```
/*

Copyright 1985, 1986, 1987, 1991, 1998  The Open Group

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation.

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
THE
OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be
used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization from The Open Group.

*/


/*
 *	Xlib.h - Header definition and support file for the C subroutine
 *	interface library (Xlib) to the X Window System Protocol (V11).
 *	Structures and symbols starting with "_" are private to the library.
 */
#ifndef _X11_XLIB_H_
#define _X11_XLIB_H_

#define XlibSpecificationRelease 6

#include <sys/types.h>

#if defined(__SCO__) || defined(__UNIXWARE__)
#include <stdint.h>
#endif

#include <X11/X.h>

/* applications should not depend on these two headers being included! */
```

```
#include <X11/Xfuncproto.h>
#include <X11/Xosdefs.h>

#ifndef X_WCHAR
#include <stddef.h>
#else
#ifdef __UNIXOS2__
#include <stdlib.h>
#else
/* replace this with #include or typedef appropriate for your system */
typedef unsigned long wchar_t;
#endif
#endif


extern int
_Xmblen(
    char *str,
    int len
    );

/* API mentioning "UTF8" or "utf8" is an XFree86 extension, introduced in
   November 2000. Its presence is indicated through the following macro. */
#define X_HAVE_UTF8_STRING 1

/* The Xlib structs are full of implicit padding to properly align members.
   We can't clean that up without breaking ABI, so tell clang not to bother
   complaining about it. */
#ifdef __clang__
#pragma clang diagnostic push
#pragma clang diagnostic ignored "-Wpadded"
#endif

typedef char *XPointer;

#define Bool int
#define Status int
#define True 1
#define False 0

#define QueuedAlready 0
#define QueuedAfterReading 1
#define QueuedAfterFlush 2

#define ConnectionNumber(dpy)      (((_XPrivDisplay)dpy)->fd)
#define RootWindow(dpy, scr)       (ScreenOfDisplay(dpy,scr)->root)
#define DefaultScreen(dpy)   (((_XPrivDisplay)dpy)->default_screen)
#define DefaultRootWindow(dpy)     (ScreenOfDisplay(dpy,DefaultScreen(dpy))->root)
#define DefaultVisual(dpy, scr) (ScreenOfDisplay(dpy,scr)->root_visual)
```

```
#define DefaultGC(dpy, scr)  (ScreenOfDisplay(dpy,scr)->default_gc)
#define BlackPixel(dpy, scr)  (ScreenOfDisplay(dpy,scr)->black_pixel)
#define WhitePixel(dpy, scr)  (ScreenOfDisplay(dpy,scr)->white_pixel)
#define AllPlanes             ((unsigned long)~0L)
#define QLength(dpy)                (((_XPrivDisplay)dpy)->qlen)
#define DisplayWidth(dpy, scr)      (ScreenOfDisplay(dpy,scr)->width)
#define DisplayHeight(dpy, scr) (ScreenOfDisplay(dpy,scr)->height)
#define DisplayWidthMM(dpy, scr)(ScreenOfDisplay(dpy,scr)->mwidth)
#define DisplayHeightMM(dpy, scr)(ScreenOfDisplay(dpy,scr)->mheight)
#define DisplayPlanes(dpy, scr) (ScreenOfDisplay(dpy,scr)->root_depth)
#define DisplayCells(dpy, scr)      (DefaultVisual(dpy,scr)->map_entries)
#define ScreenCount(dpy)      (((_XPrivDisplay)dpy)->nscreens)
#define ServerVendor(dpy)     (((_XPrivDisplay)dpy)->vendor)
#define ProtocolVersion(dpy)        (((_XPrivDisplay)dpy)->proto_major_version)
#define ProtocolRevision(dpy)       (((_XPrivDisplay)dpy)->proto_minor_version)
#define VendorRelease(dpy)    (((_XPrivDisplay)dpy)->release)
#define DisplayString(dpy)    (((_XPrivDisplay)dpy)->display_name)
#define DefaultDepth(dpy, scr)      (ScreenOfDisplay(dpy,scr)->root_depth)
#define DefaultColormap(dpy, scr)(ScreenOfDisplay(dpy,scr)->cmap)
#define BitmapUnit(dpy)       (((_XPrivDisplay)dpy)->bitmap_unit)
#define BitmapBitOrder(dpy)         (((_XPrivDisplay)dpy)->bitmap_bit_order)
#define BitmapPad(dpy)              (((_XPrivDisplay)dpy)->bitmap_pad)
#define ImageByteOrder(dpy)         (((_XPrivDisplay)dpy)->byte_order)
#define NextRequest(dpy)      (((_XPrivDisplay)dpy)->request + 1)
#define LastKnownRequestProcessed(dpy)  (((_XPrivDisplay)dpy)->last_request_read)

/* macros for screen oriented applications (toolkit) */
#define ScreenOfDisplay(dpy, scr)(&((_XPrivDisplay)dpy)->screens[scr])
#define DefaultScreenOfDisplay(dpy) ScreenOfDisplay(dpy,DefaultScreen(dpy))
#define DisplayOfScreen(s)    ((s)->display)
#define RootWindowOfScreen(s)       ((s)->root)
#define BlackPixelOfScreen(s)       ((s)->black_pixel)
#define WhitePixelOfScreen(s)       ((s)->white_pixel)
#define DefaultColormapOfScreen(s)((s)->cmap)
#define DefaultDepthOfScreen(s)     ((s)->root_depth)
#define DefaultGCOfScreen(s)        ((s)->default_gc)
#define DefaultVisualOfScreen(s)((s)->root_visual)
#define WidthOfScreen(s)      ((s)->width)
#define HeightOfScreen(s)     ((s)->height)
#define WidthMMOfScreen(s)          ((s)->mwidth)
#define HeightMMOfScreen(s)         ((s)->mheight)
#define PlanesOfScreen(s)     ((s)->root_depth)
#define CellsOfScreen(s)      (DefaultVisualOfScreen((s))->map_entries)
#define MinCmapsOfScreen(s)         ((s)->min_maps)
#define MaxCmapsOfScreen(s)         ((s)->max_maps)
#define DoesSaveUnders(s)     ((s)->save_unders)
#define DoesBackingStore(s)((s)->backing_store)
#define EventMaskOfScreen(s)        ((s)->root_input_mask)
```

```
/*
 * Extensions need a way to hang private data on some structures.
 */
typedef struct _XExtData {
        int number;             /* number returned by XRegisterExtension */
        struct _XExtData *next;         /* next item on list of data for structure */
        int (*free_private)(    /* called to free private storage */
        struct _XExtData *extension
        );
        XPointer private_data;          /* data private to this extension. */
} XExtData;

/*
 * This file contains structures used by the extension mechanism.
 */
typedef struct {                /* public to extension, cannot be changed */
        int extension;          /* extension number */
        int major_opcode;       /* major op-code assigned by server */
        int first_event;/* first event number for the extension */
        int first_error; /* first error number for the extension */
} XExtCodes;

/*
 * Data structure for retrieving info about pixmap formats.
 */

typedef struct {
   int depth;
   int bits_per_pixel;
   int scanline_pad;
} XPixmapFormatValues;


/*
 * Data structure for setting graphics context.
 */
typedef struct {
        int function;           /* logical operation */
        unsigned long plane_mask;/* plane mask */
        unsigned long foreground;/* foreground pixel */
        unsigned long background;/* background pixel */
        int line_width;         /* line width */
        int line_style;         /* LineSolid, LineOnOffDash, LineDoubleDash */
        int cap_style;          /* CapNotLast, CapButt,
                                   CapRound, CapProjecting */
        int join_style;         /* JoinMiter, JoinRound, JoinBevel */
        int fill_style;         /* FillSolid, FillTiled,
                                   FillStippled, FillOpaeueStippled */
        int fill_rule;          /* EvenOddRule, WindingRule */
	int arc_mode;		/* ArcChord, ArcPieSlice */
	Pixmap tile;		/* tile pixmap for tiling operations */
	Pixmap stipple;			/* stipple 1 plane pixmap for stipping */
	int ts_x_origin;	/* offset for tile or stipple operations */
	int ts_y_origin;
      Font font;	        /* default text font for text operations */
	int subwindow_mode;     /* ClipByChildren, IncludeInferiors */
	Bool graphics_exposures;/* boolean, should exposures be generated */
	int clip_x_origin;	/* origin for clipping */
	int clip_y_origin;
	Pixmap clip_mask;	/* bitmap clipping; other calls for rects */
	int dash_offset;	/* patterned/dashed line information */
	char dashes;
} XGCValues;

/*
 * Graphics context.  The contents of this structure are implementation
 * dependent.  A GC should be treated as opaque by application code.
 */

typedef struct _XGC
#ifdef XLIB_ILLEGAL_ACCESS
{
    XExtData *ext_data;		/* hook for extension to hang data */
    GContext gid;	/* protocol ID for graphics context */
    /* there is more to this structure, but it is private to Xlib */
}
#endif
*GC;

/*
 * Visual structure; contains information about colormapping possible.
 */
typedef struct {
	XExtData *ext_data;	/* hook for extension to hang data */
	VisualID visualid;	/* visual id of this visual */
#if defined(__cplusplus) || defined(c_plusplus)
	int c_class;		/* C++ class of screen (monochrome, etc.) */
#else
	int class;		/* class of screen (monochrome, etc.) */
#endif
	unsigned long red_mask, green_mask, blue_mask;	/* mask values */
	int bits_per_rgb;	/* log base 2 of distinct color values */
	int map_entries;	/* color map entries */
} Visual;

/*
 * Depth structure; contains information for each possible depth.
 */
```

```
typedef struct {
        int depth;              /* this depth (Z) of the depth */
        int nvisuals;           /* number of Visual types at this depth */
        Visual *visuals;        /* list of visuals possible at this depth */
} Depth;

/*
 * Information about the screen.  The contents of this structure are
 * implementation dependent.  A Screen should be treated as opaque
 * by application code.
 */

struct _XDisplay;               /* Forward declare before use for C++ */

typedef struct {
        XExtData *ext_data;     /* hook for extension to hang data */
        struct _XDisplay *display;/* back pointer to display structure */
        Window root;            /* Root window id. */
        int width, height;      /* width and height of screen */
        int mwidth, mheight;    /* width and height of  in millimeters */
        int ndepths;            /* number of depths possible */
        Depth *depths;                  /* list of allowable depths on the screen */
        int root_depth;                 /* bits per pixel */
        Visual *root_visual;    /* root visual */
        GC default_gc;                  /* GC for the root root visual */
        Colormap cmap;                  /* default color map */
        unsigned long white_pixel;
        unsigned long black_pixel;      /* White and Black pixel values */
        int max_maps, min_maps;         /* max and min color maps */
        int backing_store;      /* Never, WhenMapped, Always */
        Bool save_unders;
        long root_input_mask;           /* initial root input mask */
} Screen;

/*
 * Format structure; describes ZFormat data the screen will understand.
 */
typedef struct {
        XExtData *ext_data;     /* hook for extension to hang data */
        int depth;              /* depth of this image format */
        int bits_per_pixel;     /* bits/pixel at this depth */
        int scanline_pad;       /* scanline must padded to this multiple */
} ScreenFormat;

/*
 * Data structure for setting window attributes.
 */
typedef struct {
   Pixmap background_pixmap;            /* background or None or ParentRelative */
```

```
    unsigned long background_pixel;  /* background pixel */
    Pixmap border_pixmap;     /* border of the window */
    unsigned long border_pixel;          /* border pixel value */
    int bit_gravity;                  /* one of bit gravity values */
    int win_gravity;                  /* one of the window gravity values */
    int backing_store;                /* NotUseful, WhenMapped, Always */
    unsigned long backing_planes;/* planes to be preseved if possible */
    unsigned long backing_pixel;/* value to use in restoring planes */
    Bool save_under;                  /* should bits under be saved? (popups) */
    long event_mask;                  /* set of events that should be saved */
    long do_not_propagate_mask;       /* set of events that should not propagate */
    Bool override_redirect;      /* boolean value for override-redirect */
    Colormap colormap;                     /* color map to be associated with window */
    Cursor cursor;                    /* cursor to be displayed (or None) */
} XSetWindowAttributes;

typedef struct {
    int x, y;                         /* location of window */
    int width, height;                /* width and height of window */
    int border_width;                 /* border width of window */
    int depth;                /* depth of window */
    Visual *visual;                   /* the associated visual structure */
    Window root;          /* root of screen containing window */
#if defined(__cplusplus) || defined(c_plusplus)
    int c_class;              /* C++ InputOutput, InputOnly*/
#else
    int class;                        /* InputOutput, InputOnly*/
#endif
    int bit_gravity;                  /* one of bit gravity values */
    int win_gravity;                  /* one of the window gravity values */
    int backing_store;                /* NotUseful, WhenMapped, Always */
    unsigned long backing_planes;/* planes to be preserved if possible */
    unsigned long backing_pixel;/* value to be used when restoring planes */
    Bool save_under;                  /* boolean, should bits under be saved? */
    Colormap colormap;                     /* color map to be associated with window */
    Bool map_installed;                    /* boolean, is color map currently installed*/
    int map_state;                    /* IsUnmapped, IsUnviewable, IsViewable */
    long all_event_masks;        /* set of events all people have interest in*/
    long your_event_mask;        /* my event mask */
    long do_not_propagate_mask; /* set of events that should not propagate */
    Bool override_redirect;      /* boolean value for override-redirect */
    Screen *screen;                   /* back pointer to correct screen */
} XWindowAttributes;

/*
 * Data structure for host setting; getting routines.
 *
 */
```

```
typedef struct {
        int family;            /* for example FamilyInternet */
        int length;            /* length of address, in bytes */
        char *address;         /* pointer to where to find the bytes */
} XHostAddress;

/*
 * Data structure for ServerFamilyInterpreted addresses in host routines
 */
typedef struct {
        int typelength;        /* length of type string, in bytes */
        int valuelength;       /* length of value string, in bytes */
        char *type;            /* pointer to where to find the type string */
        char *value;           /* pointer to where to find the address */
} XServerInterpretedAddress;

/*
 * Data structure for "image" data, used by image manipulation routines.
 */
typedef struct _XImage {
   int width, height;          /* size of image */
   int xoffset;        /* number of pixels offset in X direction */
   int format;                 /* XYBitmap, XYPixmap, ZPixmap */
   char *data;                 /* pointer to image data */
   int byte_order;             /* data byte order, LSBFirst, MSBFirst */
   int bitmap_unit;            /* quant. of scanline 8, 16, 32 */
   int bitmap_bit_order;       /* LSBFirst, MSBFirst */
   int bitmap_pad;             /* 8, 16, 32 either XY or ZPixmap */
   int depth;                  /* depth of image */
   int bytes_per_line;         /* accelarator to next line */
   int bits_per_pixel;         /* bits per pixel (ZPixmap) */
   unsigned long red_mask;     /* bits in z arrangment */
   unsigned long green_mask;
   unsigned long blue_mask;
   XPointer obdata;            /* hook for the object routines to hang on */
   struct funcs {              /* image manipulation routines */
        struct _XImage *(*create_image)(
                struct _XDisplay* /* display */,
                Visual*                /* visual */,
                unsigned int   /* depth */,
                int            /* format */,
                int            /* offset */,
                char*          /* data */,
                unsigned int   /* width */,
                unsigned int   /* height */,
                int            /* bitmap_pad */,
                int            /* bytes_per_line */);
        int (*destroy_image)       (struct _XImage *);
        unsigned long (*get_pixel)  (struct _XImage *, int, int);
        int (*put_pixel)            (struct _XImage *, int, int, unsigned long);
        struct _XImage *(*sub_image)(struct _XImage *, int, int, unsigned int, unsigned int);
        int (*add_pixel)            (struct _XImage *, long);
        } f;
} XImage;

/*
 * Data structure for XReconfigureWindow
 */
typedef struct {
    int x, y;
    int width, height;
    int border_width;
    Window sibling;
    int stack_mode;
} XWindowChanges;

/*
 * Data structure used by color operations
 */
typedef struct {
        unsigned long pixel;
        unsigned short red, green, blue;
        char flags;  /* do_red, do_green, do_blue */
        char pad;
} XColor;

/*
 * Data structures for graphics operations.  On most machines, these are
 * congruent with the wire protocol structures, so reformatting the data
 * can be avoided on these architectures.
 */
typedef struct {
    short x1, y1, x2, y2;
} XSegment;

typedef struct {
    short x, y;
} XPoint;

typedef struct {
    short x, y;
    unsigned short width, height;
} XRectangle;

typedef struct {
    short x, y;
    unsigned short width, height;
    short angle1, angle2;
```

```
} XArc;


/* Data structure for XChangeKeyboardControl */

typedef struct {
        int key_click_percent;
        int bell_percent;
        int bell_pitch;
        int bell_duration;
        int led;
        int led_mode;
        int key;
        int auto_repeat_mode;   /* On, Off, Default */
} XKeyboardControl;

/* Data structure for XGetKeyboardControl */

typedef struct {
        int key_click_percent;
            int bell_percent;
            unsigned int bell_pitch, bell_duration;
            unsigned long led_mask;
            int global_auto_repeat;
            char auto_repeats[32];
} XKeyboardState;

/* Data structure for XGetMotionEvents.  */

typedef struct {
        Time time;
            short x, y;
} XTimeCoord;

/* Data structure for X{Set,Get}ModifierMapping */

typedef struct {
            int max_keypermod;  /* The server's max # of keys per modifier */
            KeyCode *modifiermap;       /* An 8 by max_keypermod array of modifiers */
} XModifierKeymap;


/*
 * Display datatype maintaining display specific data.
 * The contents of this structure are implementation dependent.
 * A Display should be treated as opaque by application code.
 */
#ifndef XLIB_ILLEGAL_ACCESS
typedef struct _XDisplay Display;
```

```
#endif

struct _XPrivate;             /* Forward declare before use for C++ */
struct _XrmHashBucketRec;

typedef struct
#ifdef XLIB_ILLEGAL_ACCESS
_XDisplay
#endif
{
        XExtData *ext_data;   /* hook for extension to hang data */
        struct _XPrivate *private1;
        int fd;               /* Network socket. */
        int private2;
        int proto_major_version;/* major version of server's X protocol */
        int proto_minor_version;/* minor version of servers X protocol */
        char *vendor;         /* vendor of the server hardware */
     XID private3;
        XID private4;
        XID private5;
        int private6;
        XID (*resource_alloc)(       /* allocator function */
               struct _XDisplay*
        );
        int byte_order;              /* screen byte order, LSBFirst, MSBFirst */
        int bitmap_unit;      /* padding and data requirements */
        int bitmap_pad;              /* padding requirements on bitmaps */
        int bitmap_bit_order; /* LeastSignificant or MostSignificant */
        int nformats;         /* number of pixmap formats in list */
        ScreenFormat *pixmap_format;        /* pixmap format list */
        int private8;
        int release;          /* release of the server */
        struct _XPrivate *private9, *private10;
        int qlen;             /* Length of input event queue */
        unsigned long last_request_read; /* seq number of last event read */
        unsigned long request;       /* sequence number of last request. */
        XPointer private11;
        XPointer private12;
        XPointer private13;
        XPointer private14;
        unsigned max_request_size; /* maximum number 32 bit words in request*/
        struct _XrmHashBucketRec *db;
        int (*private15)(
               struct _XDisplay*
               );
        char *display_name;   /* "host:display" string used on this connect*/
        int default_screen;   /* default screen for operations */
        int nscreens;         /* number of screens on this server*/
        Screen *screens;      /* pointer to list of screens */
```

```
	unsigned long motion_buffer;		/* size of motion buffer */
	unsigned long private16;
	int min_keycode;	/* minimum defined keycode */
	int max_keycode;	/* maximum defined keycode */
	XPointer private17;
	XPointer private18;
	int private19;
	char *xdefaults;	/* contents of defaults from server */
	/* there is more to this structure, but it is private to Xlib */
}
#ifdef XLIB_ILLEGAL_ACCESS
Display,
#endif
*_XPrivDisplay;

#undef _XEVENT_
#ifndef _XEVENT_
/*
 * Definitions of specific events.
 */
typedef struct {
	int type;		/* of event */
	unsigned long serial;	/* # of last request processed by server */
	Bool send_event;	/* true if this came from a SendEvent request */
	Display *display;	/* Display the event was read from */
	Window window;		/* "event" window it is reported relative to */
	Window root;	/* root window that the event occurred on */
	Window subwindow; /* child window */
	Time time;		/* milliseconds */
	int x, y;		/* pointer x, y coordinates in event window */
	int x_root, y_root;	/* coordinates relative to root */
	unsigned int state;	/* key or button mask */
	unsigned int keycode;/* detail */
	Bool same_screen;	/* same screen flag */
} XKeyEvent;
typedef XKeyEvent XKeyPressedEvent;
typedef XKeyEvent XKeyReleasedEvent;

typedef struct {
	int type;		/* of event */
	unsigned long serial;	/* # of last request processed by server */
	Bool send_event;	/* true if this came from a SendEvent request */
	Display *display;	/* Display the event was read from */
	Window window;		/* "event" window it is reported relative to */
	Window root;	/* root window that the event occurred on */
	Window subwindow; /* child window */
	Time time;		/* milliseconds */
	int x, y;		/* pointer x, y coordinates in event window */
	int x_root, y_root;	/* coordinates relative to root */
```

```
        unsigned int state;      /* key or button mask */
        unsigned int button;     /* detail */
        Bool same_screen;        /* same screen flag */
} XButtonEvent;
typedef XButtonEvent XButtonPressedEvent;
typedef XButtonEvent XButtonReleasedEvent;

typedef struct {
        int type;                /* of event */
        unsigned long serial;    /* # of last request processed by server */
        Bool send_event;         /* true if this came from a SendEvent request */
        Display *display;        /* Display the event was read from */
        Window window;                 /* "event" window reported relative to */
        Window root;          /* root window that the event occurred on */
        Window subwindow;        /* child window */
        Time time;               /* milliseconds */
        int x, y;                /* pointer x, y coordinates in event window */
        int x_root, y_root;      /* coordinates relative to root */
        unsigned int state;      /* key or button mask */
        char is_hint;            /* detail */
        Bool same_screen;        /* same screen flag */
} XMotionEvent;
typedef XMotionEvent XPointerMovedEvent;

typedef struct {
        int type;                /* of event */
        unsigned long serial;    /* # of last request processed by server */
        Bool send_event;         /* true if this came from a SendEvent request */
        Display *display;        /* Display the event was read from */
        Window window;                 /* "event" window reported relative to */
        Window root;          /* root window that the event occurred on */
        Window subwindow;        /* child window */
        Time time;               /* milliseconds */
        int x, y;                /* pointer x, y coordinates in event window */
        int x_root, y_root;      /* coordinates relative to root */
        int mode;                /* NotifyNormal, NotifyGrab, NotifyUngrab */
        int detail;
        /*
         * NotifyAncestor, NotifyVirtual, NotifyInferior,
         * NotifyNonlinear,NotifyNonlinearVirtual
         */
        Bool same_screen;        /* same screen flag */
        Bool focus;              /* boolean focus */
        unsigned int state;      /* key or button mask */
} XCrossingEvent;
typedef XCrossingEvent XEnterWindowEvent;
typedef XCrossingEvent XLeaveWindowEvent;

typedef struct {
        int type;               /* FocusIn or FocusOut */
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;                  /* window of event */
        int mode;               /* NotifyNormal, NotifyWhileGrabbed,
                                   NotifyGrab, NotifyUngrab */
        int detail;
        /*
         * NotifyAncestor, NotifyVirtual, NotifyInferior,
         * NotifyNonlinear,NotifyNonlinearVirtual, NotifyPointer,
         * NotifyPointerRoot, NotifyDetailNone
         */
} XFocusChangeEvent;
typedef XFocusChangeEvent XFocusInEvent;
typedef XFocusChangeEvent XFocusOutEvent;

/* generated on EnterWindow and FocusIn  when KeyMapState selected */
typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;
        char key_vector[32];
} XKeymapEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;
        int x, y;
        int width, height;
        int count;              /* if non-zero, at least this many more */
} XExposeEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Drawable drawable;
        int x, y;
        int width, height;
        int count;              /* if non-zero, at least this many more */
        int major_code;                 /* core is CopyArea or CopyPlane */
        int minor_code;                 /* not defined in the core */
```

```
} XGraphicsExposeEvent;

typedef struct {
        int type;
        unsigned long serial;  /* # of last request processed by server */
        Bool send_event;       /* true if this came from a SendEvent request */
        Display *display;      /* Display the event was read from */
        Drawable drawable;
        int major_code;                /* core is CopyArea or CopyPlane */
        int minor_code;                /* not defined in the core */
} XNoExposeEvent;

typedef struct {
        int type;
        unsigned long serial;  /* # of last request processed by server */
        Bool send_event;       /* true if this came from a SendEvent request */
        Display *display;      /* Display the event was read from */
        Window window;
        int state;             /* Visibility state */
} XVisibilityEvent;

typedef struct {
        int type;
        unsigned long serial;  /* # of last request processed by server */
        Bool send_event;       /* true if this came from a SendEvent request */
        Display *display;      /* Display the event was read from */
        Window parent;                 /* parent of the window */
        Window window;                 /* window id of window created */
        int x, y;              /* window location */
        int width, height;     /* size of window */
        int border_width;      /* border width */
        Bool override_redirect;        /* creation should be overridden */
} XCreateWindowEvent;

typedef struct {
        int type;
        unsigned long serial;  /* # of last request processed by server */
        Bool send_event;       /* true if this came from a SendEvent request */
        Display *display;      /* Display the event was read from */
        Window event;
        Window window;
} XDestroyWindowEvent;

typedef struct {
        int type;
        unsigned long serial;  /* # of last request processed by server */
        Bool send_event;       /* true if this came from a SendEvent request */
        Display *display;      /* Display the event was read from */
        Window event;
        Window window;
        Bool from_configure;
} XUnmapEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window event;
        Window window;
        Bool override_redirect;         /* boolean, is override set... */
} XMapEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window parent;
        Window window;
} XMapRequestEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window event;
        Window window;
        Window parent;
        int x, y;
        Bool override_redirect;
} XReparentEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window event;
        Window window;
        int x, y;
        int width, height;
        int border_width;
        Window above;
        Bool override_redirect;
} XConfigureEvent;
```

```
typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window event;
        Window window;
        int x, y;
} XGravityEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;
        int width, height;
} XResizeRequestEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window parent;
        Window window;
        int x, y;
        int width, height;
        int border_width;
        Window above;
        int detail;             /* Above, Below, TopIf, BottomIf, Opposite */
        unsigned long value_mask;
} XConfigureRequestEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window event;
        Window window;
        int place;              /* PlaceOnTop, PlaceOnBottom */
} XCirculateEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window parent;
        Window window;
        int place;            /* PlaceOnTop, PlaceOnBottom */
} XCirculateRequestEvent;

typedef struct {
        int type;
        unsigned long serial; /* # of last request processed by server */
        Bool send_event;      /* true if this came from a SendEvent request */
        Display *display;     /* Display the event was read from */
        Window window;
        Atom atom;
        Time time;
        int state;            /* NewValue, Deleted */
} XPropertyEvent;

typedef struct {
        int type;
        unsigned long serial; /* # of last request processed by server */
        Bool send_event;      /* true if this came from a SendEvent request */
        Display *display;     /* Display the event was read from */
        Window window;
        Atom selection;
        Time time;
} XSelectionClearEvent;

typedef struct {
        int type;
        unsigned long serial; /* # of last request processed by server */
        Bool send_event;      /* true if this came from a SendEvent request */
        Display *display;     /* Display the event was read from */
        Window owner;
        Window requestor;
        Atom selection;
        Atom target;
        Atom property;
        Time time;
} XSelectionRequestEvent;

typedef struct {
        int type;
        unsigned long serial; /* # of last request processed by server */
        Bool send_event;      /* true if this came from a SendEvent request */
        Display *display;     /* Display the event was read from */
        Window requestor;
        Atom selection;
        Atom target;
        Atom property;                /* ATOM or None */
        Time time;
```

```
} XSelectionEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;
        Colormap colormap;      /* COLORMAP or None */
#if defined(__cplusplus) || defined(c_plusplus)
        Bool c_new;             /* C++ */
#else
        Bool new;
#endif
        int state;              /* ColormapInstalled, ColormapUninstalled */
} XColormapEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;
        Atom message_type;
        int format;
        union {
                char b[20];
                short s[10];
                long l[5];
                } data;
} XClientMessageEvent;

typedef struct {
        int type;
        unsigned long serial;   /* # of last request processed by server */
        Bool send_event;        /* true if this came from a SendEvent request */
        Display *display;       /* Display the event was read from */
        Window window;                  /* unused */
        int request;            /* one of MappingModifier, MappingKeyboard,
                                   MappingPointer */
        int first_keycode;      /* first keycode */
        int count;              /* defines range of change w. first_keycode*/
} XMappingEvent;

typedef struct {
        int type;
        Display *display;       /* Display the event was read from */
        XID resourceid;                 /* resource id */
        unsigned long serial;   /* serial number of failed request */
        unsigned char error_code;     /* error code of failed request */
        unsigned char request_code;  /* Major op-code of failed request */
        unsigned char minor_code;    /* Minor op-code of failed request */
} XErrorEvent;

typedef struct {
        int type;
        unsigned long serial;  /* # of last request processed by server */
        Bool send_event;       /* true if this came from a SendEvent request */
        Display *display;/* Display the event was read from */
        Window window;       /* window on which event was requested in event mask */
} XAnyEvent;


/***************************************************************
 *
 * GenericEvent.  This event is the standard event for all newer extensions.
 */

typedef struct
   {
   int            type;         /* of event. Always GenericEvent */
   unsigned long  serial;       /* # of last request processed */
   Bool           send_event;   /* true if from SendEvent request */
   Display        *display;     /* Display the event was read from */
   int            extension;    /* major opcode of extension that caused the event */
   int            evtype;       /* actual event type. */
   } XGenericEvent;

typedef struct {
   int            type;         /* of event. Always GenericEvent */
   unsigned long  serial;       /* # of last request processed */
   Bool           send_event;   /* true if from SendEvent request */
   Display        *display;     /* Display the event was read from */
   int            extension;    /* major opcode of extension that caused the event */
   int            evtype;       /* actual event type. */
   unsigned int   cookie;
   void           *data;
} XGenericEventCookie;

/*
 * this union is defined so Xlib can always use the same sized
 * event structure internally, to avoid memory fragmentation.
 */
typedef union _XEvent {
      int type;             /* must not be changed; first element */
        XAnyEvent xany;
        XKeyEvent xkey;
        XButtonEvent xbutton;
	XMotionEvent xmotion;
	XCrossingEvent xcrossing;
	XFocusChangeEvent xfocus;
	XExposeEvent xexpose;
	XGraphicsExposeEvent xgraphicsexpose;
	XNoExposeEvent xnoexpose;
	XVisibilityEvent xvisibility;
	XCreateWindowEvent xcreatewindow;
	XDestroyWindowEvent xdestroywindow;
	XUnmapEvent xunmap;
	XMapEvent xmap;
	XMapRequestEvent xmaprequest;
	XReparentEvent xreparent;
	XConfigureEvent xconfigure;
	XGravityEvent xgravity;
	XResizeRequestEvent xresizerequest;
	XConfigureRequestEvent xconfigurerequest;
	XCirculateEvent xcirculate;
	XCirculateRequestEvent xcirculaterequest;
	XPropertyEvent xproperty;
	XSelectionClearEvent xselectionclear;
	XSelectionRequestEvent xselectionrequest;
	XSelectionEvent xselection;
	XColormapEvent xcolormap;
	XClientMessageEvent xclient;
	XMappingEvent xmapping;
	XErrorEvent xerror;
	XKeymapEvent xkeymap;
	XGenericEvent xgeneric;
	XGenericEventCookie xcookie;
	long pad[24];
} XEvent;
#endif

#define XAllocID(dpy) ((*((_XPrivDisplay)dpy)->resource_alloc)((dpy)))

/*
 * per character font metric information.
 */
typedef struct {
    short	lbearing;	/* origin to left edge of raster */
    short	rbearing;	/* origin to right edge of raster */
    short	width;		/* advance to next char's origin */
    short	ascent;		/* baseline to top edge of raster */
    short	descent;	/* baseline to bottom edge of raster */
    unsigned short attributes;	/* per char flags (not predefined) */
} XCharStruct;

/*
```

```
 * To allow arbitrary information with fonts, there are additional properties
 * returned.
 */
typedef struct {
    Atom name;
    unsigned long card32;
} XFontProp;

typedef struct {
    XExtData    *ext_data;       /* hook for extension to hang data */
    Font        fid;             /* Font id for this font */
    unsigned    direction;       /* hint about direction the font is painted */
    unsigned    min_char_or_byte2;/* first character */
    unsigned    max_char_or_byte2;/* last character */
    unsigned    min_byte1;       /* first row that exists */
    unsigned    max_byte1;       /* last row that exists */
    Boolall_chars_exist;/* flag if all characters have non-zero size*/
    unsigned    default_char;    /* char to print for undefined character */
    int         n_properties;    /* how many properties there are */
    XFontProp   *properties;     /* pointer to array of additional properties*/
    XCharStruct         min_bounds;   /* minimum bounds over all existing char*/
    XCharStruct         max_bounds;   /* maximum bounds over all existing char*/
    XCharStruct         *per_char;    /* first_char to last_char information */
    int         ascent;          /* log. extent above baseline for spacing */
    int         descent;         /* log. descent below baseline for spacing */
} XFontStruct;

/*
 * PolyText routines take these as arguments.
 */
typedef struct {
    char *chars;            /* pointer to string */
    int nchars;                  /* number of characters */
    int delta;                   /* delta between strings */
    Font font;                   /* font to print it in, None don't change */
} XTextItem;

typedef struct {                 /* normal 16 bit characters are two bytes */
    unsigned char byte1;
    unsigned char byte2;
} XChar2b;

typedef struct {
    XChar2b *chars;              /* two byte characters */
    int nchars;                  /* number of characters */
    int delta;                   /* delta between strings */
    Font font;                   /* font to print it in, None don't change */
} XTextItem16;
```

```
typedef union { Display *display;
                GC gc;
                Visual *visual;
                Screen *screen;
                ScreenFormat *pixmap_format;
                XFontStruct *font; } XEDataObject;

typedef struct {
    XRectangle      max_ink_extent;
    XRectangle      max_logical_extent;
} XFontSetExtents;

/* unused:
typedef void (*XOMProc)();
 */

typedef struct _XOM *XOM;
typedef struct _XOC *XOC, *XFontSet;

typedef struct {
    char           *chars;
    int            nchars;
    int            delta;
    XFontSet       font_set;
} XmbTextItem;

typedef struct {
    wchar_t        *chars;
    int            nchars;
    int            delta;
    XFontSet       font_set;
} XwcTextItem;

#define XNRequiredCharSet "requiredCharSet"
#define XNQueryOrientation "queryOrientation"
#define XNBaseFontName "baseFontName"
#define XNOMAutomatic "omAutomatic"
#define XNMissingCharSet "missingCharSet"
#define XNDefaultString "defaultString"
#define XNOrientation "orientation"
#define XNDirectionalDependentDrawing "directionalDependentDrawing"
#define XNContextualDrawing "contextualDrawing"
#define XNFontInfo "fontInfo"

typedef struct {
    int charset_count;
    char **charset_list;
} XOMCharSetList;
```

```
typedef enum {
    XOMOrientation_LTR_TTB,
    XOMOrientation_RTL_TTB,
    XOMOrientation_TTB_LTR,
    XOMOrientation_TTB_RTL,
    XOMOrientation_Context
} XOrientation;

typedef struct {
    int num_orientation;
    XOrientation *orientation; /* Input Text description */
} XOMOrientation;

typedef struct {
    int num_font;
    XFontStruct **font_struct_list;
    char **font_name_list;
} XOMFontInfo;

typedef struct _XIM *XIM;
typedef struct _XIC *XIC;

typedef void (*XIMProc)(
    XIM,
    XPointer,
    XPointer
);

typedef Bool (*XICProc)(
    XIC,
    XPointer,
    XPointer
);

typedef void (*XIDProc)(
    Display*,
    XPointer,
    XPointer
);

typedef unsigned long XIMStyle;

typedef struct {
    unsigned short count_styles;
    XIMStyle *supported_styles;
} XIMStyles;

#define XIMPreeditArea          0x0001L
```

```
#define XIMPreeditCallbacks         0x0002L
#define XIMPreeditPosition  0x0004L
#define XIMPreeditNothing  0x0008L
#define XIMPreeditNone              0x0010L
#define XIMStatusArea               0x0100L
#define XIMStatusCallbacks 0x0200L
#define XIMStatusNothing   0x0400L
#define XIMStatusNone               0x0800L

#define XNVaNestedList "XNVaNestedList"
#define XNQueryInputStyle "queryInputStyle"
#define XNClientWindow "clientWindow"
#define XNInputStyle "inputStyle"
#define XNFocusWindow "focusWindow"
#define XNResourceName "resourceName"
#define XNResourceClass "resourceClass"
#define XNGeometryCallback "geometryCallback"
#define XNDestroyCallback "destroyCallback"
#define XNFilterEvents "filterEvents"
#define XNPreeditStartCallback "preeditStartCallback"
#define XNPreeditDoneCallback "preeditDoneCallback"
#define XNPreeditDrawCallback "preeditDrawCallback"
#define XNPreeditCaretCallback "preeditCaretCallback"
#define XNPreeditStateNotifyCallback "preeditStateNotifyCallback"
#define XNPreeditAttributes "preeditAttributes"
#define XNStatusStartCallback "statusStartCallback"
#define XNStatusDoneCallback "statusDoneCallback"
#define XNStatusDrawCallback "statusDrawCallback"
#define XNStatusAttributes "statusAttributes"
#define XNArea "area"
#define XNAreaNeeded "areaNeeded"
#define XNSpotLocation "spotLocation"
#define XNColormap "colorMap"
#define XNStdColormap "stdColorMap"
#define XNForeground "foreground"
#define XNBackground "background"
#define XNBackgroundPixmap "backgroundPixmap"
#define XNFontSet "fontSet"
#define XNLineSpace "lineSpace"
#define XNCursor "cursor"

#define XNQueryIMValuesList "queryIMValuesList"
#define XNQueryICValuesList "queryICValuesList"
#define XNVisiblePosition "visiblePosition"
#define XNR6PreeditCallback "r6PreeditCallback"
#define XNStringConversionCallback "stringConversionCallback"
#define XNStringConversion "stringConversion"
#define XNResetState "resetState"
#define XNHotKey "hotKey"
```

```
#define XNHotKeyState "hotKeyState"
#define XNPreeditState "preeditState"
#define XNSeparatorofNestedList "separatorofNestedList"

#define XBufferOverflow         -1
#define XLookupNone             1
#define XLookupChars            2
#define XLookupKeySym           3
#define XLookupBoth             4

typedef void *XVaNestedList;

typedef struct {
  XPointer client_data;
  XIMProc callback;
} XIMCallback;

typedef struct {
  XPointer client_data;
  XICProc callback;
} XICCallback;

typedef unsigned long XIMFeedback;

#define XIMReverse       1L
#define XIMUnderline            (1L<<1)
#define XIMHighlight            (1L<<2)
#define XIMPrimary       (1L<<5)
#define XIMSecondary            (1L<<6)
#define XIMTertiary      (1L<<7)
#define XIMVisibleToForward     (1L<<8)
#define XIMVisibleToBackword    (1L<<9)
#define XIMVisibleToCenter      (1L<<10)

typedef struct _XIMText {
  unsigned short length;
  XIMFeedback *feedback;
  Bool encoding_is_wchar;
  union {
      char *multi_byte;
      wchar_t *wide_char;
  } string;
} XIMText;

typedef      unsigned long  XIMPreeditState;

#define      XIMPreeditUnKnown       0L
#define      XIMPreeditEnable  1L
#define      XIMPreeditDisable (1L<<1)
```

```
typedef         struct   _XIMPreeditStateNotifyCallbackStruct {
   XIMPreeditState state;
} XIMPreeditStateNotifyCallbackStruct;

typedef         unsigned long  XIMResetState;

#define         XIMInitialState                 1L
#define         XIMPreserveState      (1L<<1)

typedef unsigned long XIMStringConversionFeedback;

#define         XIMStringConversionLeftEdge       (0x00000001)
#define         XIMStringConversionRightEdge      (0x00000002)
#define         XIMStringConversionTopEdge        (0x00000004)
#define         XIMStringConversionBottomEdge     (0x00000008)
#define         XIMStringConversionConcealed      (0x00000010)
#define         XIMStringConversionWrapped        (0x00000020)

typedef struct _XIMStringConversionText {
   unsigned short length;
   XIMStringConversionFeedback *feedback;
   Bool encoding_is_wchar;
   union {
         char *mbs;
         wchar_t *wcs;
   } string;
} XIMStringConversionText;

typedef         unsigned shortXIMStringConversionPosition;

typedef         unsigned shortXIMStringConversionType;

#define         XIMStringConversionBuffer (0x0001)
#define         XIMStringConversionLine           (0x0002)
#define         XIMStringConversionWord           (0x0003)
#define         XIMStringConversionChar           (0x0004)

typedef         unsigned shortXIMStringConversionOperation;

#define         XIMStringConversionSubstitution   (0x0001)
#define         XIMStringConversionRetrieval      (0x0002)

typedef enum {
   XIMForwardChar, XIMBackwardChar,
   XIMForwardWord, XIMBackwardWord,
   XIMCaretUp, XIMCaretDown,
   XIMNextLine, XIMPreviousLine,
   XIMLineStart, XIMLineEnd,
   XIMAbsolutePosition,
   XIMDontChange
} XIMCaretDirection;

typedef struct _XIMStringConversionCallbackStruct {
   XIMStringConversionPosition position;
   XIMCaretDirection direction;
   XIMStringConversionOperation operation;
   unsigned short factor;
   XIMStringConversionText *text;
} XIMStringConversionCallbackStruct;

typedef struct _XIMPreeditDrawCallbackStruct {
   int caret;              /* Cursor offset within pre-edit string */
   int chg_first;          /* Starting change position */
   int chg_length;         /* Length of the change in character count */
   XIMText *text;
} XIMPreeditDrawCallbackStruct;

typedef enum {
   XIMIsInvisible,         /* Disable caret feedback */
   XIMIsPrimary,           /* UI defined caret feedback */
   XIMIsSecondary          /* UI defined caret feedback */
} XIMCaretStyle;

typedef struct _XIMPreeditCaretCallbackStruct {
   int position;            /* Caret offset within pre-edit string */
   XIMCaretDirection direction; /* Caret moves direction */
   XIMCaretStyle style;         /* Feedback of the caret */
} XIMPreeditCaretCallbackStruct;

typedef enum {
   XIMTextType,
   XIMBitmapType
} XIMStatusDataType;

typedef struct _XIMStatusDrawCallbackStruct {
   XIMStatusDataType type;
   union {
         XIMText *text;
         Pixmap  bitmap;
   } data;
} XIMStatusDrawCallbackStruct;

typedef struct _XIMHotKeyTrigger {
   KeySym      keysym;
   int         modifier;
   int         modifier_mask;
} XIMHotKeyTrigger;
```

```
typedef struct _XIMHotKeyTriggers {
    int                         num_hot_key;
    XIMHotKeyTrigger            *key;
} XIMHotKeyTriggers;

typedef         unsigned long  XIMHotKeyState;

#define         XIMHotKeyStateON (0x0001L)
#define         XIMHotKeyStateOFF           (0x0002L)

typedef struct {
    unsigned short count_values;
    char **supported_values;
} XIMValuesList;

_XFUNCPROTOBEGIN

#if defined(WIN32) && !defined(_XLIBINT_)
#define _Xdebug (*_Xdebug_p)
#endif

extern int _Xdebug;

extern XFontStruct *XLoadQueryFont(
    Display*                /* display */,
    _Xconst char*           /* name */
);

extern XFontStruct *XQueryFont(
    Display*                /* display */,
    XID                     /* font_ID */
);


extern XTimeCoord *XGetMotionEvents(
    Display*                /* display */,
    Window                  /* w */,
    Time                    /* start */,
    Time                    /* stop */,
    int*            /* nevents_return */
);

extern XModifierKeymap *XDeleteModifiermapEntry(
    XModifierKeymap*            /* modmap */,
#if NeedWidePrototypes
    unsigned int            /* keycode_entry */,
#else
    KeyCode                 /* keycode_entry */,
#endif
    int                     /* modifier */
);

extern XModifierKeymap      *XGetModifierMapping(
    Display*                /* display */
);

extern XModifierKeymap      *XInsertModifiermapEntry(
    XModifierKeymap*            /* modmap */,
#if NeedWidePrototypes
    unsigned int            /* keycode_entry */,
#else
    KeyCode                 /* keycode_entry */,
#endif
    int                     /* modifier */
);

extern XModifierKeymap *XNewModifiermap(
    int                     /* max_keys_per_mod */
);

extern XImage *XCreateImage(
    Display*                /* display */,
    Visual*                 /* visual */,
    unsigned int            /* depth */,
    int                     /* format */,
    int                     /* offset */,
    char*                   /* data */,
    unsigned int            /* width */,
    unsigned int            /* height */,
    int                     /* bitmap_pad */,
    int                     /* bytes_per_line */
);
extern Status XInitImage(
    XImage*                 /* image */
);
extern XImage *XGetImage(
    Display*                /* display */,
    Drawable                /* d */,
    int                     /* x */,
    int                     /* y */,
    unsigned int            /* width */,
    unsigned int            /* height */,
    unsigned long           /* plane_mask */,
    int                     /* format */
);
extern XImage *XGetSubImage(
    Display*                /* display */,
    Drawable            /* d */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned long       /* plane_mask */,
    int                 /* format */,
    XImage*             /* dest_image */,
    int                 /* dest_x */,
    int                 /* dest_y */
);

/*
 * X function declarations.
 */
extern Display *XOpenDisplay(
    _Xconst char*       /* display_name */
);

extern void XrmInitialize(
    void
);

extern char *XFetchBytes(
    Display*            /* display */,
    int*        /* nbytes_return */
);
extern char *XFetchBuffer(
    Display*            /* display */,
    int*        /* nbytes_return */,
    int                 /* buffer */
);
extern char *XGetAtomName(
    Display*            /* display */,
    Atom                /* atom */
);
extern Status XGetAtomNames(
    Display*            /* dpy */,
    Atom*               /* atoms */,
    int                 /* count */,
    char**              /* names_return */
);
extern char *XGetDefault(
    Display*            /* display */,
    _Xconst char*       /* program */,
    _Xconst char*       /* option */
);
extern char *XDisplayName(
    _Xconst char*       /* string */
```

```
);
extern char *XKeysymToString(
    KeySym                  /* keysym */
);

extern int (*XSynchronize(
    Display*                /* display */,
    Bool            /* onoff */
))(
    Display*                /* display */
);
extern int (*XSetAfterFunction(
    Display*                /* display */,
    int (*) (
                Display*    /* display */
            )               /* procedure */
))(
    Display*                /* display */
);
extern Atom XInternAtom(
    Display*                /* display */,
    _Xconst char*           /* atom_name */,
    Bool            /* only_if_exists */
);
extern Status XInternAtoms(
    Display*                /* dpy */,
    char**                  /* names */,
    int                     /* count */,
    Bool            /* onlyIfExists */,
    Atom*                   /* atoms_return */
);
extern Colormap XCopyColormapAndFree(
    Display*                /* display */,
    Colormap                /* colormap */
);
extern Colormap XCreateColormap(
    Display*                /* display */,
    Window                  /* w */,
    Visual*                 /* visual */,
    int                     /* alloc */
);
extern Cursor XCreatePixmapCursor(
    Display*                /* display */,
    Pixmap                  /* source */,
    Pixmap                  /* mask */,
    XColor*                 /* foreground_color */,
    XColor*                 /* background_color */,
    unsigned int            /* x */,
    unsigned int            /* y */
```

```
);
extern Cursor XCreateGlyphCursor(
    Display*            /* display */,
    Font        /* source_font */,
    Font        /* mask_font */,
    unsigned int        /* source_char */,
    unsigned int        /* mask_char */,
    XColor _Xconst *    /* foreground_color */,
    XColor _Xconst *    /* background_color */
);
extern Cursor XCreateFontCursor(
    Display*            /* display */,
    unsigned int        /* shape */
);
extern Font XLoadFont(
    Display*            /* display */,
    _Xconst char*       /* name */
);
extern GC XCreateGC(
    Display*            /* display */,
    Drawable            /* d */,
    unsigned long       /* valuemask */,
    XGCValues*                  /* values */
);
extern GContext XGContextFromGC(
    GC                  /* gc */
);
extern void XFlushGC(
    Display*            /* display */,
    GC                  /* gc */
);
extern Pixmap XCreatePixmap(
    Display*            /* display */,
    Drawable            /* d */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned int        /* depth */
);
extern Pixmap XCreateBitmapFromData(
    Display*            /* display */,
    Drawable            /* d */,
    _Xconst char*       /* data */,
    unsigned int        /* width */,
    unsigned int        /* height */
);
extern Pixmap XCreatePixmapFromBitmapData(
    Display*            /* display */,
    Drawable            /* d */,
    char*               /* data */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned long       /* fg */,
    unsigned long       /* bg */,
    unsigned int        /* depth */
);
extern Window XCreateSimpleWindow(
    Display*            /* display */,
    Window              /* parent */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned int        /* border_width */,
    unsigned long       /* border */,
    unsigned long       /* background */
);
extern Window XGetSelectionOwner(
    Display*            /* display */,
    Atom                /* selection */
);
extern Window XCreateWindow(
    Display*            /* display */,
    Window              /* parent */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned int        /* border_width */,
    int                 /* depth */,
    unsigned int        /* class */,
    Visual*             /* visual */,
    unsigned long       /* valuemask */,
    XSetWindowAttributes*    /* attributes */
);
extern Colormap *XListInstalledColormaps(
    Display*            /* display */,
    Window              /* w */,
    int*            /* num_return */
);
extern char **XListFonts(
    Display*            /* display */,
    _Xconst char*       /* pattern */,
    int                 /* maxnames */,
    int*            /* actual_count_return */
);
extern char **XListFontsWithInfo(
    Display*            /* display */,
    _Xconst char*       /* pattern */,
    int                   /* maxnames */,
    int*           /* count_return */,
    XFontStruct**         /* info_return */
);
extern char **XGetFontPath(
    Display*              /* display */,
    int*           /* npaths_return */
);
extern char **XListExtensions(
    Display*              /* display */,
    int*           /* nextensions_return */
);
extern Atom *XListProperties(
    Display*              /* display */,
    Window                /* w */,
    int*           /* num_prop_return */
);
extern XHostAddress *XListHosts(
    Display*              /* display */,
    int*           /* nhosts_return */,
    Bool*                 /* state_return */
);
_X_DEPRECATED
extern KeySym XKeycodeToKeysym(
    Display*              /* display */,
#if NeedWidePrototypes
    unsigned int          /* keycode */,
#else
    KeyCode               /* keycode */,
#endif
    int                   /* index */
);
extern KeySym XLookupKeysym(
    XKeyEvent*                   /* key_event */,
    int                   /* index */
);
extern KeySym *XGetKeyboardMapping(
    Display*              /* display */,
#if NeedWidePrototypes
    unsigned int          /* first_keycode */,
#else
    KeyCode               /* first_keycode */,
#endif
    int                   /* keycode_count */,
    int*           /* keysyms_per_keycode_return */
);
extern KeySym XStringToKeysym(
    _Xconst char*         /* string */
);
```

```
extern long XMaxRequestSize(
   Display*              /* display */
);
extern long XExtendedMaxRequestSize(
   Display*              /* display */
);
extern char *XResourceManagerString(
   Display*              /* display */
);
extern char *XScreenResourceString(
        Screen*                   /* screen */
);
extern unsigned long XDisplayMotionBufferSize(
   Display*              /* display */
);
extern VisualID XVisualIDFromVisual(
   Visual*               /* visual */
);

/* multithread routines */

extern Status XInitThreads(
   void
);

extern void XLockDisplay(
   Display*              /* display */
);

extern void XUnlockDisplay(
   Display*              /* display */
);

/* routines for dealing with extensions */

extern XExtCodes *XInitExtension(
   Display*              /* display */,
   _Xconst char*         /* name */
);

extern XExtCodes *XAddExtension(
   Display*              /* display */
);
extern XExtData *XFindOnExtensionList(
   XExtData**                     /* structure */,
   int                   /* number */
);
extern XExtData **XEHeadOfExtensionList(
   XEDataObject          /* object */
```

```
);

/* these are routines for which there are also macros */
extern Window XRootWindow(
    Display*                /* display */,
    int                     /* screen_number */
);
extern Window XDefaultRootWindow(
    Display*                /* display */
);
extern Window XRootWindowOfScreen(
    Screen*                 /* screen */
);
extern Visual *XDefaultVisual(
    Display*                /* display */,
    int                     /* screen_number */
);
extern Visual *XDefaultVisualOfScreen(
    Screen*                 /* screen */
);
extern GC XDefaultGC(
    Display*                /* display */,
    int                     /* screen_number */
);
extern GC XDefaultGCOfScreen(
    Screen*                 /* screen */
);
extern unsigned long XBlackPixel(
    Display*                /* display */,
    int                     /* screen_number */
);
extern unsigned long XWhitePixel(
    Display*                /* display */,
    int                     /* screen_number */
);
extern unsigned long XAllPlanes(
    void
);
extern unsigned long XBlackPixelOfScreen(
    Screen*                 /* screen */
);
extern unsigned long XWhitePixelOfScreen(
    Screen*                 /* screen */
);
extern unsigned long XNextRequest(
    Display*                /* display */
);
extern unsigned long XLastKnownRequestProcessed(
    Display*                /* display */
```

```
);
extern char *XServerVendor(
    Display*            /* display */
);
extern char *XDisplayString(
    Display*            /* display */
);
extern Colormap XDefaultColormap(
    Display*            /* display */,
    int                 /* screen_number */
);
extern Colormap XDefaultColormapOfScreen(
    Screen*             /* screen */
);
extern Display *XDisplayOfScreen(
    Screen*             /* screen */
);
extern Screen *XScreenOfDisplay(
    Display*            /* display */,
    int                 /* screen_number */
);
extern Screen *XDefaultScreenOfDisplay(
    Display*            /* display */
);
extern long XEventMaskOfScreen(
    Screen*             /* screen */
);

extern int XScreenNumberOfScreen(
    Screen*             /* screen */
);

typedef int (*XErrorHandler) (          /* WARNING, this type not in Xlib spec */
    Display*            /* display */,
    XErrorEvent*        /* error_event */
);

extern XErrorHandler XSetErrorHandler (
    XErrorHandler       /* handler */
);


typedef int (*XIOErrorHandler) (    /* WARNING, this type not in Xlib spec */
    Display*            /* display */
);

extern XIOErrorHandler XSetIOErrorHandler (
    XIOErrorHandler     /* handler */
);
```

```
extern XPixmapFormatValues *XListPixmapFormats(
    Display*                /* display */,
    int*            /* count_return */
);
extern int *XListDepths(
    Display*                /* display */,
    int                     /* screen_number */,
    int*            /* count_return */
);

/* ICCCM routines for things that don't require special include files; */
/* other declarations are given in Xutil.h                              */
extern Status XReconfigureWMWindow(
    Display*                /* display */,
    Window                  /* w */,
    int                     /* screen_number */,
    unsigned int            /* mask */,
    XWindowChanges*                 /* changes */
);

extern Status XGetWMProtocols(
    Display*                /* display */,
    Window                  /* w */,
    Atom**                  /* protocols_return */,
    int*            /* count_return */
);
extern Status XSetWMProtocols(
    Display*                /* display */,
    Window                  /* w */,
    Atom*                   /* protocols */,
    int                     /* count */
);
extern Status XIconifyWindow(
    Display*                /* display */,
    Window                  /* w */,
    int                     /* screen_number */
);
extern Status XWithdrawWindow(
    Display*                /* display */,
    Window                  /* w */,
    int                     /* screen_number */
);
extern Status XGetCommand(
    Display*                /* display */,
    Window                  /* w */,
    char***                 /* argv_return */,
    int*            /* argc_return */
);
extern Status XGetWMColormapWindows(
   Display*             /* display */,
   Window               /* w */,
   Window**             /* windows_return */,
   int*         /* count_return */
);
extern Status XSetWMColormapWindows(
   Display*             /* display */,
   Window               /* w */,
   Window*              /* colormap_windows */,
   int                  /* count */
);
extern void XFreeStringList(
   char**               /* list */
);
extern int XSetTransientForHint(
   Display*             /* display */,
   Window               /* w */,
   Window               /* prop_window */
);

/* The following are given in alphabetical order */

extern int XActivateScreenSaver(
   Display*             /* display */
);

extern int XAddHost(
   Display*             /* display */,
   XHostAddress*        /* host */
);

extern int XAddHosts(
   Display*             /* display */,
   XHostAddress*        /* hosts */,
   int                  /* num_hosts */
);

extern int XAddToExtensionList(
   struct _XExtData**           /* structure */,
   XExtData*            /* ext_data */
);

extern int XAddToSaveSet(
   Display*             /* display */,
   Window               /* w */
);
```

```
extern Status XAllocColor(
    Display*                /* display */,
    Colormap                /* colormap */,
    XColor*                 /* screen_in_out */
);

extern Status XAllocColorCells(
    Display*                /* display */,
    Colormap                /* colormap */,
    Bool        /* contig */,
    unsigned long*          /* plane_masks_return */,
    unsigned int            /* nplanes */,
    unsigned long*          /* pixels_return */,
    unsigned int            /* npixels */
);

extern Status XAllocColorPlanes(
    Display*                /* display */,
    Colormap                /* colormap */,
    Bool            /* contig */,
    unsigned long*          /* pixels_return */,
    int                     /* ncolors */,
    int                     /* nreds */,
    int                     /* ngreens */,
    int                     /* nblues */,
    unsigned long*          /* rmask_return */,
    unsigned long*          /* gmask_return */,
    unsigned long*          /* bmask_return */
);

extern Status XAllocNamedColor(
    Display*                /* display */,
    Colormap                /* colormap */,
    _Xconst char*           /* color_name */,
    XColor*                 /* screen_def_return */,
    XColor*                 /* exact_def_return */
);

extern int XAllowEvents(
    Display*                /* display */,
    int                     /* event_mode */,
    Time                    /* time */
);

extern int XAutoRepeatOff(
    Display*                /* display */
);

extern int XAutoRepeatOn(
    Display*            /* display */
);

extern int XBell(
    Display*            /* display */,
    int                 /* percent */
);

extern int XBitmapBitOrder(
    Display*            /* display */
);

extern int XBitmapPad(
    Display*            /* display */
);

extern int XBitmapUnit(
    Display*            /* display */
);

extern int XCellsOfScreen(
    Screen*             /* screen */
);

extern int XChangeActivePointerGrab(
    Display*            /* display */,
    unsigned int        /* event_mask */,
    Cursor              /* cursor */,
    Time                /* time */
);

extern int XChangeGC(
    Display*            /* display */,
    GC                  /* gc */,
    unsigned long       /* valuemask */,
    XGCValues*                  /* values */
);

extern int XChangeKeyboardControl(
    Display*            /* display */,
    unsigned long       /* value_mask */,
    XKeyboardControl*           /* values */
);

extern int XChangeKeyboardMapping(
    Display*            /* display */,
    int                 /* first_keycode */,
    int                 /* keysyms_per_keycode */,
    KeySym*             /* keysyms */,
    int                     /* num_codes */
);

extern int XChangePointerControl(
    Display*                /* display */,
    Bool            /* do_accel */,
    Bool            /* do_threshold */,
    int                     /* accel_numerator */,
    int                     /* accel_denominator */,
    int                     /* threshold */
);

extern int XChangeProperty(
    Display*                /* display */,
    Window                  /* w */,
    Atom                    /* property */,
    Atom                    /* type */,
    int                     /* format */,
    int                     /* mode */,
    _Xconst unsigned char*    /* data */,
    int                     /* nelements */
);

extern int XChangeSaveSet(
    Display*                /* display */,
    Window                  /* w */,
    int                     /* change_mode */
);

extern int XChangeWindowAttributes(
    Display*                /* display */,
    Window                  /* w */,
    unsigned long           /* valuemask */,
    XSetWindowAttributes* /* attributes */
);

extern Bool XCheckIfEvent(
    Display*                /* display */,
    XEvent*                 /* event_return */,
    Bool (*) (
               Display*                     /* display */,
           XEvent*                          /* event */,
           XPointer                         /* arg */
         )                  /* predicate */,
    XPointer                /* arg */
);

extern Bool XCheckMaskEvent(
    Display*                /* display */,
    long           /* event_mask */,
    XEvent*                /* event_return */
);

extern Bool XCheckTypedEvent(
    Display*               /* display */,
    int                    /* event_type */,
    XEvent*                /* event_return */
);

extern Bool XCheckTypedWindowEvent(
    Display*               /* display */,
    Window                 /* w */,
    int                    /* event_type */,
    XEvent*                /* event_return */
);

extern Bool XCheckWindowEvent(
    Display*               /* display */,
    Window                 /* w */,
    long           /* event_mask */,
    XEvent*                /* event_return */
);

extern int XCirculateSubwindows(
    Display*               /* display */,
    Window                 /* w */,
    int                    /* direction */
);

extern int XCirculateSubwindowsDown(
    Display*               /* display */,
    Window                 /* w */
);

extern int XCirculateSubwindowsUp(
    Display*               /* display */,
    Window                 /* w */
);

extern int XClearArea(
    Display*               /* display */,
    Window                 /* w */,
    int                    /* x */,
    int                    /* y */,
    unsigned int           /* width */,
    unsigned int           /* height */,
    Bool           /* exposures */
);
```

```
extern int XClearWindow(
   Display*          /* display */,
   Window            /* w */
);

extern int XCloseDisplay(
   Display*          /* display */
);

extern int XConfigureWindow(
   Display*          /* display */,
   Window            /* w */,
   unsigned int      /* value_mask */,
   XWindowChanges*         /* values */
);

extern int XConnectionNumber(
   Display*          /* display */
);

extern int XConvertSelection(
   Display*          /* display */,
   Atom              /* selection */,
   Atom              /* target */,
   Atom              /* property */,
   Window            /* requestor */,
   Time              /* time */
);

extern int XCopyArea(
   Display*          /* display */,
   Drawable          /* src */,
   Drawable          /* dest */,
   GC                /* gc */,
   int               /* src_x */,
   int               /* src_y */,
   unsigned int      /* width */,
   unsigned int      /* height */,
   int               /* dest_x */,
   int               /* dest_y */
);

extern int XCopyGC(
   Display*          /* display */,
   GC                /* src */,
   unsigned long     /* valuemask */,
   GC                /* dest */
);
```

```
extern int XCopyPlane(
    Display*            /* display */,
    Drawable            /* src */,
    Drawable            /* dest */,
    GC                  /* gc */,
    int                 /* src_x */,
    int                 /* src_y */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    int                 /* dest_x */,
    int                 /* dest_y */,
    unsigned long       /* plane */
);

extern int XDefaultDepth(
    Display*            /* display */,
    int                 /* screen_number */
);

extern int XDefaultDepthOfScreen(
    Screen*             /* screen */
);

extern int XDefaultScreen(
    Display*            /* display */
);

extern int XDefineCursor(
    Display*            /* display */,
    Window              /* w */,
    Cursor              /* cursor */
);

extern int XDeleteProperty(
    Display*            /* display */,
    Window              /* w */,
    Atom                /* property */
);

extern int XDestroyWindow(
    Display*            /* display */,
    Window              /* w */
);

extern int XDestroySubwindows(
    Display*            /* display */,
    Window              /* w */
);
```

```
extern int XDoesBackingStore(
    Screen*                 /* screen */
);

extern Bool XDoesSaveUnders(
    Screen*                 /* screen */
);

extern int XDisableAccessControl(
    Display*                /* display */
);


extern int XDisplayCells(
    Display*                /* display */,
    int                     /* screen_number */
);

extern int XDisplayHeight(
    Display*                /* display */,
    int                     /* screen_number */
);

extern int XDisplayHeightMM(
    Display*                /* display */,
    int                     /* screen_number */
);

extern int XDisplayKeycodes(
    Display*                /* display */,
    int*            /* min_keycodes_return */,
    int*            /* max_keycodes_return */
);

extern int XDisplayPlanes(
    Display*                /* display */,
    int                     /* screen_number */
);

extern int XDisplayWidth(
    Display*                /* display */,
    int                     /* screen_number */
);

extern int XDisplayWidthMM(
    Display*                /* display */,
    int                     /* screen_number */
);
```

```
extern int XDrawArc(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    int                 /* angle1 */,
    int                 /* angle2 */
);

extern int XDrawArcs(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    XArc*               /* arcs */,
    int                 /* narcs */
);

extern int XDrawImageString(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    _Xconst char*       /* string */,
    int                 /* length */
);

extern int XDrawImageString16(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    _Xconst XChar2b*/* string */,
    int                 /* length */
);

extern int XDrawLine(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x1 */,
    int                 /* y1 */,
    int                 /* x2 */,
    int                 /* y2 */
```

```
);

extern int XDrawLines(
    Display*                /* display */,
    Drawable                /* d */,
    GC                      /* gc */,
    XPoint*                 /* points */,
    int                     /* npoints */,
    int                     /* mode */
);

extern int XDrawPoint(
    Display*                /* display */,
    Drawable                /* d */,
    GC                      /* gc */,
    int                     /* x */,
    int                     /* y */
);

extern int XDrawPoints(
    Display*                /* display */,
    Drawable                /* d */,
    GC                      /* gc */,
    XPoint*                 /* points */,
    int                     /* npoints */,
    int                     /* mode */
);

extern int XDrawRectangle(
    Display*                /* display */,
    Drawable                /* d */,
    GC                      /* gc */,
    int                     /* x */,
    int                     /* y */,
    unsigned int            /* width */,
    unsigned int            /* height */
);

extern int XDrawRectangles(
    Display*                /* display */,
    Drawable                /* d */,
    GC                      /* gc */,
    XRectangle*                     /* rectangles */,
    int                     /* nrectangles */
);

extern int XDrawSegments(
    Display*                /* display */,
    Drawable                /* d */,
    GC                  /* gc */,
    XSegment*           /* segments */,
    int                 /* nsegments */
);

extern int XDrawString(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    _Xconst char*       /* string */,
    int                 /* length */
);

extern int XDrawString16(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    _Xconst XChar2b*/* string */,
    int                 /* length */
);

extern int XDrawText(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    XTextItem*          /* items */,
    int                 /* nitems */
);

extern int XDrawText16(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    XTextItem16*        /* items */,
    int                 /* nitems */
);

extern int XEnableAccessControl(
    Display*            /* display */
);
```

```
extern int XEventsQueued(
    Display*            /* display */,
    int                 /* mode */
);

extern Status XFetchName(
    Display*            /* display */,
    Window              /* w */,
    char**              /* window_name_return */
);

extern int XFillArc(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    int                 /* angle1 */,
    int                 /* angle2 */
);

extern int XFillArcs(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    XArc*               /* arcs */,
    int                 /* narcs */
);

extern int XFillPolygon(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    XPoint*             /* points */,
    int                 /* npoints */,
    int                 /* shape */,
    int                 /* mode */
);

extern int XFillRectangle(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */
);

extern int XFillRectangles(
    Display*            /* display */,
    Drawable            /* d */,
    GC                  /* gc */,
    XRectangle*                 /* rectangles */,
    int                 /* nrectangles */
);

extern int XFlush(
    Display*            /* display */
);

extern int XForceScreenSaver(
    Display*            /* display */,
    int                 /* mode */
);

extern int XFree(
    void*               /* data */
);

extern int XFreeColormap(
    Display*            /* display */,
    Colormap            /* colormap */
);

extern int XFreeColors(
    Display*            /* display */,
    Colormap            /* colormap */,
    unsigned long*      /* pixels */,
    int                 /* npixels */,
    unsigned long       /* planes */
);

extern int XFreeCursor(
    Display*            /* display */,
    Cursor              /* cursor */
);

extern int XFreeExtensionList(
    char**              /* list */
);

extern int XFreeFont(
    Display*            /* display */,
    XFontStruct*        /* font_struct */
);
```

```
extern int XFreeFontInfo(
   char**                 /* names */,
   XFontStruct*           /* free_info */,
   int                    /* actual_count */
);

extern int XFreeFontNames(
   char**                 /* list */
);

extern int XFreeFontPath(
   char**                 /* list */
);

extern int XFreeGC(
   Display*               /* display */,
   GC                     /* gc */
);

extern int XFreeModifiermap(
   XModifierKeymap*               /* modmap */
);

extern int XFreePixmap(
   Display*               /* display */,
   Pixmap                 /* pixmap */
);

extern int XGeometry(
   Display*               /* display */,
   int                    /* screen */,
   _Xconst char*          /* position */,
   _Xconst char*          /* default_position */,
   unsigned int           /* bwidth */,
   unsigned int           /* fwidth */,
   unsigned int           /* fheight */,
   int                    /* xadder */,
   int                    /* yadder */,
   int*           /* x_return */,
   int*           /* y_return */,
   int*           /* width_return */,
   int*           /* height_return */
);

extern int XGetErrorDatabaseText(
   Display*               /* display */,
   _Xconst char*          /* name */,
   _Xconst char*          /* message */,
   _Xconst char*       /* default_string */,
   char*               /* buffer_return */,
   int                 /* length */
);

extern int XGetErrorText(
   Display*            /* display */,
   int                 /* code */,
   char*               /* buffer_return */,
   int                 /* length */
);

extern Bool XGetFontProperty(
   XFontStruct*        /* font_struct */,
   Atom                /* atom */,
   unsigned long*      /* value_return */
);

extern Status XGetGCValues(
   Display*            /* display */,
   GC                  /* gc */,
   unsigned long       /* valuemask */,
   XGCValues*                  /* values_return */
);

extern Status XGetGeometry(
   Display*            /* display */,
   Drawable            /* d */,
   Window*             /* root_return */,
   int*        /* x_return */,
   int*        /* y_return */,
   unsigned int*       /* width_return */,
   unsigned int*       /* height_return */,
   unsigned int*       /* border_width_return */,
   unsigned int*       /* depth_return */
);

extern Status XGetIconName(
   Display*            /* display */,
   Window              /* w */,
   char**              /* icon_name_return */
);

extern int XGetInputFocus(
   Display*            /* display */,
   Window*             /* focus_return */,
   int*        /* revert_to_return */
);
```

```
extern int XGetKeyboardControl(
    Display*            /* display */,
    XKeyboardState*     /* values_return */
);

extern int XGetPointerControl(
    Display*            /* display */,
    int*        /* accel_numerator_return */,
    int*        /* accel_denominator_return */,
    int*        /* threshold_return */
);

extern int XGetPointerMapping(
    Display*            /* display */,
    unsigned char*      /* map_return */,
    int                 /* nmap */
);

extern int XGetScreenSaver(
    Display*            /* display */,
    int*        /* timeout_return */,
    int*        /* interval_return */,
    int*        /* prefer_blanking_return */,
    int*        /* allow_exposures_return */
);

extern Status XGetTransientForHint(
    Display*            /* display */,
    Window              /* w */,
    Window*             /* prop_window_return */
);

extern int XGetWindowProperty(
    Display*            /* display */,
    Window              /* w */,
    Atom                /* property */,
    long        /* long_offset */,
    long        /* long_length */,
    Bool        /* delete */,
    Atom                /* req_type */,
    Atom*               /* actual_type_return */,
    int*        /* actual_format_return */,
    unsigned long*      /* nitems_return */,
    unsigned long*      /* bytes_after_return */,
    unsigned char**     /* prop_return */
);

extern Status XGetWindowAttributes(
    Display*            /* display */,
    Window                  /* w */,
    XWindowAttributes*          /* window_attributes_return */
);

extern int XGrabButton(
    Display*                /* display */,
    unsigned int            /* button */,
    unsigned int            /* modifiers */,
    Window                  /* grab_window */,
    Bool            /* owner_events */,
    unsigned int            /* event_mask */,
    int                     /* pointer_mode */,
    int                     /* keyboard_mode */,
    Window                  /* confine_to */,
    Cursor                  /* cursor */
);

extern int XGrabKey(
    Display*                /* display */,
    int                     /* keycode */,
    unsigned int            /* modifiers */,
    Window                  /* grab_window */,
    Bool            /* owner_events */,
    int                     /* pointer_mode */,
    int                     /* keyboard_mode */
);

extern int XGrabKeyboard(
    Display*                /* display */,
    Window                  /* grab_window */,
    Bool            /* owner_events */,
    int                     /* pointer_mode */,
    int                     /* keyboard_mode */,
    Time                    /* time */
);

extern int XGrabPointer(
    Display*                /* display */,
    Window                  /* grab_window */,
    Bool            /* owner_events */,
    unsigned int            /* event_mask */,
    int                     /* pointer_mode */,
    int                     /* keyboard_mode */,
    Window                  /* confine_to */,
    Cursor                  /* cursor */,
    Time                    /* time */
);

extern int XGrabServer(
    Display*            /* display */
);

extern int XHeightMMOfScreen(
    Screen*             /* screen */
);

extern int XHeightOfScreen(
    Screen*             /* screen */
);

extern int XIfEvent(
    Display*            /* display */,
    XEvent*             /* event_return */,
    Bool (*) (
             Display*                 /* display */,
           XEvent*                    /* event */,
           XPointer                   /* arg */
         )              /* predicate */,
    XPointer            /* arg */
);

extern int XImageByteOrder(
    Display*            /* display */
);

extern int XInstallColormap(
    Display*            /* display */,
    Colormap            /* colormap */
);

extern KeyCode XKeysymToKeycode(
    Display*            /* display */,
    KeySym              /* keysym */
);

extern int XKillClient(
    Display*            /* display */,
    XID                 /* resource */
);

extern Status XLookupColor(
    Display*            /* display */,
    Colormap            /* colormap */,
    _Xconst char*       /* color_name */,
    XColor*             /* exact_def_return */,
    XColor*             /* screen_def_return */
);
```

```
extern int XLowerWindow(
    Display*            /* display */,
    Window              /* w */
);

extern int XMapRaised(
    Display*            /* display */,
    Window              /* w */
);

extern int XMapSubwindows(
    Display*            /* display */,
    Window              /* w */
);

extern int XMapWindow(
    Display*            /* display */,
    Window              /* w */
);

extern int XMaskEvent(
    Display*            /* display */,
    long        /* event_mask */,
    XEvent*             /* event_return */
);

extern int XMaxCmapsOfScreen(
    Screen*             /* screen */
);

extern int XMinCmapsOfScreen(
    Screen*             /* screen */
);

extern int XMoveResizeWindow(
    Display*            /* display */,
    Window              /* w */,
    int                 /* x */,
    int                 /* y */,
    unsigned int        /* width */,
    unsigned int        /* height */
);

extern int XMoveWindow(
    Display*            /* display */,
    Window              /* w */,
    int                 /* x */,
    int                 /* y */
);
```

```
extern int XNextEvent(
    Display*            /* display */,
    XEvent*             /* event_return */
);

extern int XNoOp(
    Display*            /* display */
);

extern Status XParseColor(
    Display*            /* display */,
    Colormap            /* colormap */,
    _Xconst char*       /* spec */,
    XColor*             /* exact_def_return */
);

extern int XParseGeometry(
    _Xconst char*       /* parsestring */,
    int*           /* x_return */,
    int*           /* y_return */,
    unsigned int*       /* width_return */,
    unsigned int*       /* height_return */
);

extern int XPeekEvent(
    Display*            /* display */,
    XEvent*             /* event_return */
);

extern int XPeekIfEvent(
    Display*            /* display */,
    XEvent*             /* event_return */,
    Bool (*) (
             Display*           /* display */,
           XEvent*              /* event */,
           XPointer             /* arg */
         )              /* predicate */,
    XPointer            /* arg */
);

extern int XPending(
    Display*            /* display */
);

extern int XPlanesOfScreen(
    Screen*             /* screen */
);
```

```
extern int XProtocolRevision(
    Display*                /* display */
);

extern int XProtocolVersion(
    Display*                /* display */
);


extern int XPutBackEvent(
    Display*                /* display */,
    XEvent*                 /* event */
);

extern int XPutImage(
    Display*                /* display */,
    Drawable                /* d */,
    GC                      /* gc */,
    XImage*                 /* image */,
    int                     /* src_x */,
    int                     /* src_y */,
    int                     /* dest_x */,
    int                     /* dest_y */,
    unsigned int            /* width */,
    unsigned int            /* height */
);

extern int XQLength(
    Display*                /* display */
);

extern Status XQueryBestCursor(
    Display*                /* display */,
    Drawable                /* d */,
    unsigned int        /* width */,
    unsigned int            /* height */,
    unsigned int*           /* width_return */,
    unsigned int*           /* height_return */
);

extern Status XQueryBestSize(
    Display*                /* display */,
    int                     /* class */,
    Drawable                /* which_screen */,
    unsigned int            /* width */,
    unsigned int            /* height */,
    unsigned int*           /* width_return */,
    unsigned int*           /* height_return */
);
```

```
extern Status XQueryBestStipple(
    Display*            /* display */,
    Drawable            /* which_screen */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned int*       /* width_return */,
    unsigned int*       /* height_return */
);

extern Status XQueryBestTile(
    Display*            /* display */,
    Drawable            /* which_screen */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    unsigned int*       /* width_return */,
    unsigned int*       /* height_return */
);

extern int XQueryColor(
    Display*            /* display */,
    Colormap            /* colormap */,
    XColor*             /* def_in_out */
);

extern int XQueryColors(
    Display*            /* display */,
    Colormap            /* colormap */,
    XColor*             /* defs_in_out */,
    int                 /* ncolors */
);

extern Bool XQueryExtension(
    Display*            /* display */,
    _Xconst char*       /* name */,
    int*        /* major_opcode_return */,
    int*        /* first_event_return */,
    int*        /* first_error_return */
);

extern int XQueryKeymap(
    Display*            /* display */,
    char [32]           /* keys_return */
);

extern Bool XQueryPointer(
    Display*            /* display */,
    Window              /* w */,
    Window*             /* root_return */,
    Window*                 /* child_return */,
    int*            /* root_x_return */,
    int*            /* root_y_return */,
    int*            /* win_x_return */,
    int*            /* win_y_return */,
    unsigned int*       /* mask_return */
);

extern int XQueryTextExtents(
    Display*                /* display */,
    XID                     /* font_ID */,
    _Xconst char*           /* string */,
    int                     /* nchars */,
    int*            /* direction_return */,
    int*            /* font_ascent_return */,
    int*            /* font_descent_return */,
    XCharStruct*            /* overall_return */
);

extern int XQueryTextExtents16(
    Display*                /* display */,
    XID                     /* font_ID */,
    _Xconst XChar2b*/* string */,
    int                     /* nchars */,
    int*            /* direction_return */,
    int*            /* font_ascent_return */,
    int*            /* font_descent_return */,
    XCharStruct*            /* overall_return */
);

extern Status XQueryTree(
    Display*                /* display */,
    Window                  /* w */,
    Window*                 /* root_return */,
    Window*                 /* parent_return */,
    Window**                /* children_return */,
    unsigned int*           /* nchildren_return */
);

extern int XRaiseWindow(
    Display*                /* display */,
    Window                  /* w */
);

extern int XReadBitmapFile(
    Display*                /* display */,
    Drawable                /* d */,
    _Xconst char*           /* filename */,
    unsigned int*           /* width_return */,
    unsigned int*      /* height_return */,
    Pixmap*            /* bitmap_return */,
    int*        /* x_hot_return */,
    int*        /* y_hot_return */
);

extern int XReadBitmapFileData(
    _Xconst char*      /* filename */,
    unsigned int*      /* width_return */,
    unsigned int*      /* height_return */,
    unsigned char**    /* data_return */,
    int*        /* x_hot_return */,
    int*        /* y_hot_return */
);

extern int XRebindKeysym(
    Display*           /* display */,
    KeySym             /* keysym */,
    KeySym*            /* list */,
    int                /* mod_count */,
    _Xconst unsigned char*    /* string */,
    int                /* bytes_string */
);

extern int XRecolorCursor(
    Display*           /* display */,
    Cursor             /* cursor */,
    XColor*            /* foreground_color */,
    XColor*            /* background_color */
);

extern int XRefreshKeyboardMapping(
    XMappingEvent*     /* event_map */
);

extern int XRemoveFromSaveSet(
    Display*           /* display */,
    Window             /* w */
);

extern int XRemoveHost(
    Display*           /* display */,
    XHostAddress*      /* host */
);

extern int XRemoveHosts(
    Display*           /* display */,
    XHostAddress*      /* hosts */,
    int                /* num_hosts */
```

```
);

extern int XReparentWindow(
    Display*            /* display */,
    Window              /* w */,
    Window              /* parent */,
    int                 /* x */,
    int                 /* y */
);

extern int XResetScreenSaver(
    Display*            /* display */
);

extern int XResizeWindow(
    Display*            /* display */,
    Window              /* w */,
    unsigned int        /* width */,
    unsigned int        /* height */
);

extern int XRestackWindows(
    Display*            /* display */,
    Window*             /* windows */,
    int                 /* nwindows */
);

extern int XRotateBuffers(
    Display*            /* display */,
    int                 /* rotate */
);

extern int XRotateWindowProperties(
    Display*            /* display */,
    Window              /* w */,
    Atom*               /* properties */,
    int                 /* num_prop */,
    int                 /* npositions */
);

extern int XScreenCount(
    Display*            /* display */
);

extern int XSelectInput(
    Display*            /* display */,
    Window              /* w */,
    long        /* event_mask */
);
```

```
extern Status XSendEvent(
    Display*                /* display */,
    Window                  /* w */,
    Bool            /* propagate */,
    long            /* event_mask */,
    XEvent*                 /* event_send */
);

extern int XSetAccessControl(
    Display*                /* display */,
    int                     /* mode */
);

extern int XSetArcMode(
    Display*                /* display */,
    GC                      /* gc */,
    int                     /* arc_mode */
);

extern int XSetBackground(
    Display*                /* display */,
    GC                      /* gc */,
    unsigned long           /* background */
);

extern int XSetClipMask(
    Display*                /* display */,
    GC                      /* gc */,
    Pixmap                  /* pixmap */
);

extern int XSetClipOrigin(
    Display*                /* display */,
    GC                      /* gc */,
    int                     /* clip_x_origin */,
    int                     /* clip_y_origin */
);

extern int XSetClipRectangles(
    Display*                /* display */,
    GC                      /* gc */,
    int                     /* clip_x_origin */,
    int                     /* clip_y_origin */,
    XRectangle*                     /* rectangles */,
    int                     /* n */,
    int                     /* ordering */
);
```

```
extern int XSetCloseDownMode(
   Display*                /* display */,
   int                     /* close_mode */
);

extern int XSetCommand(
   Display*                /* display */,
   Window                  /* w */,
   char**                  /* argv */,
   int                     /* argc */
);

extern int XSetDashes(
   Display*                /* display */,
   GC                      /* gc */,
   int                     /* dash_offset */,
   _Xconst char*           /* dash_list */,
   int                     /* n */
);

extern int XSetFillRule(
   Display*                /* display */,
   GC                      /* gc */,
   int                     /* fill_rule */
);

extern int XSetFillStyle(
   Display*                /* display */,
   GC                      /* gc */,
   int                     /* fill_style */
);

extern int XSetFont(
   Display*                /* display */,
   GC                      /* gc */,
   Font            /* font */
);

extern int XSetFontPath(
   Display*                /* display */,
   char**                  /* directories */,
   int                     /* ndirs */
);

extern int XSetForeground(
   Display*                /* display */,
   GC                      /* gc */,
   unsigned long           /* foreground */
);
```

```
extern int XSetFunction(
    Display*                /* display */,
    GC                      /* gc */,
    int                     /* function */
);

extern int XSetGraphicsExposures(
    Display*                /* display */,
    GC                      /* gc */,
    Bool            /* graphics_exposures */
);

extern int XSetIconName(
    Display*                /* display */,
    Window                  /* w */,
    _Xconst char*           /* icon_name */
);

extern int XSetInputFocus(
    Display*                /* display */,
    Window                  /* focus */,
    int                     /* revert_to */,
    Time                    /* time */
);

extern int XSetLineAttributes(
    Display*                /* display */,
    GC                      /* gc */,
    unsigned int            /* line_width */,
    int                     /* line_style */,
    int                     /* cap_style */,
    int                     /* join_style */
);

extern int XSetModifierMapping(
    Display*                /* display */,
    XModifierKeymap*                /* modmap */
);

extern int XSetPlaneMask(
    Display*                /* display */,
    GC                      /* gc */,
    unsigned long           /* plane_mask */
);

extern int XSetPointerMapping(
    Display*                /* display */,
    _Xconst unsigned char*          /* map */,
    int                     /* nmap */
);

extern int XSetScreenSaver(
    Display*                /* display */,
    int                     /* timeout */,
    int                     /* interval */,
    int                     /* prefer_blanking */,
    int                     /* allow_exposures */
);

extern int XSetSelectionOwner(
    Display*                /* display */,
    Atom                /* selection */,
    Window                  /* owner */,
    Time                    /* time */
);

extern int XSetState(
    Display*                /* display */,
    GC                      /* gc */,
    unsigned long           /* foreground */,
    unsigned long           /* background */,
    int                     /* function */,
    unsigned long           /* plane_mask */
);

extern int XSetStipple(
    Display*                /* display */,
    GC                      /* gc */,
    Pixmap                  /* stipple */
);

extern int XSetSubwindowMode(
    Display*                /* display */,
    GC                      /* gc */,
    int                     /* subwindow_mode */
);

extern int XSetTSOrigin(
    Display*                /* display */,
    GC                      /* gc */,
    int                     /* ts_x_origin */,
    int                     /* ts_y_origin */
);

extern int XSetTile(
    Display*                /* display */,
    GC                      /* gc */,
    Pixmap              /* tile */
);

extern int XSetWindowBackground(
    Display*            /* display */,
    Window              /* w */,
    unsigned long       /* background_pixel */
);

extern int XSetWindowBackgroundPixmap(
    Display*            /* display */,
    Window              /* w */,
    Pixmap              /* background_pixmap */
);

extern int XSetWindowBorder(
    Display*            /* display */,
    Window              /* w */,
    unsigned long       /* border_pixel */
);

extern int XSetWindowBorderPixmap(
    Display*            /* display */,
    Window              /* w */,
    Pixmap              /* border_pixmap */
);

extern int XSetWindowBorderWidth(
    Display*            /* display */,
    Window              /* w */,
    unsigned int        /* width */
);

extern int XSetWindowColormap(
    Display*            /* display */,
    Window              /* w */,
    Colormap            /* colormap */
);

extern int XStoreBuffer(
    Display*            /* display */,
    _Xconst char*       /* bytes */,
    int                 /* nbytes */,
    int                 /* buffer */
);

extern int XStoreBytes(
    Display*            /* display */,
    _Xconst char*       /* bytes */,
    int                     /* nbytes */
);

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

extern int XStoreColors(
    Display*                /* display */,
    Colormap                /* colormap */,
    XColor*                 /* color */,
    int                     /* ncolors */
);

extern int XStoreName(
    Display*                /* display */,
    Window                  /* w */,
    _Xconst char*           /* window_name */
);

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

extern int XSync(
    Display*                /* display */,
    Bool            /* discard */
);

extern int XTextExtents(
    XFontStruct*            /* font_struct */,
    _Xconst char*           /* string */,
    int                     /* nchars */,
    int*            /* direction_return */,
    int*            /* font_ascent_return */,
    int*            /* font_descent_return */,
    XCharStruct*            /* overall_return */
);

extern int XTextExtents16(
    XFontStruct*            /* font_struct */,
    _Xconst XChar2b*/* string */,
    int                     /* nchars */,
    int*          /* direction_return */,
    int*          /* font_ascent_return */,
    int*          /* font_descent_return */,
    XCharStruct*         /* overall_return */
);

extern int XTextWidth(
    XFontStruct*         /* font_struct */,
    _Xconst char*        /* string */,
    int                  /* count */
);

extern int XTextWidth16(
    XFontStruct*         /* font_struct */,
    _Xconst XChar2b*/* string */,
    int                  /* count */
);

extern Bool XTranslateCoordinates(
    Display*             /* display */,
    Window               /* src_w */,
    Window               /* dest_w */,
    int                  /* src_x */,
    int                  /* src_y */,
    int*          /* dest_x_return */,
    int*          /* dest_y_return */,
    Window*              /* child_return */
);

extern int XUndefineCursor(
    Display*             /* display */,
    Window               /* w */
);

extern int XUngrabButton(
    Display*             /* display */,
    unsigned int         /* button */,
    unsigned int         /* modifiers */,
    Window               /* grab_window */
);

extern int XUngrabKey(
    Display*             /* display */,
    int                  /* keycode */,
    unsigned int         /* modifiers */,
    Window               /* grab_window */
);

extern int XUngrabKeyboard(
    Display*            /* display */,
    Time                /* time */
);

extern int XUngrabPointer(
    Display*            /* display */,
    Time                /* time */
);

extern int XUngrabServer(
    Display*            /* display */
);

extern int XUninstallColormap(
    Display*            /* display */,
    Colormap            /* colormap */
);

extern int XUnloadFont(
    Display*            /* display */,
    Font        /* font */
);

extern int XUnmapSubwindows(
    Display*            /* display */,
    Window              /* w */
);

extern int XUnmapWindow(
    Display*            /* display */,
    Window              /* w */
);

extern int XVendorRelease(
    Display*            /* display */
);

extern int XWarpPointer(
    Display*            /* display */,
    Window              /* src_w */,
    Window              /* dest_w */,
    int                 /* src_x */,
    int                 /* src_y */,
    unsigned int        /* src_width */,
    unsigned int        /* src_height */,
    int                 /* dest_x */,
    int                 /* dest_y */
);
```

```
extern int XWidthMMOfScreen(
    Screen*             /* screen */
);

extern int XWidthOfScreen(
    Screen*             /* screen */
);

extern int XWindowEvent(
    Display*            /* display */,
    Window              /* w */,
    long        /* event_mask */,
    XEvent*             /* event_return */
);

extern int XWriteBitmapFile(
    Display*            /* display */,
    _Xconst char*       /* filename */,
    Pixmap              /* bitmap */,
    unsigned int        /* width */,
    unsigned int        /* height */,
    int                 /* x_hot */,
    int                 /* y_hot */
);

extern Bool XSupportsLocale (void);

extern char *XSetLocaleModifiers(
    const char*         /* modifier_list */
);

extern XOM XOpenOM(
    Display*                    /* display */,
    struct _XrmHashBucketRec*           /* rdb */,
    _Xconst char*               /* res_name */,
    _Xconst char*               /* res_class */
);

extern Status XCloseOM(
    XOM                         /* om */
);

extern char *XSetOMValues(
    XOM                         /* om */,
    ...
) _X_SENTINEL(0);

extern char *XGetOMValues(
    XOM                         /* om */,
    ...
) _X_SENTINEL(0);

extern Display *XDisplayOfOM(
    XOM                         /* om */
);

extern char *XLocaleOfOM(
    XOM                         /* om */
);

extern XOC XCreateOC(
    XOM                         /* om */,
    ...
) _X_SENTINEL(0);

extern void XDestroyOC(
    XOC                         /* oc */
);

extern XOM XOMOfOC(
    XOC                         /* oc */
);

extern char *XSetOCValues(
    XOC                         /* oc */,
    ...
) _X_SENTINEL(0);

extern char *XGetOCValues(
    XOC                         /* oc */,
    ...
) _X_SENTINEL(0);

extern XFontSet XCreateFontSet(
    Display*              /* display */,
    _Xconst char*         /* base_font_name_list */,
    char***               /* missing_charset_list */,
    int*            /* missing_charset_count */,
    char**                /* def_string */
);

extern void XFreeFontSet(
    Display*              /* display */,
    XFontSet              /* font_set */
);

extern int XFontsOfFontSet(
    XFontSet              /* font_set */,
    XFontStruct***      /* font_struct_list */,
    char***             /* font_name_list */
);

extern char *XBaseFontNameListOfFontSet(
    XFontSet            /* font_set */
);

extern char *XLocaleOfFontSet(
    XFontSet            /* font_set */
);

extern Bool XContextDependentDrawing(
    XFontSet            /* font_set */
);

extern Bool XDirectionalDependentDrawing(
    XFontSet            /* font_set */
);

extern Bool XContextualDrawing(
    XFontSet            /* font_set */
);

extern XFontSetExtents *XExtentsOfFontSet(
    XFontSet            /* font_set */
);

extern int XmbTextEscapement(
    XFontSet            /* font_set */,
    _Xconst char*       /* text */,
    int                 /* bytes_text */
);

extern int XwcTextEscapement(
    XFontSet            /* font_set */,
    _Xconst wchar_t*    /* text */,
    int                 /* num_wchars */
);

extern int Xutf8TextEscapement(
    XFontSet            /* font_set */,
    _Xconst char*       /* text */,
    int                 /* bytes_text */
);

extern int XmbTextExtents(
    XFontSet            /* font_set */,
    _Xconst char*       /* text */,
    int                  /* bytes_text */,
    XRectangle*                 /* overall_ink_return */,
    XRectangle*                 /* overall_logical_return */
);

extern int XwcTextExtents(
    XFontSet             /* font_set */,
    _Xconst wchar_t*  /* text */,
    int                  /* num_wchars */,
    XRectangle*                 /* overall_ink_return */,
    XRectangle*                 /* overall_logical_return */
);

extern int Xutf8TextExtents(
    XFontSet             /* font_set */,
    _Xconst char*        /* text */,
    int                  /* bytes_text */,
    XRectangle*                 /* overall_ink_return */,
    XRectangle*                 /* overall_logical_return */
);

extern Status XmbTextPerCharExtents(
    XFontSet             /* font_set */,
    _Xconst char*        /* text */,
    int                  /* bytes_text */,
    XRectangle*                 /* ink_extents_buffer */,
    XRectangle*                 /* logical_extents_buffer */,
    int                  /* buffer_size */,
    int*          /* num_chars */,
    XRectangle*                 /* overall_ink_return */,
    XRectangle*                 /* overall_logical_return */
);

extern Status XwcTextPerCharExtents(
    XFontSet             /* font_set */,
    _Xconst wchar_t*  /* text */,
    int                  /* num_wchars */,
    XRectangle*                 /* ink_extents_buffer */,
    XRectangle*                 /* logical_extents_buffer */,
    int                  /* buffer_size */,
    int*          /* num_chars */,
    XRectangle*                 /* overall_ink_return */,
    XRectangle*                 /* overall_logical_return */
);

extern Status Xutf8TextPerCharExtents(
    XFontSet             /* font_set */,
    _Xconst char*        /* text */,
    int                  /* bytes_text */,
    XRectangle*                  /* ink_extents_buffer */,
    XRectangle*                  /* logical_extents_buffer */,
    int                  /* buffer_size */,
    int*          /* num_chars */,
    XRectangle*                  /* overall_ink_return */,
    XRectangle*                  /* overall_logical_return */
);

extern void XmbDrawText(
    Display*             /* display */,
    Drawable             /* d */,
    GC                   /* gc */,
    int                  /* x */,
    int                  /* y */,
    XmbTextItem*         /* text_items */,
    int                  /* nitems */
);

extern void XwcDrawText(
    Display*             /* display */,
    Drawable             /* d */,
    GC                   /* gc */,
    int                  /* x */,
    int                  /* y */,
    XwcTextItem*         /* text_items */,
    int                  /* nitems */
);

extern void Xutf8DrawText(
    Display*             /* display */,
    Drawable             /* d */,
    GC                   /* gc */,
    int                  /* x */,
    int                  /* y */,
    XmbTextItem*         /* text_items */,
    int                  /* nitems */
);

extern void XmbDrawString(
    Display*             /* display */,
    Drawable             /* d */,
    XFontSet             /* font_set */,
    GC                   /* gc */,
    int                  /* x */,
    int                  /* y */,
    _Xconst char*        /* text */,
    int                  /* bytes_text */
);
```

```
extern void XwcDrawString(
    Display*         /* display */,
    Drawable         /* d */,
    XFontSet         /* font_set */,
    GC               /* gc */,
    int              /* x */,
    int              /* y */,
    _Xconst wchar_t* /* text */,
    int              /* num_wchars */
);

extern void Xutf8DrawString(
    Display*         /* display */,
    Drawable         /* d */,
    XFontSet         /* font_set */,
    GC               /* gc */,
    int              /* x */,
    int              /* y */,
    _Xconst char*    /* text */,
    int              /* bytes_text */
);

extern void XmbDrawImageString(
    Display*         /* display */,
    Drawable         /* d */,
    XFontSet         /* font_set */,
    GC               /* gc */,
    int              /* x */,
    int              /* y */,
    _Xconst char*    /* text */,
    int              /* bytes_text */
);

extern void XwcDrawImageString(
    Display*         /* display */,
    Drawable         /* d */,
    XFontSet         /* font_set */,
    GC               /* gc */,
    int              /* x */,
    int              /* y */,
    _Xconst wchar_t* /* text */,
    int              /* num_wchars */
);

extern void Xutf8DrawImageString(
    Display*         /* display */,
    Drawable         /* d */,
    XFontSet         /* font_set */,
    GC               /* gc */,
    int                     /* x */,
    int                     /* y */,
    _Xconst char*           /* text */,
    int                     /* bytes_text */
);

extern XIM XOpenIM(
    Display*                        /* dpy */,
    struct _XrmHashBucketRec*       /* rdb */,
    char*                           /* res_name */,
    char*                           /* res_class */
);

extern Status XCloseIM(
    XIM /* im */
);

extern char *XGetIMValues(
    XIM /* im */, ...
) _X_SENTINEL(0);

extern char *XSetIMValues(
    XIM /* im */, ...
) _X_SENTINEL(0);

extern Display *XDisplayOfIM(
    XIM /* im */
);

extern char *XLocaleOfIM(
    XIM /* im*/
);

extern XIC XCreateIC(
    XIM /* im */, ...
) _X_SENTINEL(0);

extern void XDestroyIC(
    XIC /* ic */
);

extern void XSetICFocus(
    XIC /* ic */
);

extern void XUnsetICFocus(
    XIC /* ic */
);
```

```
extern wchar_t *XwcResetIC(
    XIC /* ic */
);

extern char *XmbResetIC(
    XIC /* ic */
);

extern char *Xutf8ResetIC(
    XIC /* ic */
);

extern char *XSetICValues(
    XIC /* ic */, ...
) _X_SENTINEL(0);

extern char *XGetICValues(
    XIC /* ic */, ...
) _X_SENTINEL(0);

extern XIM XIMOfIC(
    XIC /* ic */
);

extern Bool XFilterEvent(
    XEvent*     /* event */,
    Window      /* window */
);

extern int XmbLookupString(
    XIC                     /* ic */,
    XKeyPressedEvent*           /* event */,
    char*                   /* buffer_return */,
    int                     /* bytes_buffer */,
    KeySym*                 /* keysym_return */,
    Status*                 /* status_return */
);

extern int XwcLookupString(
    XIC                     /* ic */,
    XKeyPressedEvent*           /* event */,
    wchar_t*                /* buffer_return */,
    int                     /* wchars_buffer */,
    KeySym*                 /* keysym_return */,
    Status*                 /* status_return */
);

extern int Xutf8LookupString(
    XIC                     /* ic */,
    XKeyPressedEvent*        /* event */,
    char*                /* buffer_return */,
    int                  /* bytes_buffer */,
    KeySym*              /* keysym_return */,
    Status*              /* status_return */
);

extern XVaNestedList XVaCreateNestedList(
    int /*unused*/, ...
) _X_SENTINEL(0);

/* internal connections for IMs */

extern Bool XRegisterIMInstantiateCallback(
    Display*                 /* dpy */,
    struct _XrmHashBucketRec*        /* rdb */,
    char*                    /* res_name */,
    char*                    /* res_class */,
    XIDProc                  /* callback */,
    XPointer                 /* client_data */
);

extern Bool XUnregisterIMInstantiateCallback(
    Display*                 /* dpy */,
    struct _XrmHashBucketRec*        /* rdb */,
    char*                    /* res_name */,
    char*                    /* res_class */,
    XIDProc                  /* callback */,
    XPointer                 /* client_data */
);

typedef void (*XConnectionWatchProc)(
    Display*                 /* dpy */,
    XPointer                 /* client_data */,
    int                      /* fd */,
    Bool                 /* opening */,  /* open or close flag */
    XPointer*                /* watch_data */ /* open sets, close uses */
);


extern Status XInternalConnectionNumbers(
    Display*                 /* dpy */,
    int**                    /* fd_return */,
    int*                 /* count_return */
);

extern void XProcessInternalConnection(
    Display*                 /* dpy */,
    int                      /* fd */
```

```
);

extern Status XAddConnectionWatch(
    Display*                  /* dpy */,
    XConnectionWatchProc      /* callback */,
    XPointer                  /* client_data */
);

extern void XRemoveConnectionWatch(
    Display*                  /* dpy */,
    XConnectionWatchProc      /* callback */,
    XPointer                  /* client_data */
);

extern void XSetAuthorization(
    char *                    /* name */,
    int                       /* namelen */,
    char *                    /* data */,
    int                       /* datalen */
);

extern int _Xmbtowc(
    wchar_t *                 /* wstr */,
    char *                    /* str */,
    int                       /* len */
);

extern int _Xwctomb(
    char *                    /* str */,
    wchar_t                   /* wc */
);

extern Bool XGetEventData(
    Display*                  /* dpy */,
    XGenericEventCookie*      /* cookie*/
);

extern void XFreeEventData(
    Display*                  /* dpy */,
    XGenericEventCookie*      /* cookie*/
);

#ifdef __clang__
#pragma clang diagnostic pop
#endif

_XFUNCPROTOEND

#endif /* _X11_XLIB_H_ */
```