HP 9000 Series 300/800 Computers

HEWLETT PACKARD

X Window System C
Quick Reference Guide

X Window System C Quick Reference Guide

HP 9000 Series 300/800 Computers

HP Part Number 98794-90003

Hewlett-Packard Company
1000 NE Circle Blvd., Corvallis OR 97330

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Certification of conformance with the OSF/Motif user environment is pending.

OSF, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

December 1988 ... Edition 1
September 1989 ... Edition 2

Introduction

The *X Window System C Quick Reference Guide* provides a list of the C functions and widgets that are available to the users of the X Window System through the Xt Intrinsics, Xlib, and the HP Motif widgets.

Xlib is a C subroutine library that application programs (clients) use to interface with the window system. The HP Motif Widgets provide a base of predefined features that the programmer can use to build an application. A widget programmer can design new widgets by combining existing widgets or by using the Xt Intrinsics.

For detailed information about each function or widget, refer to *Programming with the Xt Intrinsics, Programming with Xlib, HP Motif Programmer's Guide,* and *HP Motif Programmer's Reference.*

This quick reference guide is organized as follows:

Chapter 1	Intrinsics functions alphabetized by name. Each entry contains the function call and parameter types.
Chapter 2	Xlib functions alphabetized by name. Each entry contains the function call and parameter types.
Chapter 3	Resource sets and convenience functions for widgets and gadgets listed alphabetically.

Contents

1

Intrinsics Functions

The items in this table are in alphabetical order, disregarding the type, such as void or int.

void **MenuPopdown**(*shell_name*)
String *shell_name;*

void **MenuPopup**(*shell_name*)
String *shell_name;*

void **XtAddActions**(*actions, num_actions*)
XtActionList *actions;*
Cardinal *num_actions;*

void **XtAddCallback**(*w, callback_name, callback, client_data*)
Widget *w;*
String *callback_name;*
XtCallbackProc *callback;*
caddr_t *client_data;*

void **XtAddCallbacks**(*w, callback_name, callbacks*)
Widget *w;*
String *callback_name;*
XtCallbackList *callbacks;*

void **XtAddConverter**(*from_type, to_type, converter, convert_args, num_args*)
String *from_type, to_type;*
XtConverter *converter;*
XtConvertArgList *convert_args;*
Cardinal *num_args;*

void **XtAddEventHandler**(*w, event_mask, nonmaskable, proc, client_data*)
Widget *w;*
EventMask *event_mask;*
Boolean *nonmaskable;*
XtEventHandler *proc;*
caddr_t *client_data;*

void **XtAddExposureToRegion**(*event, region*)
XEvent **event;*
Region *region;*

void **XtAddGrab**(*w, exclusive, spring_loaded*)
Widget *w;*
Boolean *exlusive, spring_loaded;*

XtInputId **XtAddInput**(*source, condition, proc, client_data*)
int *source;*
caddr_t *condition, client_data;*
XtInputCallbackProc *proc;*

void **XtAddRawEventHandler**(*w, event_mask, nonmaskable, proc, client_data*)
Widget *w;*
EventMask *event_mask;*
Boolean *nonmaskable;*
XtEventHandler *proc;*
caddr_t *client_data;*

XtIntervalId **XtAddTimeOut**(*interval, proc, client_data*)
unsigned long *interval;*
XtTimerCallbackProc *proc;*
caddr_t *client_data;*

XtWorkProcId **XtAddWorkProc**(*proc, closure*)
XtWorkProc *proc;*
Opaque *closure;*

void **XtAppAddActions**(*app_context, actions, num_actions*)
XtAppContext *app_context;*
XtActionList *actions;*
Cardinal *num_actions;*

void **XtAppAddConverter**(*app_context, from_type, to_type, converter, convert_args, num_args*)
XtAppContext *app_context;*
String *from_type, to_type;*
XtConverter *converter;*
XtConvertArgList *convert_args;*
Cardinal *num_args;*

XtInputId **XtAppAddInput**(*app_context, source, condition, proc, client_data*)
XtAppContext *app_context;*
int *source;*
caddr_t *condition, client_data;*
XtInputCallbackProc *proc;*

XtIntervalId **XtAppAddTimeOut**(*app_context, interval, proc, client_data*)
XtAppContext *app_context;*
unsigned long *interval;*
XtTimerCallbackProc *proc;*
caddr_t *client_data;*

XtWorkProcId **XtAppAddWorkProc**(*app_context, proc, client_data*)
XtAppContext *app_context;*
XtWorkProc *proc;*
caddr_t *client_data;*

Widget **XtAppCreateShell**(*application_name, application_class, widget_class, display, args, num_args*)
String *application_name, application_class;*
WidgetClass *widget_class;*
Display **display;*
Arglist *args;*
Cardinal *num_args;*

void **XtAppError**(*app_context, message*)
XtAppContext *app_context;*
String *message;*

void **XtAppErrorMsg**(*app_context, name, type, class, default, params, num_params*)
XtAppContext *app_context;*
String *name, type, class, default, *params;*
Cardinal **num_params;*

XrmDatabase **XtAppGetErrorDatabase**(*app_context*)
XtAppContext *app_context;*

void **XtAppGetErrorDatabaseText**(*app_context, name, type, class, default, buffer_return, nbytes, database*)
XtAppContext *app_context;*
char **name, *type, *class, *default, *buffer_return;*
int nbytes*;*
XrmDatabase database*;*

unsigned long **XtAppGetSelectionTimeout**(*app_context*)
XtAppContext *app_context;*

void **XtAppMainLoop**(*app_context*)
XtAppContext *app_context;*

void **XtAppNextEvent**(*app_context, event_return*)
XtAppContext *app_context;*
XEvent **event_return;*

Boolean **XtAppPeekEvent**(*app_context, event_return*)
XtAppContext *app_context;*
XEvent **event_return;*

Boolean **XtAppPending**(*app_context*)
XtAppContext *app_context;*

void **XtAppProcessEvent**(*app_context, mask*)
XtAppContext *app_contest;*
XtInputMask *mask;*

void **XtAppSetErrorHandler**(*app_context, handler*)
XtAppContext *app_context;*
XtErrorHandler *handler;*

void **XtAppSetErrorMsgHandler**(*app_context, msg_handler*)
XtAppContext *app_context;*
XtErrorMsgHandler *msg_handler;*

void **XtAppSetSelectionTimeout**(*app_context, timeout*)
XtAppContext *app_context;*
unsigned long *timeout;*

void **XtAppSetWarningHandler**(*app_context, handler*)
XtAppContext *app_context;*
XtErrorHandler *handler;*

void **XtAppSetWarningMsgHandler**(*app_context, msg_handler*)
XtAppContext *app_context;*
XtErrorMsgHandler *msg_handler;*

void **XtAppWarning**(*app_context, message*)
XtAppContext *app_context;*
String *message;*

void **XtAppWarningMsg**(*app_context, name, type, class, default, params, num_params*)
XtAppContext *app_context;*
String *name, type, class, default, *params;*
Cardinal **num_params;*

void **XtAugmentTranslations**(*w, translations*)
Widget *w;*
XtTranslations *translations;*

void **XtBuildEventMask**(*w*)
Widget *w;*

Boolean **XtCallAcceptFocus**(*w, time*)
Widget *w;*
Time **time;*

void **XtCallbackExclusive**(*w, client_data, call_data*)
Widget *w;*
caddr_t *client_data, call_data;*

void **XtCallbackNone**(*w, client_data, call_data*)
Widget *w;*
caddr_t *client_data, call_data;*

void **XtCallbackNonexclusive**(*w, client_data, call_data*)
Widget *w;*
caddr_t *client_data, call_data;*

void **XtCallbackPopdown**(*w, client_data, call_data*)
Widget *w;*
caddr_t *client_data, call_data;*

void **XtCallCallbacks**(*w, callback_name, call_data*)
Widget *w;*
String *callback_name;*
caddr_t *call_data;*

char ***XtCalloc**(*num, size*)
Cardinal *num, size;*

void **XtCheckSubclass**(*w, widget_class, message*)
Widget *w;*
WidgetClass *widget_class;*
String *message;*

WidgetClass **XtClass**(*w*)
Widget *w;*

void **XtCloseDisplay**(*display*)
Display **display;*

void **XtConfigureWidget**(*w, x, y, width, height, border_width*)
Widget *w;*
Position *x, y;*
Dimension *width, height, border_width;*

void **XtConvert**(*w, from_type, from, to_type, to_return*)
Widget *w;*
String *from_type, to_type;*
XrmValuePtr *from, to_return;*

void **XtConvertCase**(*display, keysym, lower_return, upper_return*)
Display **display;*
KeySym *keysym, *lower_return, *upper_return;*

XtAppContext **XtCreateApplicationContext**()

Widget **XtCreateApplicationShell**(*name, widget_class, args, num_args*)
String *name;*
WidgetClass *widget_class;*
Arglist *args;*
Cardinal *num_args;*

Widget **XtCreateManagedWidget**(*name, widget_class, parent, args, num_args*)
String *name;*
WidgetClass *widget_class;*
Widget *parent;*
Arglist *args;*
Cardinal *num_args;*

Widget **XtCreatePopupShell**(*name, widget_class, parent, args, num_args*)
String *name;*
WidgetClass *widget_class;*
Widget *parent;*
Arglist *args;*
Cardinal *num_args;*

Widget **XtCreateWidget**(*name, widget_class, parent, args, num_args*)
String *name;*
WidgetClass *widget_class;*
Widget *parent;*
Arglist *args;*
Cardinal *num_args;*

void **XtCreateWindow**(*w, window_class, visual, value_mask, attributes*)
Widget *w;*
unsigned int *window_class;*
Visual **visual;*
XtValueMask *value_mask;*
XSetWindowAttributes **attributes;*

XrmDatabase **XtDatabase**(*display*)
Display **display;*

void **XtDestroyApplicationContext**(*app_context*)
XtAppContext *app_context;*

void **XtDestroyGC**(*gc*)
GC *gc;*

void **XtDestroyWidget**(*w*)
Widget *w;*

void **XtDirectConvert**(*converter, args, num_args, from, to_return*)
XtConverter *converter;*
XrmValuePtr *args, from, to_return;*
Cardinal *num_args;*

void **XtDisownSelection**(*w, selection, time*)
Widget *w;*
Atom *selection;*
Time *time;*

void **XtDispatchEvent**(*event*)
XEvent **event;*

Display ***XtDisplay**(*w*)
Widget *w;*

void **XtDisplayInitialize**(*app_context, display, application_name, application_class, options, num_options, argc, argv*)
XtAppContext *app_context;*
Display **display;*
String *application_name, application_class;*
XrmOptionDescRec *options[ ];*
Cardinal *num_options, *argc;*
String *argv[ ];*

void **XtError**(*message*)
String *message;*

void **XtErrorMsg**(*name, type, class, default, params, num_params*)
String *name, type, class, default, *params;*
Cardinal **num_params;*

void **XtFree**(*ptr*)
char **ptr;*

void **XtGetApplicationResources**(*w, base, resources, num_resources, args, num_args*)
Widget *w;*
caddr_t *base;*
XtResourceList *resources;*
Cardinal *num_resources, num_args;*
ArgList *args;*

XrmDatabase **XtGetErrorDatabase**()

void **XtGetErrorDatabaseText**(*name, type, class, default, buffer_return, nbytes*)
char **name, **type, **class, **default, **buffer_return;*
int *nbytes;*

GC **XtGetGC**(*w, value_mask, values*)
Widget *w;*
XtGCMask *value_mask;*
XGCValues **values;*

void **XtGetResourceList**(*class, resources_return, num_resources_return*)
WidgetClass *class;*
XtResourceList **resources_return;*
Cardinal **num_resources_return;*

unsigned long **XtGetSelectionTimeout**()

void **XtGetSelectionValue**(*w, selection, target, callback, client_data, time*);
Widget *w;*
Atom *selection, target;*
XtSelectionCallbackProc *callback;*
caddr_t *client_data;*
Time *time;*

void **XtGetSelectionValues**(*w, selection, targets, count, callback, client_data, time*)
Widget *w;*
Atom *selection, **targets;*
int *count;*
XtSelectionCallbackProc *callback;*
caddr_t *client_data;*
Time *time;*

void **XtGetSubresources**(*w, base, name, class, resources, num_resources, args, num_args*)
Widget *w;*
caddr_t *base;*
String *name, class;*
XtResourceList *resources;*
Cardinal *num_resources, num_args;*
ArgList *args;*

void **XtGetSubvalues**(*base, resources, num_resources, args, num_args*)
caddr_t *base;*
XtResourceList *resource;*
Cardinal *num_resources, num_args;*
ArgList *args;*

void **XtGetValues**(*w, args, num_args*)
Widget *w;*
ArgList *args;*
Cardinal *num_args;*

XtCallbackStatus **XtHasCallbacks**(*w, callback_name*)
Widget *w;*
String *callback_name;*

Widget **XtInitialize**(*shell_name, application_class, options, num_options, argc, argv*)
String *shell_name, application_class;*
XrmOptionDescRec *options[ ];*
Cardinal *num_options, *argc;*
String *argv[ ];*

void **XtInstallAccelerators**(*destination, source*)
Widget *destination, source;*

void **XtInstallAllAccelerators**(*destination, source*)
Widget *destination, source;*

Boolean **XtIsComposite**(*w*)
Widget *w;*

Boolean **XtIsManaged**(*w*)
Widget *w;*

Boolean **XtIsRealized**(*w*)
Widget *w;*

Boolean **XtIsSensitive**(*w*)
Widget *w;*

Boolean **XtIsSubclass**(*w, widget_class*)
Widget *w;*
WidgetClass *widget_class;*

XtGeometryResult **XtMakeGeometryRequest**(*w, request, reply_return*)
Widget *w;*
XtWidgetGeometry **request, *reply_return;*

XtGeometryResult **XtMakeResizeRequest**(*w, width, height, width_return, height_return*)
Widget *w;*
Dimension *width, height;*
Dimension **width_return, *height_return;*

char ***XtMalloc**(*size*)
Cardinal *size;*

void **XtManageChild**(*child*)
Widget *child;*

void **XtManageChildren**(*children, num_children*)
WidgetList *children;*
Cardinal *num_children;*

XtMapWidget(*w*)
Widget *w;*

ArgList **XtMergeArgLists**(*arg1, num_args1, arg2, num_args2*)
ArgList *arg1, arg2;*
Cardinal *num_args1, num_args2;*

void **XtMoveWidget**(*w, x, y*)
Widget *w;*
Position *x, y;*

Widget **XtNameToWidget**(*reference, names*)
Widget *reference;*
String *names;*

type ***XtNew**(*type*)
type;

String **XtNewString**(*string*)
String *string;*

void **XtNextEvent**(*event_return*);
XEvent **event_return;*

Cardinal **XtNumber**(*array*)
ArrayVariable *array;*

Cardinal **XtOffset**(*pointer_type, field_name*)
Type *pointer_type;*
Field *field_name;*

Display ***XtOpenDisplay**(*app_context, display_name, application_name, application_class, options, num_options, argc, argv*)
XtAppContext *app_context;*
String *display_name, application_name, application_class;*
XrmOptionDescRec **options;*
Cardinal *num_options, *argc;*
String **argv;*

void **XtOverrideTranslations**(*w, translations*)
Widget *w;*
XtTranslations *translations;*

Boolean **XtOwnSelection**(*w, selection, time, convert_proc, lose_selection, done_proc*)
Widget *w;*
Atom *selection;*
Time *time;*
XtConvertSelectionProc *convert_proc;*
XtLoseSelectionProc *lose_selection;*
XtSelectionDoneProc *done_proc;*

Widget **XtParent**(*w*)
Widget *w;*

XtAccelerators **XtParseAcceleratorTable**(*source*)
String *source;*

XtTranslations **XtParseTranslationTable**(*table*)
String *table;*

Boolean **XtPeekEvent**(*event_return*)
XEvent *event_return;*

Boolean **XtPending**()

void **XtPopdown**(*popup_shell*)
Widget *popup_shell;*

void **XtPopup**(*popup_shell, grab_kind*)
Widget *popup_shell;*
XtGrabKind *grab_kind;*

void **XtProcessEvent**(*mask*)
XtInputMask *mask;*

XtGeometryResult **XtQueryGeometry**(*w, intended, preferred_return*)
Widget *w;*
XtWidgetGeometry **intended, *preferred_return;*

void **XtRealizeWidget**(*w*)
Widget *w;*

char ***XtRealloc**(*ptr, num*)
char **ptr;*
Cardinal *num;*

void **XtRegisterCaseConverter**(*display, proc, start, stop*)
Display **display;*
XtCaseProc *proc;*
KeySym *start, stop;*

void **XtReleaseGC**(*w, gc*)
Widget *w;*
GC *gc;*

void **XtRemoveAllCallbacks**(*w, callback_name*)
Widget *w;*
String *callback_name;*

void **XtRemoveCallback**(*w, callback_name, callback, client_data*)
Widget *w;*
String *callback_name;*
XtCallbackProc *callback;*
caddr_t *client_data;*

void **XtRemoveCallbacks**(*w, callback_name, callback*)
Widget *w;*
String *callback_name;*
XtCallbackList *callbacks;*

void **XtRemoveEventHandler**(*w, event_mask, nonmaskable, proc, client_data*)
Widget *w;*
EventMask *event_mask;*
Boolean *nonmaskable;*
XtEventHandler *proc;*
caddr_t *client_data;*

void **XtRemoveGrab**(*w*)
Widget *w;*

void **XtRemoveInput**(*id*)
XtInputID *id;*

void **XtRemoveRawEventhandler**(*w, event_mask, nonmaskable, proc, client_data*)
Widget *w;*
EventMask *event_mask;*
Boolean *nonmaskable;*
XtEventHandler *proc;*
caddr_t *client_data;*

void **XtRemoveTimeOut**(*timer*)
XtIntervalId *timer;*

void **XtRemoveWorkProc**(*id*)
XtWorkProcId *id;*

void **XtResizeWidget**(*w, width, height, border_width*)
Widget *w;*
Dimension *width, height, border_width;*

void **XtResizeWindow**(*w*)
Widget *w;*

Screen ***XtScreen**(*w*)
Widget *w;*

XtSetArg(*arg, name, value*)
Arg *arg;*
String *name;*
XtArgVal *value;*

void **XtSetErrorHandler**(*handler*)
XtErrorHandler *handler;*

void **XtSetErrorMsgHandler**(*msg_handler*)
XtErrorMsgHandler *msg_handler;*

XtSetKeyboardFocus(*subtree, descendent*)
Widget *subtree, descendent;*

void **XtSetKeyTranslator**(*display, proc*)
Display **display;*
XtKeyProc *proc;*

void **XtSetMappedWhenManaged**(*w, map_when_managed*)
Widget *w;*
Boolean *map_when_managed;*

void **XtSetSelectionTimeout**(*timeout*);
unsigned long *timeout;*

void **XtSetSensitive**(*w, sensitive*)
Widget *w;*
Boolean *sensitive;*

void **XtSetSubvalues**(*base, resources, num_resources, args, num_args*)
caddr_t *base;*
XtResourceList *resources;*
Cardinal *num_resources, num_args;*
ArgList *args;*

void **XtSetValues**(*w, args, num_args*)
Widget *w;*
ArgList *args;*
Cardinal *num_args;*

void **XtSetWarningMsgHandler**(*msg_handler*)
XtErrorMsgHandler *msg_handler;*

void **XtStringConversionWarning**(*src, dst_type*)
String *src, dst_type;*

WidgetClass **XtSuperclass**(*w*)
Widget *w;*

void **XtTranslateCoords**(*w, x, y, rootx_return, rooty_return*)
Widget *w;*
Position *x, y, *rootx_return, *rooty_return;*

void **XtTranslateKeycode**(*display, keycode, modifiers, modifier_return, keysym_name*)
Display **display;*
KeyCode *keycode;*
Modifiers *modifiers, *modifier_return;*
KeySym **keysym_name;*

void **XtUninstallTranslations**(*w*)
Widget *w;*

void **XtUnmanageChild**(*child*)
Widget *child;*

void **XtUnmanageChildren**(*children, num_children*)
WidgetList *children;*
Cardinal *num_children;*

XtUnmapWidget(*w*)
Widget *w;*

void **XtUnrealizeWidget**(*w*)
Widget *w;*

void **XtWarning**(*message*)
String *message;*

void **XtWarningMsg**(*name, type, class, default, params, num_params*)
String *name, type, class, default, *params;*
Cardinal *num_params;*

XtAppContext **XtWidgetToApplicationContext**(*w*)
Widget *w;*

Window **XtWindow**(*w*)
Widget *w;*

Widget **XtWindowToWidget**(*display, window*)
Display *display;*
Window *window;*

2

Xlib Functions

The items in this table are in alphabetical order, disregarding the type, such as void or int.

FlushGC(*display, gc*)
Display **display;*
GC *gc;*

XActivateScreenSaver(*display*)
Display **display;*

XAddHost(*display, host*)
Display **display;*
XHostAddress **host;*

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

XAddPixel(*ximage, value*)
XImage **ximage;*
unsigned long *value;*

XAddToExtensionList(*structure, ext_data*)
struct _XExtData ***structure;*
XExtData **ext_data;*

XAddToSaveSet(*display, w*)
Display **display;*
Window *w;*

Status **XAllocColor**(*display, cmap, screen_in_out*)
Display **display;*
Colormap *cmap;*
XColor **screen_in_out;*

Status **XAllocColorCells**(*display, cmap, contig, plane_masks_return, nplanes, pixels_return, ncolors*)
    Display **display;*
    Colormap *cmap;*
    Bool *contig;*
    unsigned long *plane_masks_return[ ], pixels_return[ ];*
    unsigned int *nplanes, ncolors;*

Status **XAllocColorPlanes**(*display, cmap, contig, pixels_return, ncolors, nreds, ngreens, nblues, rmask_return, gmask_return, bmask_return*)
    Display **display;*
    Colormap *cmap;*
    Bool *contig;*
    unsigned long *pixels_return[ ];*
    int *ncolors, nreds, ngreens, nblues;*
    unsigned long **rmask_return, *gmask_return, *bmask_return;*

Status **XAllocNamedColor**(*display, cmap, color_name, screen_def_return, exact_def_return*)
    Display **display;*
    Colormap *cmap;*
    char **color_name;*
    XColor **screen_def_return, *exact_def_return;*

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

unsigned long **XAllPlanes**( )

XAutoRepeatOff(*display*)
    Display **display;*

XAutoRepeatOn(*display*)
    Display **display;*

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

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

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

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

unsigned long **XBlackPixel**(*display, screen_number*)
    Display **display;*
    int *screen_number;*

unsigned long **XBlackPixelOfScreen**(*screen*)
    Screen **screen;*

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

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

XChangeGC(*display, gc, valuemask_change, values*)
   Display **display;*
   GC *gc;*
   unsigned long *valuemask_change;*
   XGCValues **values;*

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

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

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

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

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

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

Bool XCheckIfEvent(*display, event_return, predicate, arg*)
    Display **display;*
    XEvent **event_return;*
    Bool "(**predicate*)();
    char **arg;*

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

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

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

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

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

XCirculateSubwindowsDown(*display, w*)
    Display **display;*
    Window *w;*

XCirculateSubwindowsUp(*display, w*)
    Display **display;*
    Window *w;*

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

XClearWindow(*display, w*)
    Display **display;*
    Window *w;*

XClipBox(*r, rect_return*)
    Region *r;*
    XRectangle **rect_return;*

XCloseDisplay(*display*)
    Display **display;*

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

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

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

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

Colormap **XCopyColormapAndFree**(*display, cmap*)
Display **display;*
Colormap *cmap;*

XCopyGC(*display, src, valuemask_copy, dest*)
Display **display;*
GC *src, dest;*
unsigned long *valuemask_copy;*

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

XAssocTable *XCreateAssocTable(*size*)
int *size;*

Pixmap **XCreateBitmapFromData**(*display, d, data, width, height*)
    Display **display;*
    Drawable *d;*
    char **data;*
    unsigned int *width, height;*

Colormap **XCreateColormap**(*display, w, visual, alloc*)
    Display **display;*
    Window *w;*
    Visual **visual;*
    int *alloc;*

Cursor **XCreateFontCursor**(*display, shape*)
    Display **display;*
    unsigned int *shape;*

GC **XCreateGC**(*display, d, valuemask_create, values*)
    Display **display;*
    Drawable *d;*
    unsigned long *valuemask_create;*
    XGCValues **values;*

Cursor **XCreateGlyphCursor**(*display, source_font, mask_font, source_char, mask_char, foreground_color, background_color*)
    Display **display;*
    Font *source_font, mask_font;*
    unsigned int *source_char, mask_char;*
    XColor **foreground_color, *background_color;*

XImage ***XCreateImage**(*display, visual, depth, format, offset, data, width, height, bitmap_pat, bytes_per_line*)
    Display **display;*
    Visual **visual;*
    unsigned int *depth, width, height;*
    int *format, offset, bitmap_pad, bytes_per_line;*
    char **data;*

Pixmap **XCreatePixmap**(*display, d, width, height, depth*)
    Display **display;*
    Drawable *d;*
    unsigned int *width, height, depth;*

Cursor **XCreatePixmapCursor**(*display, source, mask, foreground_color, background_color*)
Display **display;*
Pixmap *source, mask;*
XColor **foreground_color, background_color;*
unsigned int *x, y;*

Pixmap **XCreatePixmapFromBitmapData**(*display, d, data, width, height, fg, bg, depth*)
Display **display;*
Drawable *d;*
char **data;*
unsigned int *width, height, depth;*
unsigned long *fg, bg;*

Region **XCreateRegion**()

Window **XCreateSimpleWindow**(*display, parent, x, y, width, height, border_width, border, background*)
Display **display;*
Window *parent;*
int *x, y;*
unsigned int *width, height, border_width;*
unsigned long *border, background;*

Window **XCreateWindow**(*display, parent, x, y, width, height, border_width, depth, class, visual, valuemask, attributes*)
Display **display;*
Window *parent;*
int *x, y, depth;*
unsigned int *width, height, border_width, class;*
Visual *visual;
unsigned long *valuemask;*
XSetWindowAttributes **attributes;*

Colormap **XDefaultColormap**(*display, screen_number*)
Display **display;*
int *screen_number;*

Colormap **XDefaultColormapOfScreen**(*screen*)
Screen **screen;*

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

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

GC **XDefaultGC**(*display, screen_number*)
    Display **display;*
    int *screen_number;*

GC **XDefaultGCOfScreen**(*screen*)
    Screen **screen;*

Window **XDefaultRootWindow**(*display*)
    Display **display;*

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

Screen ***XDefaultScreenOfDisplay**(*display*)
    Display **display;*

Visual ***XDefaultVisual**(*display, screen_number*)
    Display **display;*
    int *screen_number;*

Visual ***XDefaultVisualOfScreen**(*screen*)
    Screen **screen;*

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

XDeleteAssoc(*display, table, x_id*)
    Display **display;*
    XAssocTable **table;*
    XID *x_id;*

int **XDeleteContext**(*display, w, context*)
    Display **display;*
    Window *w;*
    XContext *context;*

XModifierKeymap ***XDeleteModifiermapEntry**(*modmap, keycode_entry, modifier*)
    XModifierKeymap **modmap;*
    KeyCode *keycode_entry;*
    int *modifier;*

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

XDestroyAssocTable(*table*)
    XAssocTable **table;*

int **XDestroyImage**(*ximage*)
    XImage **ximage;*

XDestroyRegion(*r*)
    Region *r;*

XDestroySubwindows(*display, w*)
    Display **display;*
    Window *w;*

XDestroyWindow(*display, w*)
    Display **display;*
    Window *w;*

XDisableAccessControl(*display*)
    Display **display;*

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

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

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

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

Display *XDisplayOfScreen(*screen*)
    Screen **screen;*

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

char *XDisplayString(*display*)
    Display **display;*

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

int XDisplayWidthMM(*display, screen_number*)
    Display **display;*
    int screen_number;

int XDoesBackingStore(*screen*)
    Screen **screen;*

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

Status XDraw(*display, d, gc, vlist, vcount*)
    Display **display;*
    Drawable *d;*
    GC *gc;*
    Vertex **vlist;*
    int *vcount;*

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

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

Status **XDrawFilled**(*display, d, gc, vlist, vcount*)
    Display **display;*
    Drawable *d;*
    GC *gc;*
    Vertex **vlist;*
    int *vcount;*

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

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

XDrawLine(*display, d, gc, x1, y1, x2, y2*)
    Display **display;*
    Drawable *d;*
    GC *gc;*
    int *x1, y1, x2, y2;*

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

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

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

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

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

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

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

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

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

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

Bool **XEmptyRegion**(*r*)
Region *r;*

XEnableAccessControl(*display*)
Display **display;*

Bool **XEqualRegion**(*r1, r2*)
Region *r1, r2;*

long **XEventMaskOfScreen**(*screen*)
Screen **screen;*

int **XEventsQueued**(*display, mode*)
Display **display;*
int *mode;*

char ***XFetchBuffer**(*display, nbytes_return, buffer*)
Display **display;*
int **nbytes_return, buffer;*

char ***XFetchBytes**(*display, nbytes_return*)
Display **display;*
int **nbytes_return;*

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

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

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

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

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

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

int **XFindContext**(*display, w, context, data_return*)
Display **display;*
Window *w;*
XContext *context;*
caddr_t **data_return;*

XExtData ***XFindOnExtensionList**(*structure, number*)
    struct _XExtData ***structure;*
    int *number;*

XFlush(*display*)
    Display **display;*

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

XFree(*data*)
    char **data;*

XFreeColormap(*display, cmap*)
    Display **display;*
    Colormap *cmap;*

XFreeColors(*display, cmap, pixels, npixels, planes*)
    Display **display;*
    Colormap *cmap;*
    unsigned long *pixels[], planes;*
    int *npixels;*

XFreeCursor(*display, cursor*)
    Display **display;*
    Cursor *cursor;*

XFreeFont(*display, font_struct*)
    Display **display;*
    XFontStruct **font_struct;*

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

XFreeFontNames(*list*)
    char **list[];*

XFreeFontPath(*list*)
    char ***list;*

XFreeGC(*display, gc*)
Display **display;*
GC *gc;*

XFreeModifiermap(*modmap*)
XModifierKeymap **modmap;*

XFreePixmap(*display, pixmap*)
Display **display;*
Pixmap *pixmap;*

GContext **XGContextFromGC**(*gc*)
GC *gc;*

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

char ***XGetAtomName**(*display, atom*)
Display **display;*
Atom *atom;*

Status **XGetClassHint**(*display, w, class_hints_return*)
Display **display;*
Window *w;*
XClassHint **class_hints_return;*

char ***XGetDefault**(*display, program, options*)
Display **display;*
char **program, *options;*

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

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

char ***XGetFontPath**(*display, npaths_return*)
Display **display;*
int **npaths_return;*

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

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

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

Status **XGetIconSizes**(*display, w, size_list_return, count_return*)
Display **display;*
Window *w;*
XIconSize ***size_list_return;*
int **count_return;*

XImage ***XGetImage**(*display, d, x, y, width, height, plane_mask, format*)
Display **display;*
Drawable *d;*
int *x, y, format;*
unsigned int *width, height;*
unsigned long *plane_mask;*

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

XGetKeyboardControl(*display, values_return*)
Display **display;*
XKeyboardState **values_return;*

KeySym ***XGetKeyboardMapping**(*display, first_keycode_wanted, keycode_count, keysyms_per_keycode_return*)
Display **display;*
KeyCode *first_keycode_wanted;*
int *keycode_count, *keysyms_per_keycode_return;*

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

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

Status **XGetNormalHints**(*display, w, hints_return*)
Display **display;*
Window *w;*
XSizeHints **hints_return;*

unsigned long **XGetPixel**(*ximage, x, y*)
XImage **ximage;*
int *x, y;*

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

int **XGetPointerMapping**(*display, map_return, nmap*)
Display **display;*
unsigned char *map_return[ ];*
int *nmap;*

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

Window XGetSelectionOwner(*display, selection*)
Display **display;*
Atom *selection;*

Status XGetSizeHints(*display, w, hints_return, property*)
Display **display;*
Window *w;*
XSizeHints **hints_return;*
Atom *property;*

Status XGetStandardColormap(*display, w, cmap_return, property*)
Display **display;*
Window *w;*
XStandardColormap **cmap_return;*
Atom *property;*

XImage *XGetSubImage(*display, d, x, y, width, height, plane_mask, format, dest_image, dest_x, dest_y*)
Display **display;*
Drawable *d;*
int *x, y, format, dest_x, dest_y;*
unsigned int *width, height;*
unsigned long *plane_mask;*
XImage **dest_image;*

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

XVisualInfo *XGetVisualInfo(*display, vinfo_mask, vinfo_template, nitems_return*)
Display **display;*
long *vinfo_mask;*
XVisualInfo **vinfo_template;*
int **nitems_return;*

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

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

XWMHints ***XGetWMHints**(*display, w*)
Display **display;*
Window *w;*

Status **XGetZoomHints**(*display, w, zhints_return*)
Display **display;*
Window *w;*
XSizeHints **zhints_return;*

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

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

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

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

XGrabServer(*display*)
    Display **display;*

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

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

XHPAcknowledge(*display, deviceid, acknowledge*)
    Display **display;*
    XID *deviceid;*
    unsigned int *acknowledge;*

XHPChangeDeviceControl(*display, deviceid, value_mask, values*)
    Display **display;*
    XID *deviceid;*
    unsigned long *value_mask;*
    XHPDeviceControl **values;*

XHPChangeDeviceKeyMapping(*display, deviceid, first_keycode, keysyms_per_keycode, keysyms, num_codes*)
    Display **display;*
    XID *deviceid;*
    int *first_keycode, keysyms_per_keycode, num_codes;*
    KeySyms **keysyms;*

XHPConvertLookup(*event_struct, buffer_return, bytes_buffer, keysym_return, status_in_out*)
Keyevent **event_struct;*
char **buffer_return;*
int *bytes_buffer;*
Keysym **keysym_return;*
XComposeStatus **status_in_out;*

XHPDeviceAutoRepeatOff(*display, deviceid*)
Display **display;*
XID *deviceid;*

XHPDeviceAutoRepeatOn(*display, deviceid, mode*)
Display **display;*
XID *deviceid;*
unsigned int *mode;*

XHPDisableReset(*display*)
Display **display;*

XHPEnableReset(*display*)
Display **display;*

int **XHPFileToPixmap**(*display, pixmap, cmap, gc, src_x, src_y, dst_x, dst_y, width, height, filename*)
Display **display;*
Pixmap *pixmap;*
Colormap *cmap;*
GC *gc;*
int *src_x, src_y, dst_x, dst_y;*
unsigned int *width, height;*
char *filename;*

int **XHPFileToWindow**(*display, w, modify_cmap, gc, src_x, src_y, dst_x, dst_y, width, height, filename*)
Display **display;*
Window *w;*
int *modify_cmap, src_x, src_y, dst_x, dst_y;*
GC *gc;*
unsigned int *width, height;*
char *filename;*

void **XHPFreeDeviceList**(*list*)
XHPDeviceList **list;*

XFontStruct **XHPGet16bitMixedFont**(*font*)
XFontStruct *font;

XHPGetCurrentDeviceMask(*display, window, deviceid, mask_return*)
Display **display;*
Window *window;*
XID *deviceid;*
Mask **mask_return;*

XHPGetDeviceControl(*display, deviceid, values_return*)
Display **display;*
XID *deviceid;*
XHPDeviceState **values_return;*

XHPGetDeviceFocus(*display, deviceid, focus_return, revert_to_return*)
Display **display;*
XID *deviceid;*
Window **focus_return;*
int **revert_to_return;*

KeySym ***XHPGetDeviceKeyMapping**(*display, deviceid, first_keycode_wanted, keycode_count, keysyms_per_keycode_return*)
Display **display;*
XID *deviceid;*
KeyCode *first_keycode_wanted;*
int *keycode_count, *keysyms_per_keycode_return;*

XModifierKeyMap ***XHPGetDevideModifierMapping**(*display, deviceid*)
Display **display;*
XID *deviceid*i;

XHPTimeCoord ***XHPGetDeviceMotionEvents**(*display, deviceid, w, start, stop, nevents_return*)
Display **display;*
XID *deviceid;*
Window *w;*
Time *start, stop;*
int **nevents_return;*

int **XHPGetExtEventMask**(*display, event_constant, event_type, mask*)
Display **display;*
long *event_constant, *event_type, *mask;*

XHPGetServerMode(*display, screen*)
Display **display;*
int *screen;*

XHPGrabDevice(*display, deviceid, grab_window, pointer_mode, device_mode, owner_events, time*)
Display **display;*
char *deviceid;*
Window *grab_window;*
int *pointer_mode, device_mode;*
Bool *owner_events;*
Time *time;*

XHPGrabDeviceButton(*display, deviceid, button, modifiers, grab_window, owner_events, event_mask, pointer_mode, device_mode*)
Display **display;*
XID *deviceid;*
unsigned int *button, modifiers, event_mask;*
Window *grab_window;*
Bool *owner_events;*
int *pointer_mode, device_mode;*

XHPGrabDeviceKey(*display, deviceid, keycode, modifiers, grab_window, owner_events, pointer_mode, device_mode*)
Display **display;*
XID *deviceid;*
unsigned int *keycode, modifiers;*
Window *grab_window;*
Bool *owner_events;*
int *pointer_mode, device_mode;*

XHPInputChinese_s(*display, keysym, modifiers, buffer_return, bytes_buffer, status_in_out*)
Display **display;*
KeySym *keysym;*
unsigned int *modifiers;*
char **buffer_return;*
int *bytes_buffer;*
XComposeStatus **status_in_out;*

XHPInputChinese_t(*display, keysym, modifiers, buffer_return, bytes_buffer, status_in_out*)
Display **display;*
KeySym *keysym;*
unsigned int *modifiers;*
char **buffer_return;*
int *bytes_buffer;*
XComposeStatus **status_in_out;*

XHPInputISO7sub(*display, keysym, modifiers, buffer_return, bytes_buffer, status_in_out*)
Display **display;*
KeySym *keysym;*
unsigned int *modifiers;*
char **buffer_return;*
int *bytes_buffer;*
XComposeStatus **status_in_out;*

XHPInputJapanese(*display, keysym, modifiers, buffer_return, bytes_buffer, status_in_out*)
Display **display;*
KeySym *keysym;*
unsigned int *modifiers;*
char **buffer_return;*
int *bytes_buffer;*
XComposeStatus **status_in_out;*

XHPInputKorean(*display, keysym, modifiers, buffer_return, bytes_buffer, status_in_out*)
Display **display;*
KeySym *keysym;*
unsigned int *modifiers;*
char **buffer_return;*
int *bytes_buffer;*
XComposeStatus **status_in_out;*

XHPInputRoman8(*display, keysym, modifiers, buffer_return, bytes_buffer, status_in_out*)
Display **display;*
KeySym *keysym;*
unsigned int *modifiers;*
char **buffer_return;*
int *bytes_buffer;*
XComposeStatus **status_in_out;*

Bool **XHPIs16bitCharacter**(*font, byte1, byte2*)
    Font *font;*
    unsigned char *byte1, byte2;*

int **XHPKeysymToRoman8**(*keysym, r8_return*)
    Keysym *keysym;*
    char **r8_return;*

XHPDeviceList ***XHPListInputDevices**(*display, ndevices*)
    Display **display;*
    int **ndevices;*

Status **XHPNlioctl**(*display, status_in_out, command, arg*)
    Display **display;*
    XComposeStatus **status_in_out;*
    int *command;*
    char **arg;*

int **XHPPixmapToFile**(*display, pixmap, color_w, x, y, width, height, plane_mask, format, filename*)
    Display **display;*
    Pixmap *pixmap;*
    Window *color_w;*
    int *x, y, format;*
    unsigned int *width, height;*
    long *plane_mask;*
    char *filename;*

XHPPrompt(*display, deviceid, prompt*)
    Display **display;*
    XID *deviceid;*
    unsigned int *prompt;*

int **XHPQueryImageFile**(*filename, xwd_header_return*)
    char *filename;*
    XWDFileHeader *xwd_header_return;*

Keysym **XHPRoman8ToKeysym**(*r8_char*)
    char *r8_char;*

XHPSelectExtensionEvent(*display, window, deviceid, mask*)
Display **display;*
Window *window;*
XID *deviceid;*
Mask *mask;*

XHPSetDeviceFocus(*display, deviceid, focus, revert_to, time*)
Display **display;*
XID *deviceid;*
Window *focus;*
int *revert_to;*
Time *time;*

XHPSetDeviceModifierMapping(*display, deviceid, modmap*)
Display **display;*
XID *deviceid;*
XModifierKeymap **modmap;*

PFI XHPSetErrorHandler(*display, routine*)
Display **display;*
int *(*routine) ();*

int XHPSetInputDevice(*display, deviceid, mode*)
Display **display;*
XID *deviceid;*
int *mode;*

Status XHPSetKeyboardMapping(*display, kbd_id*)
Display **display;*
KEYBOARD_ID *kbd_id;*

*XHPUngrabDevice(*display, deviceid, time*)
Display **display;*
XID *deviceid;*
Time *time;*

XHPUngrabDeviceButton(*display, deviceid, button, modifiers, ungrab_window*)
Display **display;*
XID *deviceid;*
unsigned int *button, modifiers;*
Window *ungrab_window;*

XHPUngrabDeviceKey(*display, deviceid, keycode, modifiers, ungrab_window*)
Display **display;*
XID *deviceid;*
unsigned int *keycode, modifiers;*
Window *ungrab_window;*

int **XHPWindowToFile**(*display, w, x, y, width, height, format, filename*)
Display **display;*
Window *w;*
int *x, y, format;*
unsigned int *width, height;*
char *filename;*

XIfEvent(*display, event_return, predicate, arg*)
Display **display;*
XEvent **event_return;*
Bool "(**predicate*)();
char **arg;*

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

XExtCodes ***XInitExtension**(*display, name*)
Display **display;*
char **name;*

XModifierKeymap ***XInsertModifiermapEntry**(*modmap, keycode_entry, modifier*)
XModifierKeymap **modmap;*
KeyCode *keycode_entry;*
int *modifier;*

XInstallColormap(*display, cmap*)
Display **display;*
Colormap *cmap;*

Atom **XInternAtom**(*display, atom_name, only_if_exists*)
Display **display;*
char **atom_name;*
Bool *only_if_exists;*

XIntersectRegion(*sra, srb, dr_return*)
Region *sra, srb, dr_return;*

IsCursorKey(*keysym*)
KeySym *keysym;*

IsFunctionKey(*keysym*)
KeySym *keysym;*

IsKeypadKey(*keysym*)
KeySym *keysym;*

IsMiscFunctionKey(*keysym*)
KeySym *keysym;*

IsModifierKey(*keysym*)
KeySym *keysym;*

IsPFKey(*keysym*)
KeySym *keysym;*

KeySym XKeycodeToKeysym(*display, keycode, index*)
Display **display;*
KeyCode *keycode;*
int *index;*

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

char *XKeysymToString(*keysym*)
KeySym *keysym;*

XKillClient(*display, resource*)
Display **display;*
XID *resource;*

unsigned long XLastKnownRequestProcessed(*display*)
Display **display;*

char **XListExtensions(*display, nextensions_return*)
Display **display;*
int **nextentions_return;*

char **XListFonts(*display, pattern, maxnames, actual_count_return*)
Display **display;*
char **pattern;*
int *maxnames, **actual_count_return;*

char **XListFontsWithInfo(*display, pattern, maxnames, count_return, info_return*)
    Display *display;
    char *pattern;
    int *maxnames, *count_return;
    XFontStruct **info_return;

XHostAddress *XListHosts(*display, nhosts_return, state_return*)
    Display *display;
    int *nhosts_return;
    Bool *state_return;

Colormap *XListInstalledColormaps(*display, w, num_return*)
    Display *display;
    Window *w*;
    int *num_return;

Atom *XListProperties(*display, w, num_prop_return*)
    Display *display;
    Window *w*;
    int *num_prop_returned;

Font XLoadFont(*display, name*)
    Display *display;
    char *name;

XFontStruct *XLoadQueryFont(*display, name*)
    Display *display;
    char *name;

LockDisplay(*display*)
    Display *display;

char *XLookUpAssoc(*display, table, x_id*)
    Display *display;
    XAssocTable *table;
    XID *x_id;

Status XLookupColor(*display, cmap, color_name, exact_def_return, screen_def_return*)
    Display *display;
    Colormap *cmap*;
    char *color_name;
    XColor *exact_def_return, *screen_def_return;

KeySym **XLookupKeysym**(*key_event, index*)
    XKeyEvent **key_event;*
    int *index;*

int **XLookupString**(*event_struct, buffer_return, bytes_buffer, keysym_return, status_in_out*)
    XKeyEvent **event_struct;*
    char **buffer_return;*
    int *bytes_buffer;*
    KeySym **keysym_return;*
    XComposeStatus **status_in_out;*

XLowerWindow(*display, w*)
    Display **display;*
    Window *w;*

XMakeAssoc(*display, table, x_id, data*)
    Display **display;*
    XAssocTable **table;*
    XID *x_id;*
    char **data;*

XMapRaised(*display, w*)
    Display **display;*
    Window *w;*

XMapSubwindows(*display, w*)
    Display **display;*
    Window *w;*

XMapWindow(*display, w*)
    Display **display;*
    Window *w;*

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

Status **XMatchVisualInfo**(*display, screen, depth, class, vinfo_return*)
    Display **display;*
    int *screen, depth, class;*
    XVisualInfo **vinfo_return;*

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

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

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

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

XNextEvent(*display, event_return*)
    Display **display;*
    XEvent **event_return;*

unsigned long **XNextRequest**(*display*)
    Display **display;*

XNoOp(*display*)
    Display **display;*

XOffsetRegion(*r, dx, dy*)
    Region *r;*
    int *dx, dy;*

Display ****XOpenDisplay**(*display_name*)
    char **display_name;*

Status **XParseColor**(*display, cmap, spec, exact_def_return*)
    Display **display;*
    Colormap *cmap;*
    char **spec;*
    XColor **exact_def_return;*

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

XPeekEvent(*display, event_return*)
    Display **display;*
    XEvent **event_return;*

XPeekIfEvent(*display, event_return, predicate, arg*)
    Display **display;*
    XEvent **event_return;*
    Bool (**predicate*)();
    char **arg;*

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

char ***Xpermalloc**(*size*)
    unsigned int *size;*

int **XPlanesOfScreen**(*screen*)
    Screen **screen;*

Bool **XPointInRegion**(*r, x, y*)
    Region *r;*
    int *x, y;*

Region **XPolygonRegion**(*points, n, fill_rule*)
    XPoint *points[ ];*
    int *n, fill_rule;*

int **XProtocolRevision**(*display*)
    Display **display*);

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

XPutBackEvent(*display, event*)
    Display **display;*
    XEvent **event;*

XPutImage(*display, d, gc, image, src_x, src_y, dst_x, dst_y, width, height*)
    Display **display;*
    Drawable *d;*
    GC *gc;*
    int *src_x, src_y, dst_x, dst_y;*
    unsigned int *width, height;*

int **XPutPixel**(*ximage, x, y, pixel*)
    XImage **ximage;*
    int *x, y;*
    unsigned long *pixel;*

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

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

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

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

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

XQueryColor(*display, cmap, def_in_out*)
    Display **display;*
    Colormap *cmap;*
    XColor **def_in_out;*

XQueryColors(*display, cmap, defs_in_out, ncolors*)
    Display **display;*
    Colormap *cmap;*
    XColor *defs_in_out[ ];*
    int *ncolors;*

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

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

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

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

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

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

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

XRaiseWindow(*display, w*)
    Display **display;*
    Window *w;*

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

XRebindKeysym(*display, keysym, list, mod_count, string, bytes_string*)
    Display **display;*
    KeySym *keysym, list[ ];*
    int *mod_count, bytes_string;*
    unsigned char **string;*

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

int **XRectInRegion**(*r, x, y, width, height*)
    Region *r;*
    int *x, y;*
    unsigned int *width, height;*

XRefreshKeyboardMapping(*event_map*)
    XMappingEvent **event_map;*

XRemoveFromSaveSet(*display, w*)
    Display **display;*
    Window *w;*

XRemoveHost(*display, host*)
Display **display;*
XHostAddress **host;*

XRemoveHosts(*display, hosts, num_hosts*)
Display **display;*
XHostAddress **hosts;*
int *num_hosts;*

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

XResetScreenSaver(*display*)
Display **display;*

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

XRestackWindows(*display, windows, nwindows*)
Display **display;*
Window *windows[ ];*
int *nwindows;*

XrmDatabase **XrmGetFileDatabase**(*filename*)
char **filename;*

Bool **XrmGetResource**(*database, str_name, str_class, str_type_return, str_value_return*)
XrmDatabase *database;*
char **str_name, *str_class, **str_type_return;*
XrmValue **value_return;*

XrmDatabase **XrmGetStringDatabase**(*string*)
char **data;*

void **XrmInitialize**()

void **XrmMergeDatabases**(*source_db, target_db*)
XrmDatabase *source_db, target_db;*

void **XrmParseCommand**(*db, table, table_count, name, argc_in_out, argv_in_out*)
XrmDatabase **db;*
XrmOptionDescList *table;*
int *table_count, *argc_in_out;*
char **name, **argv_in_out;*

void **XrmPutFileDatabase**(*database, stored_db*)
XrmDatabase *database;*
char **stored_db;*

void **XrmPutLineResource**(*database, line*)
XrmDatabase **database;*
char **line;*

void **XrmPutResource**(*database, specifier, type, value*)
XrmDatabase **database;*
char **specifier, type;*
XrmValue **value;*

void **XrmPutStringResource**(*database, specifier, value*)
XrmDatabase **database;*
char **specifier, *value;*

Bool **XrmQGetResource**(*database, quark_name, quark_class, quark_type_return, value_return*)
XrmDatabase *database;*
XrmNameList *quark_name;*
XrmClassList *quark_class;*
XrmRepresentation **quark_type_return;*
XrmValue **value_return;*

Bool **XrmQGetSearchList**(*database, names, classes, list_return, list_length*)
XrmDatabase *database;*
XrmNameList *names;*
XrmClassList *classes;*
XrmSearchList *list_return;*
int *list_length;*

Bool **XrmQGetSearchResource**(*list, name, class, type, type_return, value_return*)
XrmSearchList *list;*
XrmName *name;*
XrmClass *class;*
XrmRepresentation **type_return;*
XrmValue **value_return;*

void **XrmQPutResource**(*database, bindings, quarks, type, value*)
XrmDatabase **database;*
XrmBindingList *bindings;*
XrmQuarkList *quarks;*
XrmRepresentation *type;*
XrmValue **value;*

void **XrmQPutStringResource**(*database, bindings, quarks, value*)
XrmDatabase **database;*
XrmBindingList *bindings;*
XrmQuarkList *quarks;*
char **value;*

char ***XrmQuarkToString**(*quark*)
XrmQuark *quark;*

XrmStringToBindingQuarkList(*string, bindings_return, quarks_return*)
char **string;*
XrmBindingList *bindings_return;*
XrmQuarkList *quarks_return;*

XrmQuark **XrmStringToQuark**(*string*)
char **string;*

void **XrmStringToQuarkList**(*string, quarks_return*)
char **string;*
XrmQuarkList *quarks_return;*

XrmQuark **XrmUniqueQuark**()

Window **XRootWindow**(*display, screen_number*)
Display **display;*
int *screen_number;*

Window **XRootWindowOfScreen**(*screen*)
Screen **screen;*

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

XRotateWindowProperties(*display, w, properties, num_prop, npositions*)
    Display **display;*
    Window *w;*
    Atom *properties[ ];*
    int *num_prop, npositions;*

int **XSaveContext**(*display, w, context, data*)
    Display **display;*
    Window *w;*
    XContext *context;*
    caddr_t *data;*

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

Screen ***XScreenOfDisplay**(*display, screen_number*)
    Display **display;*
    int *screen_number;*

XSelectInput(*display, w, event_mask*)
    Display **display;*
    Window *w;*
    long *event_mask;*

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

char ***XServerVendor**(*display*)
    Display **display;*

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

int(***XSetAfterFunction**(*display, proc*))()
    Display **display;*
    int "(**proc*)();

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

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

XSetClassHint(*display, w, class_hints*)
    Display **display;*
    Window *w;*
    XClassHint **class_hints;*

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

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

XSetClipRectangles(*display, gc, clip_x_origin, clip_y_origin, rectangles, n, ordering*)
    Display **display;*
    GC *gc;*
    int *clip_x_origin, clip_y_origin, n, ordering;*
    XRectangle *rectangles[ ];*

XSetCloseDownMode(*display, close_mode*)
    Display **display;*
    int *close_mode;*

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

XSetDashes(*display, gc, dash_offset, dash_list, n*)
Display **display;*
GC *gc;*
int *dash_offset, n;*
char *dash_list[ ];*

XSetErrorHandler(*handler*)
int "(**handler*) (Display *, XErrorEvent *);

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

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

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

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

XSetForeground(*display, gc, foreground*)
Display **display;*
GC *gc;*
unsigned long *foreground;*

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

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

XSetIconName(*display, w, icon_name*)
Display **display;*
Window *w;*
char **icon_name;*

XSetIconSizes(*display, w, size_list, count*)
Display **display;*
Window *w;*
XIconSize **size_list;*
int *count;*

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

XSetIOErrorHandler(*handlers*)
int (**handler*)(Display *);

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

in **XSetModifierMapping**(*display, modmap*)
Display **display;*
XModifierKeymap **modmap;*

void **XSetNormalHints**(*display, w, hints*)
    Display **display;*
    Window *w;*
    XSizeHints **hints;*

XSetPlaneMask(*display, gc, plan_mask*)
    Display **display;*
    GC *gc;*
    unsigned long *plane_mask;*

int **XSetPointerMapping**(*display, map, nmap*)
    Display **display;*
    unsigned char *map[ ];*
    int *nmap;*

XSetRegion(*display, gc, r*)
    Display **display;*
    GC *gc;*
    Region *r;*

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

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

XSetSizeHints(*display, w, hints, property*)
    Display **display;*
    Window *w;*
    XSizeHints **hints;*
    Atom *property;*

void **XSetStandardColormap**(*display, w, cmap, property*)
    Display **display;*
    Window *w;*
    XStandardColormap **cmap;*
    Atom *property;*

XSetStandardProperties(*display, w, window_name, icon_name, icon_pixmap, argv, argc, hints*)
Display **display;*
Window *w;*
char **window_name, *icon_name, **argv;*
Pixmap *icon_pixmap;*
int *argc;*
XSizeHints **hints;*

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

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

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

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

XSetTransientForHint(*display, w, prop_window*)
Display **display;*
Window *w;*
Window *prop_window;*

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

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

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

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

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

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

XSetWindowColormap(*display, w, cmap*)
Display **display;*
Window *w;*
Colormap *cmap;*

XSetWMHints(*display, w, wmhints*)
Display **display;*
Window *w;*
XWMHints **wmhints;*

XSetZoomHints(*display, w, zhints*)
Display **display;*
Window *w;*
XSizeHints **zhints;*

XShrinkRegion(*r, dx, dy*)
Region *r;*
int *dy, dy;*

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

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

XStoreColor(*display, cmap, color*)
Display **display;*
Colormap *cmap;*
XColor *color;*

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

XStoreName(*display, w, window_name*)
Display **display;*
Window *w;*
char **window_name;*

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

KeySym **XStringToKeysym**(*string*)
char **string;*

XImage ***XSubImage**(*ximage, x, y, subimage_width, subimage_height*)
XImage **ximage;*
int *x, y;*
unsigned int *subimage_width, subimage_height;*

XSubtractRegion(*sra, srb, dr_return*)
Region *sra, srb, dr_return;*

XSync(*display, discard*)
Display **display;*
Bool *discard;*

int (*XSynchronize(*display, onoff*))()
    Display **display;*
    int *onoff;*

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

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

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

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

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

XUndefineCursor(*display, w*)
    Display **display;*
    Window *w;*

XUngrabButton(*display, button_ungrab, modifiers, ungrab_window*)
Display **display;*
unsigned int *button_ungrab, modifiers;*
Window *ungrab_window;*

XUngrabKey(*display, keycode, modifiers, ungrab_window*)
Display **display;*
int *keycode;*
unsigned int *modifiers;*
Window *ungrab_window;*

XUngrabKeyboard(*display, time*)
Display **display;*
Time *time;*

XUngrabPointer(*display, time*)
Display **display;*
Time *time;*

XUngrabServer(*display*)
Display **display;*

XContext **XUniqueContext**()

XUninstallColormap(*display, cmap*)
Display **display;*
Colormap *cmap;*

XUnionRectWithRegion(*rectangle, src_region, dest_region_return*)
Rectangle **rectangle;*
Region *src_region, dest_region_return;*

XUnionRegion(*sra, srb, dr_return*)
Region *sra, srb, dr_return;*

XUnloadFont(*display, font*)
Display **display;*
Font *font;*

UnlockDisplay(*display*)
Display **display;*

XUnmapSubwindows(*display, w*)
Display **display;*
Window *w;*

XUnmapWindow(*display, w*)
Display **display;*
Window *w;*

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

VisualID **XVisualIDFromVisual**(*visual*)
Visual **viaual;*

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

unsigned long **XWhitePixel**(*display, screen_number*)
Display **display;*
int *screen_number;*

unsigned long **XWhitePixelOfScreen**(*screen*)
Screen **screen;*

int **XWidthMMOfScreen**(*screen*)
Screen **screen;*

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

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

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

XXorRegion(*sra, srb, dr_return*)
Region *sra, srb, dr_return;*

3

HP Motif Widgets, Gadgets, and Convenience Functions

For more information about widgets, gadgets, and convenience functions, refer to *HP Motif Programmer's Guide* or *HP Motif Programmer's Reference.*

Resource Sets

The following resource sets may be used by widgets or gadgets.

Class Name	Widget Class	Resources
ApplicationShell	applicationShellWidgetClass	Core Composite Shell WMShell VendorShell TopLevelShell
Composite	compositeWidgetClass	Core
Constraint	constraintWidgetClass	Core Composite
Core	coreWidgetClass	
Object	objectWidgetClass	
OverrideShell	overrideShellWidgetClass	Core Composite Shell

Class Name	Widget Class	Resources
RectObj	rectObjWidgetClass	Object
Shell	shellWidgetClass	Core Composite
TopLevelShell	topLevelShellWidgetClass	Core Composite Shell WMShell VendorShell
TransientShell	transientShellWidgetClass	Core Composite Shell WMShell
VendorShell	vendorShellWidgetClass	Core Composite Shell WMShell
WMShell	wmShellWidgetClass	Core Composite Shell

Widgets

The widgets in this table are listed in alphabetical order by widget class name.

Class Name	Widget Class	Resources
XmArrowButton	xmArrowButtonWidgetClass	Core Primitive
XmBulletinBoard	xmBulletinBoardWidgetClass	Core Composite XmManager
XmCommand	xmCommandWidgetClass	Core Composite XmManager XmBulletinBoard XmSelectionBox
XmDialogShell	xmDialogShellWidgetClass	Core Composite Shell WMShell VendorShell TransientShell
XmDrawingArea	xmDrawingAreaWidgetClass	Core Composite XmManager
XmDrawnButton	xmDrawnButtonWidgetClass	Core XmPrimitive XmLabel
XmFileSelectionBox	xmFileSelectionBoxWidgetClass	Core Composite XmManager XmBulletinBoard XmSelectionBox

Class Name	Widget Class	Resources
XmForm	xmFormWidgetClass	Core Composite XmManager XmBulletinBoard
XmFrame	xmFrameWidgetClass	Core Composite XmManager
XmLabel	xmLabelWidgetClass	Core XmPrimitive
XmList	xmListWidgetClass	Core XmPrimitive
XmMainWindow	xmMainWindowWidgetClass	Core Composite XmManager XmScrolledWindow
XmManager	xmManagerWidgetClass	Core Composite
XmMenuShell	xmMenuShellWidgetClass	Core Composite Shell OverrideShell
XmMessageBox	xmMessageBoxWidgetClass	Core Composite XmManager XmBulletinBoard
XmPanedWindow	xmPanedWindowWidgetClass	Core Composite XmManager
XmPrimitive	xmPrimitiveWidgetClass	Core
XmPushButton	xmPushButtonWidgetClass	Core XmPrimitive XmLabel

Class Name	Widget Class	Resources
XmRowColumn	xmRowColumnWidgetClass	Core Composite XmManager
XmScale	xmScaleWidgetClass	Core Composite XmManager
XmScrollBar	xmScrollBarWidgetClass	Core XmPrimitive
XmScrolledWindow	xmScrolledWindowWidgetClass	Core Composite XmManager
XmSelectionBox	xmSelectionBoxWidgetClass	Core Composite XmManager XmBulletinBoard
XmSeparator	xmSeparatorWidgetClass	Core XmPrimitive
XmText	xmTextWidgetClass	Core XmPrimitive
XmToggleButton	xmToggleButtonWidgetClass	Core XmPrimitive XmLabel
XmVendorShell	xmVendorShellWidgetClass	Core Composite Shell WMShell

Gadgets

The gadgets in this table are listed in alphabetical order by gadget class name.

Class Name	Gadget Class	Resources
XmArrowButtonGadget	xmArrowButtonGadgetClass	Object RectObj XmGadget XmLabelGadget
XmGadget	xmGadgetClass	Object RectObj
XmLabelGadget	xmLabelGadgetClass	Object RectObj XmGadget
XmPushButtonGadget	xmPushButtonGadgetClass	Object RectObj XmGadget XmLabelGadget
XmSeparatorGadget	xmSeparatorGadgetClass	Object RectObj XmGadget
XmToggleButtonGadget	xmToggleButtonGadgetClass	Object RectObj XmGadget XmLabelGadget

Convenience Functions

Convenience functions all use the same format:

Widget `Function-name`(*parent, name, arglist, argcount*)
Widget *parent;*
String *name;*
ArgList *arglist;*
Cardinal *argcount;*

Function Name	Function Name
XmCreateArrowButton	XmCreateFrame
XmCreateArrowButtonGadget	XmCreateInformationDialog
XmCreateBulletinBoard	XmCreateLabel
XmCreateBulletinBoardDialog	XmCreateLabelGadget
XmCreateCascadeButton	XmCreateMainWindow
XmCreateCascadeButtonGadget	XmCreateList
XmCreateCommand	XmCreateMenuBar
XmCreateDialogShell	XmCreateMenuShell
XmCreateDrawnButton	XmCreateMessageBox
XmCreateDrawingArea	XmCreateMessageDialog
XmCreateErrorDialog	XmCreateOptionMenu
XmCreateFileSelectionBox	XmCreatePanedWindow
XmCreateFileSelectionDialog	XmCreatePopupMenu
XmCreateForm	XmCreatePromptDialog
XmCreateFormDialog	XmCreatePushButton

Function Name	Function Name
XmCreatePushButtonGadget	XmCreateScrolledText
XmCreatePulldownMenu	XmCreateScrolledWindow
XmCreateQuestionDialog	XmCreateSelectionBox
XmCreateRadioBox	XmCreateSelectionDialog
XmCreateRowColumn	XmCreateSeparatorGadget
XmCreateScale	XmCreateText
XmCreateSeparator	XmCreateToggleButton
XmCreateScrollBar	XmCreateToggleButtonGadget
XmCreateScrolledList	XmCreateWarningDialog

MANUAL COMMENT CARD

HP Part Number 98794-90003 E0989

Your comments and suggestions help us determine how well we meet your needs.

X Window System C
Quick Reference Guide

	Agree				**Disagree**
The manual is well organized.	○	○	○	○	○
It is easy to find information in the manual.	○	○	○	○	○
The manual explains features well.	○	○	○	○	○
The manual contains enough examples.	○	○	○	○	○
The examples are appropriate for my needs.	○	○	○	○	○
The manual covers enough topics.	○	○	○	○	○
Overall, the manual meets my expectations.	○	○	○	○	○

You have used this product:

__ Less than 1 week __ Less than 1 year __ More than 2 years
__ Less than 1 month __ 1 to 2 years

fold

Please write additional comments, particularly if you disagree with a statement above. Use additional pages if you wish. The more specific your comments, the more useful they are to us.

Comments: _____

Please print or type your name and address.

Name: _____

Company: _____

Address: _____

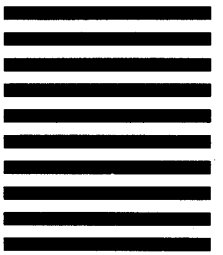
City, State, Zip: _____

Telephone: _____

Additional Comments: _____

X Window System C Quick Reference Guide
HP Part Number 98794-90003
E0989

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 38 CORVALLIS, OR

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD COMPANY
CIS PRODUCT MARKETING
1000 NE CIRCLE BLVD.
CORVALLIS, OR 97330-9974

HP Part Number
98794-90003
Microfiche No. 98794-99003
Printed in U.S.A. E0989

98794-90603
For Internal Use Only