXGetMouseLine(widget, p1, p2, anchorp)
   INTEGER widget
   INTEGER p1, p2
   INTEGER anchorp
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXGetMouseLine(Widget widget, CMXMousePoint_t p1,
                  CMXMousePoint_t p2, int anchorp);
```

ARGUMENTS

widget	Specifies a widget.
p1	Specifies the first point’s *x* and *y* coordinates, and its button member.
p2	Specifies the second point’s *x* and *y* coordinates, and its button member.
anchorp	An integer specifying whether *p1* is to be used as an input parameter.

DESCRIPTION

CMXGetMouseLine draws a mouse-controlled “rubber band” line. First it waits for a mouse button to be pressed; then it draws a moving line from that point to the current point as the mouse is dragged, until the mouse button is released. **CMXGetMouseLine** returns the start coordinate in *p1* and the end coordinate in *p2*.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

If *anchorp* is true (nonzero), the first point is "anchored" to the contents of *p1*. In other words, *p1* is used as an input *and* output parameter, and the point where the button is pressed is insignificant; only the release point is used.

The button members of *p1* and *p2* are filled in with **Button1**, **Button2**, or **Button3**, depending on which button is pressed and released.

Copyright © 1993 Thinking Machines Corporation

CMXGetMousePoint

Returns the coordinates of a button release.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXGetMousePoint(widget, point)
   INTEGER widget
   INTEGER point
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXGetMousePoint(Widget widget, CMXMousePoint_t point);
```

ARGUMENTS

widget	Specifies a widget.
point	Specifies the point's *x* and *y* coordinates, and its button member.

DESCRIPTION

`CMXGetMousePoint` waits for a button to be pressed and released, and then returns (in *point*) the coordinates of the release. The "button" member of the point is filled in with one of the constants **`Button1`**, **`Button2`**, or **`Button3`**, depending on which button is pressed.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXGetMouseRectangle

Draws a “rubber band” rectangle.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

SUBROUTINE CMXGetMouseRectangle(widget, p1, p2, anchorp)
   INTEGER widget
   INTEGER p1, p2
   INTEGER anchorp
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
void
CMXGetMouseRectangle(Widget widget, CMXMousePoint_t p1,
                     CMXMousePoint_t p2, int anchorp);
```

ARGUMENTS

widget	Specifies a widget.
p1	Specifies the first point’s *x* and *y* coordinates, and its button member.
p2	Specifies the second point’s *x* and *y* coordinates, and its button member.
anchorp	An integer specifying whether *p1* is to be used as an input parameter.

Copyright © 1993 Thinking Machines Corporation

DESCRIPTION

CMXGetMouseRectangle draws a mouse-controlled "rubber band" rectangle. First it waits for a mouse button to be pressed; then it draws a moving rectangle from that point to the current point as the mouse is dragged, until the mouse button is released. **CMXGetMouseRectangle** returns the start coordinate in *p1* and the end coordinate in *p2*.

If *anchorp* is true (nonzero), the first point is "anchored" to the contents of *p1*. In other words, *p1* is used as an input *and* output parameter, and the point where the button is pressed is insignificant; only the release point is used.

The button members of *p1* and *p2* are filled in with **Button1**, **Button2**, or **Button3**, depending on which button is pressed and released.

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXGetVisualInfo

Gets X11 **XVisualInfo** identifier for a CMX11 display.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetVisualInfo(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
XVisualInfo*
CMXGetVisualInfo(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t	A CMX11 display identifier as returned by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**.

DESCRIPTION

The visual information structure is created for a CMX11 display when the display is created by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**. You will not need to access this structure for most CMX11 applications.

For programs that are integrating CMX11 with an X11 application, **CMXGetVisualInfo** returns an identifier for the **XVisualInfo** structure of the X11 window to which *cmx_display_t* is connected. This identifier may be passed to CMX11 or X11 routines that accept a pointer to an **XVisualInfo** structure.

The CMX11 routines that take a visual information identifier as an argument are **CMXCreateDisplay** and **CMXSetVisualInfo**.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXGetDisplay
CMXGetDrawable
CMXGetColormap
CMXGetGC
CMXGetWidget
CMXGetXAppClass
CMXGetXAppName
CMXGetXResourceInteger
CMXGetXResourceString

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation

CMXGetWidget

Gets X11 widget identifier for a CMX display.

CM FORTRAN SYNTAX

```
INCLUDE '/usr/include/cm/cmx-cmf.h'

INTEGER FUNCTION CMXGetWidget(cmx_display_t)
   INTEGER cmx_display_t
```

C* SYNTAX

```
#include <cm/cmxdisplay.h>
#include <cm/cmx-cs.h>
Widget
CMXGetWidget(CMX_display_t cmx_display_t);
```

ARGUMENT

cmx_display_t	A CMX11 display identifier as returned by **CMXCreateSimpleDisplay** or **CMXCreateDisplayClass**.

DESCRIPTION

CMXGetWidget returns the X11 **Widget** identifier for the X Window System display to which *cmx_display_t* is connected.

The widget identifier is passed to all the information routines that return information about the window and to the double buffering routines **CMXAllocMBuffers**, **CMXSwitchMBuffers**, and **CMXFreeMBuffers**.

Copyright © 1993 Thinking Machines Corporation

SEE ALSO

CMXGetDisplay
CMXGetDrawable
CMXGetGC
CMXGetColormap

Version 1.6, October 1993
Copyright © 1993 Thinking Machines Corporation