X Window System
C Quick Reference Guide

HP Part No. B1171-90056
Printed in U.S.A. July 1992

Edition 4
DRAFT 9/12/97 20:47

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 1.2 user environment is pending.

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

Hewlett-Packard Company
User Interface Technology Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330

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
February 1991 ... Edition 3
July 1992 ... Edition 4

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 X11 Xt Intrinsics, X11 Xlib, and the OSF/Motif widgets.

This quick reference guide is organized as follows:

Chapter 1	R5 Xt Intrinsics functions alphabetized by name.
Chapter 2	R5 Xlib functions alphabetized by name.
Chapter 3	R5 Xlib macros alphabetized by name.
Chapter 4	OSF/Motif 1.2 functions alphabetized by name.

Contents

1

X11 R5 Toolkit Intrinsics Functions

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

void `XtAddCallback`(*object, callback_name, callback, client_data*)
    Widget *object*;
    String *callback_name*;
    XtCallbackProc *callback*;
    XtPointer *client_data*;

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

void `XtAddConverter`(*from_type, to_type, converter, convert_args, num_args*)
    String *from_type*;
    String *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*;
    XtPointer *client_data*;

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

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

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

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

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

XtWorkProcId **XtAddWorkProc**(*proc, client_data*)
XtWorkProc *proc*;
XtPointer *client_data*;

GC **XtAllocateGC**(*object, depth, value_mask, values, dynamic_mask, dont_care_mask*)
Widget *object*;
Cardinal *depth*;
XtGCMask *value_mask, dynamic_mask, dont_care_mask*;
XtGCValues **values*;

XtActionHookId **XtAppAddActionHook**(*app, proc, client_data*)
XtAppContext *app*;
XtActionHookProc *proc*;
XtPointer *client_data*;

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*;
XtPointer *condition*;
XtInputCallbackProc *proc*;
XtPointer *client_data*;

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

XtWorkProcId **XtAppAddWorkProc**(*app_context, proc, client_data*)
XtAppContext *app_context*;
XtWorkProc *proc*;
XtPointer *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*;
String *name, type, class, default, buffer_return*;
int *nbytes*;
XrmDatabase *database*;

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

Widget **XtAppInitialize**(*app_context_return, application_class, options, num_options, argc_in_out, argv_in_out, fallback_resources, args, num_args*)
XtAppContext **app_context_return*;
String *application_class*;
XrmOptionDescList *options*;
Cardinal *num_options, num_args*;
int **argc_in_out*;
String **argv_in_out, *fallback_resources*;
ArgList *args*;

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*;

XtInputMask **XtAppPending**(*app_context*)
XtAppContext *app_context*;

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

void **XtAppReleaseCacheRefs**(*app, refs*)
XtAppContext *app*;
XtCacheRef **refs*;

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

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

void **XtAppSetFallbackResources**(*app_context, specification_list*)
XtAppContext *app_context*;
String **specification_list*;

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

void **XtAppSetTypeConverter**(*app_context, from_type, to_type, converter, convert_args, num_args, cache_type, destructor*)
XtAppContext *app_context*;
String *from_type, to_type*;
XtTypeConverter *converter*;
XtConvertArgList *convert_args*;
Cardinal *num_args*;
XtCacheType *cache_type*;
XtDestructor *destructor*;

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

XtErrorMsgHandler **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*;

EventMask **XtBuildEventMask**(*w*)
Widget *w*;

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

void **XtCallActionProc**(*widget, action, event, params, num_params*)
Widget *widget*;
String *action, *params*;
XEvent **event*;
Cardinal *num_params*;

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

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

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

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

void XtCallbackReleaseCacheRef(*object, client_data, call_data*)
    Widget *object*;
    XtPointer *client_data, call_data*;

void XtCallbackReleaseCacheRefList(*object, client_data, call_data*)
    Widget *object*;
    XtPointer *client_data, call_data*;

void XtCallCallbackList(*widget, callbacks, call_data*)
    Widget *widget*;
    XtCallbackList *callbacks*;
    XtPointer *call_data*;

void XtCallCallbacks(*widget, callback_name, call_data*)
    Widget *widget*;
    String *callback_name*;
    XtPointer *call_data*;

Boolean XtCallConverter(*display, converter, args, num_args, from, to_in_out, cache_ref_return*)
    Display* *display*;
    XtTypeConverter *converter*;
    XrmValuePtr *args, from, to_in_out*;
    Cardinal *num_args*;
    XtCacheRef **cache_ref_return*;

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

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

WidgetClass XtClass(*widget*)
    Widget *widget*;

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*;

Boolean XtConvertAndStore(*object, from_type, from, to_type, to_in_out*)
Widget *object*;
String *from_type, to_type*;
XrmValuePtr *from, to_in_out*;

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

XtAppContext XtCreateApplicationContext()

Widget XtCreateApplicationShell(*application_name, widget_class, args, num_args*)
String *application_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, object, parent, args, num_args*)
String *name*;
WidgetClass *object*;
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(*w, gc*)
Widget *w*;
GC *gc*;

void XtDestroyWidget(*widget*)
Widget *widget*;

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*;

Boolean 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, *argv*;
XrmOptionDescRec **options*;
Cardinal *num_options*;
int **argc*;

Display *XtDisplayOfObject(*object*)
Widget *object*;

void XtDisplayStringConversionWarning(*display, from_value, to_type*)
Display **display*;
String *from_value, to_type*;

XtAppContext **XtDisplayToApplicationContext**(*display*)
Display **display*;

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

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

String **XtFindFile**(*path, substitutions, num_substitutions, predicate*)
String *path*;
Substitution *substitutions*;
Cardinal *num_substitutions*;
XtFilePredicate *predicate*;

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

KeySym **XtGetActionKeysym**(*event, modifiers_return*)
XEvent **event*;
Modifiers **modifiers_return*;

void **XtGetActionList**(*widget_class, actions_return, num_actions_return*)
WidgetClass *widget_class*;
XtActionList **actions_return*;
Cardinal **num_actions_return*;

void **XtGetApplicationNameAndClass**(*display, name_return, class_return*)
Display **display*;
String **name_return, class_return*;

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

void **XtGetConstraintResourceList**(*object_class, resources_return, num_resources_return*)
WidgetClass *object_class*;
XtResourceList **resources_return*;
Cardinal **num_resources_return*;

XrmDatabase ***XtGetErrorDatabase**()

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

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

KeySym *XtGetKeysymTable(*display, min_keycode_return, keysyms_per_keycode_return*)
Display **display*;
KeyCode **min_keycode_return*;
int **keysyms_per_keycode_return*;

int XtGetMultiClickTime(*display*)
Display **display*;

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

XSelectionRequestEvent *XtGetSelectionRequest(*w, selection, request_id*)
Widget *w*;
Atom *selection*;
XtRequestId *request_id*;

unsigned long XtGetSelectionTimeout()

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

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

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

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

void XtGetSubresources(*w, base, subpart_name, subpart_class, resources, num_resources, args, num_args*)
Widget *w*;
XtPointer *base*;
String *subpart_name, subpart_class*;
XtResourceList *resources*;
Cardinal *num_resources, num_args*;
ArgList *args*;

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

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

void XtGrabButton(*widget, button, modifiers, owner_events, event_mask, pointer_mode, keyboard_mode, confine_to, cursor*)
Widget *widget*;
int *button, pointer_mode, keyboard_mode*;
Modifiers *modifiers*;
Boolean *owner_events*;
unsigned int *event_mask*;
Window *confine_to*;
Cursor *cursor*;

void XtGrabKey(*widget, keycode, modifiers, owner_events, pointer_mode, keyboard_mode*)
Widget *widget*;
KeyCode *keycode*;
Modifiers *modifiers*;
Boolean *owner_events*;
int *pointer_mode, keyboard_mode*;

int XtGrabKeyboard(*widget, owner_events, pointer_mode, keyboard_mode, time*)
Widget *widget*;
Boolean *owner_events*;
int *pointer_mode, keyboard_mode*;
Time *time*;

int XtGrabPointer(*widget, owner_events, event_mask, pointer_mode, keyboard_mode, confine_to, cursor, time*)
Widget *widget*;
Boolean *owner_events*;
unsigned int *event_mask*;
int *pointer_mode, keyboard_mode*;
Window *confine_to*;
Cursor *cursor*;
Time *time*;

XtCallbackStatus XtHasCallbacks(*object, callback_name*)
Widget *object*;
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*;
char **argv*[];

void XtInitializeWidgetClass(*object_class*)
WidgetClass *object_class*;

void XtInsertEventHandler(*w, event_mask, nonmaskable, proc, client_data, position*)
Widget *w*;
EventMask *event_mask*;
Boolean *nonmaskable*;
XtEventHandler *proc*;
XtPointer *client_data*;
XtListPosition *position*;

void XtInsertRawEventHandler(*w, event_mask, nonmaskable, proc, client_data, position*)
Widget *w*;
EventMask *event_mask*;
Boolean *nonmaskable*;
XtEventHandler *proc*;
XtPointer *client_data*;
XtListPosition *position*;

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

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

Boolean XtIsApplicationShell(*object*)
Widget *object*;

Boolean XtIsComposite(*object*)
Widget *object*;

Boolean XtIsConstraint(*object*)
Widget *object*;

Boolean XtIsManaged(*object*)
Widget *object*;

Boolean XtIsObject(*object*)
Widget *object*;

Boolean XtIsOverrideShell(*object*)
Widget *object*;

Boolean XtIsRealized(*object*)
Widget *object*;

Boolean XtIsRectObj(*object*)
Widget *object*;

Boolean XtIsSensitive(*object*)
Widget *object*;

Boolean XtIsShell(*object*)
Widget *object*;

Boolean XtIsSubclass(*object, object_class*)
    Widget *object*;
    WidgetClass *object_class*;

Boolean XtIsTopLevelShell(*object*)
    Widget *object*;

Boolean XtIsTransientShell(*object*)
    Widget *object*;

Boolean XtIsVendorShell(*object*)
    Widget *object*;

Boolean XtIsWMShell(*object*)
    Widget *object*;

Boolean XtIsWidget(*object*)
    Widget *object*;

void XtKeysymToKeycodeList(*display, keysym, keycodes_return, keycount_return*)
    Display **display*;
    KeySym *keysym*;
    KeyCode ***keycodes_return*;
    Cardinal **keycount_return*;

Time XtLastTimestampProcessed(*display*)
    Display **display*;

void XtMainLoop()

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, *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**(*args1, num_args1, args2, num_args2*)
ArgList *args1, args2*;
Cardinal *num_args1, num_args2*;

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

String **XtName**(*object*)
Widget *object*;

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

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

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

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

Cardinal **XtNumber**(*array*)

Cardinal **XtOffset**(*pointer_type, field_name*)

Cardinal **XtOffsetOf**(*structure_type, field_name*)

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

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

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

Boolean XtOwnSelectionIncremental(*w, selection, time, convert_callback, lose_callback, done_callback, cancel_callback, client_data*)
Widget *w*;
Atom *selection*;
Time *time*;
XtConvertSelectionIncrProc *convert_callback*;
XtLoseSelectionIncrProc *lose_callback*;
XtSelectionDoneIncrProc *done_callback*;
XtCancelConvertSelectionProc *cancel_callback*;
XtPointer *client_data*;

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

XtAccelerators XtParseAcceleratorTable(*table*)
String *table*;

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

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

XtInputMask XtPending()

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

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

void XtPopupSpringLoaded(*popup_shell*)
Widget *popup_shell*;

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 XtRegisterGrabAction(*action_proc, owner_events, event_mask, pointer_mode, keyboard_mode*)
XtActionProc *action_proc*;
Boolean *owner_events*;
unsigned int *event_mask*;
int *pointer_mode, keyboard_mode*;

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

void XtRemoveActionHook(*id*)
XtActionHookId *id*;

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*;
XtPointer *client_data*;

void XtRemoveCallbacks(*w, callback_name, callbacks*)
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*;
XtPointer *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*;
XtPointer *client_data*;

void XtRemoveTimeOut(*id*)
XtIntervalId *id*;

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

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

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

String XtResolvePathname(*display, type, filename, suffix, path, substitutions, num_substitutions, predicate*)
Display **display*;
String *type, filename, suffix, path*;
Substitution *substitutions*;
Cardinal *num_substitutions*;
XtFilePredicate *predicate*;

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

XrmDatabase XtScreenDatabase(*screen*)
Screen **screen*;

Screen *XtScreenOfObject(*object*)
Widget *object*;

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

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

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

void XtSetKeyboardFocus(*subtree, descendant*)
Widget *subtree, descendant*;

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

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

void XtSetMultiClickTime(*display, time*)
Display **display*;
int *time*;

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

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

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

void XtSetTypeConverter(*from_type, to_type, converter, convert_args, num_args, cache_type, destructor*)
String *from_type, to_type*;
XtTypeConverter *converter*;
XtConvertArgList *convert_args*;
Cardinal *num_args*;
XtCacheType *cache_type*;
XtDestructor *destructor*;

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

void XtSetWarningHandler(*handler*)
XtErrorHandler *handler*;

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

void XtSetWMColormapWindows(*widget, list, count*)
Widget *widget, *list*;
Cardinal *count*;

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

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

void XtToolkitInitialize()

void XtTranslateCoords(*w, x, y, root_x_return, root_y_return*)
Widget *w*;
Position *x, y, *root_x_return, *root_y_return*;

void XtTranslateKey(*display, keycode, modifiers, modifiers_return, keysym_return*)
Display **display*;
KeyCode *keycode*;
Modifiers *modifiers, *modifiers_return*;
KeySym **keysym_return*;

void XtTranslateKeycode(*display, keycode, modifiers, modifiers_return, keysym_return*)
Display **display*;
KeyCode *keycode*;
Modifiers *modifiers, *modifiers_return*;
KeySym **keysym_return*;

void XtUngrabButton(*widget, button, modifiers*)
Widget *widget*;
unsigned int *button*;
Modifiers *modifiers*;

void XtUngrabKey(*widget, keycode, modifiers*)
Widget *widget*;
KeyCode *keycode*;
Modifiers *modifiers*;

void XtUngrabKeyboard(*widget, time*)
Widget *widget*;
Time *time*;

void XtUngrabPointer(*widget, time*)
Widget *widget*;
Time *time*;

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*;

Widget XtVaAppCreateShell(*application_name, application_class, widget_class, display, ...)*
String *application_name, application_class*;
WidgetClass *widget_class*;
Display **display*;

Widget XtVaAppInitialize(*app_context_return, application_class, options, num_options, argc_in_out, argv_in_out, fallback_resources, ...)*
XtAppContext **app_context_return*;
String *application_class*;
XrmOptionDescList *options*;
Cardinal *num_options*;
int **argc_in_out*;
String **argv_in_out, *fallback_resources*;

XtVarArgsList XtVaCreateArgsList(*unused, ...)*
XtPointer *unused*;

Widget XtVaCreateManagedWidget(*name, widget_class, parent, ...)*
String *name*;
WidgetClass *widget_class*;
Widget *parent*;

Widget XtVaCreatePopupShell(*name, widget_class, parent, ...)*
String *name*;
WidgetClass *widget_class*;
Widget *parent*;

Widget XtVaCreateWidget(*name, object_class, parent, ...)*
String *name*;
WidgetClass *object_class*;
Widget *parent*;

void XtVaGetApplicationResources(*object, base, resources, num_resources, ...)*
Widget *object*;
XtPointer *base*;
XtResourceList *resources*;
Cardinal *num_resources*;

void XtVaGetSubresources(*object, base, name, class, resources, num_resources, ...)*
Widget *object*;
XtPointer *base*;
String *name, class*;
XtResourceList *resources*;
Cardinal *num_resources*;

void **XtVaGetSubvalues**(*base, resources, num_resources, ...)*
XtPointer *base*;
XtResourceList *resources*;
Cardinal *num_resources*;

void **XtVaGetValues**(*object, ...)*
Widget *object*;

void **XtVaSetSubvalues**(*base, resources, num_resources, ...)*
XtPointer *base*;
XtResourceList *resources*;
Cardinal *num_resources*;

void **XtVaSetValues**(*object, ...)*
Widget *object*;

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*;

Window **XtWindowOfObject**(*object*)
Widget *object*;

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

2

X11 R5 Xlib Functions

Header files for these functions can be found in `libXi.a`, `Xcms.h`, `Xlib.h`, `Xresource.h`, and `Xutil.h`.

void `XActivateScreenSaver`(*display*)
    Display **display*;

XExtCodes `*XAddExtension`(*display*)
    Display **display*;

`XAddToExtensionList`(*ext_data*)
    XExtData ***ext_data*;

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

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

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

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

XClassHint `*XAllocClassHint`()

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

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

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

XIconSize *XAllocIconSize()

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

XSizeHints *XAllocSizeHints()

XStandardColormap *XAllocStandardColormap()

XWMHints *XAllocWMHints()

int XAllowDeviceEvents(*display, device, event_mode, time*)
Display **display*;
XDevice **device*;
int *event_mode*;
Time *time*;

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

unsigned long XAllPlanes()

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

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

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

void 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*;

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

int XChangeDeviceControl(*display, device, controlType, control*)
Display **display*;
XDevice **device*;
int *controlType*;
XDeviceControl **control*;

int XChangeDeviceDontPropagateList(*display, window, count, event_list, mode*)
Display **display*;
Window *window*;
int *count, mode*;
XEventClass **event_list*;

int XChangeDeviceKeyMapping(*display, device, first_keycode, keysyms_per_keycode, keysyms, keycode_count*)
Display **display*;
XDevice **device*;
int *first_keycode, keysyms_per_keycode, keycode_count*;
KeySym **keysyms*;

int XChangeFeedbackControl(*display, device, mask, control*)
Display **display*;
XDevice **device*;
Mask *mask*;
XFeedbackControl **control*;

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

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

Status XChangeKeyboardDevice(*display, device*)
Display **display*;
XDevice **device*;

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

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

Status XChangePointerDevice(*display, device xaxis, yaxis*)
Display **display*;
XDevice **device*;
int *xaxis, yaxis*;

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

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

void 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)()*;
XPointer **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*;

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

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

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

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

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

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

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

int **XCloseDevice**(*display, device*)
Display **display*;
XDevice **device*;

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

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

Status **XcmsAllocColor**(*display, colormap, color_in_out, result_format*)
Display **display*;
Colormap *colormap*;
XcmsColor **color_in_out*;
XcmsColorFormat *result_format*;

Status **XcmsAllocNamedColor**(*display, colormap, color_string, result_format, color_screen_return, color_exact_return*)
Display **display*;
Colormap *colormap*;
char **color_string*;
XcmsColorFormat *result_format*;
XcmsColor **color_screen_return, *color_exact_return*;

XcmsCCC XcmsCCCofColormap(*display, colormap*)
    Display **display*;
    Colormap *colormap*;

Status XcmsCIELabQueryMaxC(*ccc, hue_angle, L_star, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, L_star*;
    XcmsColor **color_return*;

Status XcmsCIELabQueryMaxL(*ccc, hue_angle, chroma, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, chroma*;
    XcmsColor **color_return*;

Status XcmsCIELabQueryMaxLC(*ccc, hue_angle, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, *color_return*;

Status XcmsCIELabQueryMinL(*ccc, hue_angle, chroma, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, chroma*;
    XcmsColor **color_return*;

Status XcmsCIELuvQueryMaxC(*ccc, hue_angle, L_star, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, L_star*;
    XcmsColor **color_return*;

Status XcmsCIELuvQueryMaxL(*ccc, hue_angle, chroma, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, chroma*;
    XcmsColor **color_return*;

Status XcmsCIELuvQueryMaxLC(*ccc, hue_angle, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle*;
    XcmsColor **color_return*;

Status XcmsCIELuvQueryMinL(*ccc, hue_angle, chroma, color_return*)
    XcmsCCC *ccc*;
    XcmsFloat *hue_angle, chroma*;
    XcmsColor **color_return*;

Status **XcmsConvertColors**(*ccc, colors_in_out, ncolors, target_format, compression_flags_return*)
XcmsCCC *ccc*;
XcmsColor *colors_in_out[]*;
unsigned int *ncolors*;
XcmsColorFormat *target_format*;
Bool *compression_flags_return[]*;

XcmsCCC **XcmsCreateCCC**(*display, screen_number, visual, client_white_point, compression_proc, compression_client_data, white_adjust_proc, white_adjust_client_data*)
Display **display*;
int *screen_number*;
Visual **visual*;
XcmsColor **client_white_point*;
XcmsCompressionProc *compression_proc*;
XPointer *compression_client_data*;
XcmsWhiteAdjustProc *white_adjust_proc*;
XPointer *white_adjust_client_data*;

XcmsCCC **XcmsDefaultCCC**(*display, screen_number*)
Display **display*;
int *screen_number*;

void **XcmsFreeCCC**(*ccc*)
XcmsCCC *ccc*;

Status **XcmsLookupColor**(*display, colormap, color_string, color_exact_return, color_screen_return, result_format*)
Display **display*;
Colormap *colormap*;
char **color_string*;
XcmsColor **color_exact_return, *color_screen_return*;
XcmsColorFormat *result_format*;

Status **XcmsQueryBlack**(*ccc, target_format, color_return*)
XcmsCCC *ccc*;
XcmsColorFormat *target_format*;
XcmsColor **color_return*;

Status **XcmsQueryBlue**(*ccc, target_format, color_return*)
XcmsCCC *ccc*;
XcmsColorFormat *target_format*;
XcmsColor **color_return*;

Status XcmsQueryColor(*display, colormap, color_in_out, result_format*)
Display **display*;
Colormap *colormap*;
XcmsColor **color_in_out*;
XcmsColorFormat *result_format*;

Status XcmsQueryColors(*display, colormap, colors_in_out, ncolors, result_format*)
Display **display*;
Colormap *colormap*;
XcmsColor *colors_in_out[]*;
unsigned int *ncolors*;
XcmsColorFormat *result_format*;

Status XcmsQueryGreen(*ccc, target_format, color_return*)
XcmsCCC *ccc*;
XcmsColorFormat *target_format*;
XcmsColor **color_return*;

Status XcmsQueryRed(*ccc, target_format, color_return*)
XcmsCCC *ccc*;
XcmsColorFormat *target_format*;
XcmsColor **color_return*;

Status XcmsQueryWhite(*ccc, target_format, color_return*)
XcmsCCC *ccc*;
XcmsColorFormat *target_format*;
XcmsColor **color_return*;

XcmsCCC XcmsSetCCCOfColormap(*display, colormap, ccc*)
Display **display*;
Colormap *colormap*;

XcmsWhiteAdjustProc XcmsSetWhiteAdjustProc(*ccc, white_adjust_proc, client_data*)
XcmsCCC *ccc*;
XcmsWhiteAdjustProc *white_adjust_proc*;
XPointer *client_data*;

Status XcmsSetWhitePoint(*ccc, color*)
XcmsCCC *ccc*;
XcmsColor **color*;

Status XcmsStoreColor(*display, colormap, color*)
Display **display*;
Colormap *colormap*;
XcmsColor **color*;

Status **XcmsStoreColors**(*display, colormap, colors, ncolors, compression_flags_return*)
Display **display*;
Colormap *colormap*;
XcmsColor *colors[]*;
int *ncolors*;
Bool *compression_flags_return[]*;

Status **XcmsTekHVCQueryMaxC**(*ccc, hue, value, color_return*)
XcmsCCC *ccc*;
XcmsFloat *hue, value*;
XcmsColor **color_return*;

Status **XcmsTekHVCQueryMaxV**(*ccc, hue, chroma, color_return*)
XcmsCCC *ccc*;
XcmsFloat *hue, chroma*;
XcmsColor **color_return*;

Status **XcmsTekHVCQueryMaxVC**(*ccc, hue, color_return*)
XcmsCCC *ccc*;
XcmsFloat *hue*;
XcmsColor **color_return*;

Status **XcmsTekHVCQueryMaxVSamples**(*ccc, hue, colors_return, nsamples*)
XcmsCCC *ccc*;
XcmsFloat *hue*;
XcmsColor *colors_return[]*;
unsigned int *nsamples*;

Status **XcmsTekHVCQueryMinV**(*ccc, hue, chroma, color_return*)
XcmsCCC *ccc*;
XcmsFloat *hue, chroma*;
XcmsColor **color_return*;

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

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

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

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

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

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

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

void 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*;
unsigned int *width, height*;
int *dest_x, dest_y*;
unsigned long *plane*;

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*;

XFontSet XCreateFontSet(*display, base_font_name_list, missing_charset_list_return, missing_charset_count_return, def_string_return*)
Display **display*;
char **base_font_name_list*;
char ***missing_charset_list_return*;
int **missing_charset_count_return*;
char ***def_string_return*;

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

XIC XCreateIC(*im, ... *)
XIM *im*;

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

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, x, y*)
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*;
unsigned long *fg, bg*;
unsigned int *depth*;

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*;
unsigned int *width, height, border_width*;
int *depth*;
unsigned int *class*;
Visual *visual
unsigned long *valuemask*;
XSetWindowAttributes **attributes*;

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

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

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

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*;

char *XDefaultString()

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

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

int XDeleteContext(*display, rid, context*)
Display **display*;
XID *rid*;
XContext *context*;

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

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

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

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

void XDestroyRegion(*r*)
Region *r*;

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

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

Status XDeviceBell(*display, device, feedbackclass, feedbackid, percent*)
Display **display*;
XDevice **device*;
XID *feedbackclass*;
XID *feedbackid*;
int *percent*;

void 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*;

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

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

char *XDisplayName(*string*)
char **string*;

DisplayOfCCC(*ccc*)
XcmsCCC *ccc*;

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

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*;

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

XextData **XEHeadOfExtensionList(*object*)
    XEDataObject *object*;

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

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

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

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

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

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

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*;

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

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

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

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

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

Bool **XFilterEvent**(*event, w*)
XEvent **event*;
Window *w*;

int **XFindContext**(*display, rid, context, data_return*)
Display **display*;
XID *rid*;
XContext *context*;
XPointer **data_return*;

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

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

void **XFlushGC**(*display, gc*)
Display **display*;
GC *gc*;

int **XFontsOfFontSet**(*font_set, font_struct_list_return, font_name_list_return*)
XFontSet *font_set*;
XFontStruct ****font_struct_list_return*;
char ****font_name_list_return*;

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

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

void **XFreeColormap**(*display, colormap*)
Display **display*;
Colormap *colormap*;

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

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

void XFreeDeviceList(*list*)
    XDeviceInfo **list*;

void XFreeExtensionList(*list*)
      char * **list*;

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

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

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

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

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

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

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

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

void XFreeStringList(*list*)
    char ***list*;

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

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

Status XGetCommand(*display, w, argv_return, argc_return*)
Display **display*;
Window *w*;
char ****argv_return*;
int **argc_return*;

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

int XGetDeviceButtonMapping(*display, device, map_return, nmap*)
Display **display*;
XDevice **device*;
unsigned *char map_return[]*;
int *nmap*;

XDeviceControl *XGetDeviceControl(*display, device, control*)
Display **display*;
XDevice **device*;
int **controlType*;

unsigned XEventClass *XGetDeviceDontPropagateList(*display, window, count*)
Display **display*;
Window *window*;
int **count*;

int XGetDeviceFocus(*display, device, focus_return, revert_to_return, time_return*)
Display **display, *device*;
Window **focus_return*;
int **revert_to_return, *time_return*;

KeySym *XGetDeviceKeyMapping(*display, device, first_keycode, keycode_count, keysyms_per_keycode_return*)
Display **display*;
XDevice **device*;
KeyCode *first_keycode*;
int *keycode_count, *keysyms_per_keycode_return*;

XModifierKeymap *XGetDeviceModifierMapping(*display, device*)
Display **display*;
XDevice **device*;

XDeviceTimeCoord *XGetDeviceMotionEvents(*display, device, start, stop, nevents_return, mode_return, axis_count_return*)
Display **display*;
XDevice **device*;
Time *start, stop*;
int **nevents_return, *mode_return, *axis_count_return*;

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

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

XExtensionVersion *XGetExtensionVersion(*display, name*)
Display **display*;
char **name*;

XFeedbackState *XGetFeedbackControl(*display, device, num_feedbacks*)
Display **display*;
XDevice **device*;
int **num_feedbacks*;

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 XGetGCValues(*display, gc, valuemask, values_return*)
Display **display*;
GC *gc*;
unsigned long *valuemask*;
XGCValues **values_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*;

char * XGetICValues(*ic, ... *)
XIC *ic*;

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

char *XGetIMValues(*im, ... *)
XIM *im*;

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

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

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

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

void 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*;

Status XGetRGBColormaps(*display, w, std_colormap_return, count_return, property*)
Display **display*;
Window *w*;
XStandardColormap ***std_colormap_return*;
int **count_return*;
Atom *property*;

void 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*;

int XGetSelectedExtensionEvents(*display, w, this_client_event_count_return, this_client_event_list_return, all_clients_event_count_return, all_clients_event_list_return*)
Display **display*;
Window *w*;
int **this_client_event_count_return*;
int **all_clients_event_count_return*;
XEventClass ***this_client_event_list_return*;
XEventClass ***all_clients_event_list_return*;

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

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

Status XGetTextProperty(*display, w, text_prop_return, property*)
Display **display*;
Window *w*;
XTextProperty **text_prop_return*;
Atom *property*;

Status XGetTransientForHint(*display, w, prop_window_return*)
Display **display*;
Window *w, *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*;
long *long_offset, long_length*;
Bool *delete*;
Atom *req_type*;
Atom **actual_type_return*;
int **actual_format_return*;
unsigned long **nitems_return, *bytes_after_return*;
unsigned char ***prop_return*;

Status XGetWMClientMachine(*display, w, text_prop_return*)
Display **display*;
Window *w*;
XTextProperty **text_prop_return*;

Status XGetWMColormapWindows(*display, w, colormap_windows_return, count_return*)
Display **display*;
Window *w, **colormap_windows_return*;
int **count_return*;

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

Status XGetWMIconName(*display, w, text_prop_return*)
Display **display*;
Window *w*;
XTextProperty **text_prop_return*;

Status XGetWMName(*display, w, text_prop_return*)
Display **display*;
Window *w*;
XTextProperty **text_prop_return*;

Status XGetWMNormalHints(*display, w, hints_return, supplied_return*)
Display **display*;
Window *w*;
XSizeHints **hints_return*;
long **supplied_return*;

Status XGetWMProtocols(*display, w, protocols_return, count_return*)
Display **display*;
Window *w*;
Atom ***protocols_return*;
int **count_return*;

Status XGetWMSizeHints(*display, w, hints_return, supplied_return, property*)
Display **display*;
Window *w*;
XSizeHints **hints_return*;
long **supplied_return*;
Atom *property*;

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

int XGrabDevice(*display, device, grab_window, owner_events, event_count, event_list, this_device_mode, other_devices_mode, time*)
Display **display*;
XDevice **device*;
Window *grab_window*;
Bool *owner_events*;
int *event_count, this_device_mode, other_devices_mode*;
XEventClass **event_list*;
Time *time*;

int XGrabDeviceButton(*display, device, button, modifiers, modifier_device, grab_window, owner_events, event_count, event_list, this_device_mode, other_devices_mode*)
Display **display*;
XDevice **device, *modifier_device*;
unsigned int *button, modifiers, event_count*;
Window *grab_window*;
Bool *owner_events*;
XEventClass **event_list*;
int *this_device_mode, other_devices_mode*;

int XGrabDeviceKey(*display, device, Key, modifiers, modifier_device, grab_window, owner_events, event_count, event_list, this_device_mode, other_devices_mode*)
Display **display*;
XDevice **device, *modifier_device*;
unsigned int *Key, modifiers, event_count*;
Window *grab_window*;
Bool *owner_events*;
XEventClass *event_list*;
int *this_device_mode, other_devices_mode*;

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

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*;
Bool *owner_events*;
unsigned int *event_mask*;
int *pointer_mode, keyboard_mode*;
Window *confine_to*;
Cursor *cursor*;
Time *time*;

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

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

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

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

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

int 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*;

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

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

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

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

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

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

XFontStruct *XHPGet16bitMixedFontStruct(*font*)
    Font *font*;

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

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

int 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 *XHPGetDeviceModifierMapping(*display, deviceid*)
Display **display*;
XID *deviceid*;

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

PtrFuncInt XHPGetEurasianCvt(*display*)
Display **display*;

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

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

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

int 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*;

int 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*;

int 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*;

int 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*;

int 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*;

int 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*;

int 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*;

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

XHPRefreshKeyboardMapping(*event_map*)
    XMappingEvent **event_map*;

KeySym XHPRoman8ToKeysym(*r8_char*)
    char *r8_char*;

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

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

int 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*;

XHPSetKbdMapInit(*display, kbd_id, force_read, status_in_out*)
    Display **display*;
    KEYBOARD_ID *kbd_id*;
    int *force_read*;
    XComposeStatus **status_in_out*;

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

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

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

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

Status XIconifyWindow(*display, w, screen_number*)
Display **display*;
Window *w*;
int *screen_number*;

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

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

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

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

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

void XInstallColormap(*display, colormap*)
Display **display*;
Colormap *colormap*;

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

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

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

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

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

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

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

int *XListDepths(*display, screen_number, count_return*)
Display **display*;
int *screen_number*;
int **count_return*;

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

XDeviceInfo *XListInputDevices(*display, ndevices_return*)
Display *display*;
int *ndevices_return*;

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

XPixmapFormatValues *XListPixmapFormats(*display, count_return*)
Display *display*;
int *count_return*;

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

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

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

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

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

Status XLookupColor(*display, colormap, color_name, exact_def_return, screen_def_return*)
Display *display*;
Colormap *colormap*;
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*;

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

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

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

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

void **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*;

long **XMaxRequestSize**(*display*)
    Display **display*;

void **XmbDrawImageString**(*display, d, font_set, gc, x, y string, num_bytes*)
    Display **display*
    Drawable *d*;
    XFontSet *font_set*;
    GC *gc*;
    int *x, y*;
    char **string*;
    int *num_bytes*;

void XmbDrawString(*display, d, font_set, gc, x, y, string, num_bytes*)
Display **display*;
Drawable *d*;
XFontSet *font_set*;
GC *gc*;
int *x, y*;
char **string*;
int *num_bytes*;

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

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

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

void XmbSetWMProperties(*display, w, window_name, icon_name, argv, argc, normal_hints, wm_hints, class_hints*)
Display **display*;
Window *w*;
char **window_name, *icon_name, *argv*[];
int *argc*;
XSizeHints **normal_hints*;
XWMHints **wm_hints*;
XClassHint **class_hints*;

int XmbTextEscapement(*font_set, string, num_bytes*)
XFontSet *font_set*;
char **string*;
int *num_bytes*;

int **XmbTextExtents**(*font_set, string, num_bytes, overall_return*)
XFontSet *font_set*;
char **string*;
int *num_bytes*;
XRectangle **overall_ink_return, *overall_logical_return*;

int **XmbTextListToTextProperty**(*display, list, count, style, text_prop_return*)
Display **display*;
char ***list*;
int *count*;
XICCEncodingStyle *style*;
XTextProperty **text_prop_return*;

Status **XmbTextPerCharExtents**(*font_set, string, num_bytes, ink_array_return, logical_array_return, array_size, num_chars_return, overall_return*)
XFontSet *font_set*;
char **string*;
int *num_bytes*;
XRectangle **ink_array_return, *logical_array_return*;
int *array_size, *um_chars_return*;
XRectangle **overall_ink_return, *overall_logical_return*;

int **XmbTextPropertyToTextList**(*display, text_prop, list_return, count_return*)
Display **display*;
XTextProperty **text_prop*;
char ****list_return*;
int **count_return*;

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

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

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

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

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

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

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

XDevice *XOpenDevice(*display, device_id*)
    Display **display*;
    XID *device_id*;

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

XIM XOpenIM(*display, db, res_name, res_class*)
    Display **display*;
    XrmDatabase *db*;
    char **res_name*;
    char **res_class*;

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

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

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

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

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

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 XProtocolVersionR(*display*)
    Display **display*;

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

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

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, *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*;

void XQueryColor(*display, colormap, def_in_out*)
Display **display*;
Colormap *colormap*;
XColor **def_in_out*;

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

XDeviceState *XQueryDeviceState(*display, device*)
Display **display*;
XDevice **device*;

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*;
int **first_error_return*;

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

void 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, *win_x_return, *win_y_return*;
unsigned int **mask_return*;

void **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*;

void **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*;

void **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*;

void **XRebindKeysym**(*display, keysym, list, mod_count, string, num_bytes*)
Display **display*;
KeySym *keysym, *list*;
int *mod_count*;
unsigned char **string*;
int *num_bytes*;

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

Status XReconfigureWMWindow(*display, w, screen_number, value_mask, values*)
Display **display*;
Window *w*;
int *screen_number*;
unsigned int *value_mask*;
XWindowChanges **values*;

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

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

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

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

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

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

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

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

char *XResourceManagerString(*display*)
Display **display*;

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

void XrmCombineDatabase(*source_db, target_db, override*)
XrmDatabase *source_db, *target_db*;
Bool *override*;

void XrmCombineFileDatabase(*filename, target_db, override*)
char **filename*;
XrmDatabase **target_db*;
Bool *override*;

void XrmDestroyDatabase(*database*)
XrmDatabase *database*;

Bool XrmEnumerateDatabase(*database, name_prefix, class_prefix, mode, proc, arg*)
XrmDatabase *database*;
XrmNameList *name_prefix*;
XrmClassList *class_prefix*;
int *mode*;
Bool (**proc*)();
XPointer *arg*;

XrmDatabase XrmGetDatabase(*display*)
Display **display*;

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

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

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

void XrmInitialize();

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

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

XrmQuark **XrmPermStringToQuark**(*string*)
char **string*;

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_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*;

XrmString *XrmQuarkToString(*quark*)
XrmQuark *quark*;

void XrmSetDatabase(*display, database*)
Display **display*;
XrmDatabase *database*;

void 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*;

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

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

int XSaveContext(*display, rid, context, data*)
Display **display*;
XID *rid*;
XContext *context*;
XPointer *data*;

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

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

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

char *XScreenResourceString(*screen*)
Screen **screen*;

int XSelectExtensionEvent(*display, w, event_list, event_count*)
Display **display*;
Window *w*;
XEventClass **event_list*;
int *event_count*;

void 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*;

Status XSendExtensionEvent(*display, device, destination, propagate, event_count, event_list, event_send*)
Display **display*;
XDevice **device*;
Window *destination*;
Bool *propagate*;
int *event_count*;
XEventClass **event_list*;
XEvent **event_send*;

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

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

int (*XSetAfterFunction(*display, procedure*))()
Display **display*;
int (**procedure*)();

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

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

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

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

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

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

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

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

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

int XSetDeviceButtonMapping(*display, device, map, nmap*)
Display **display*;
XDevice **device*;
unsigned char *map[]*;
int *nmap*;

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

int XSetDeviceMode(*display, device, mode*)
Display **display*;
XDevice **device*;
int *mode*;

int XSetDeviceModifierMapping(*display, device, modmap*)
Display **display*;
XDevice **device*;
XModifierKeymap **modmap*;

int XSetDeviceValuators(*display, device, valuators, first_valuator, num_valuators*)
    Display **display*;
    XDevice **device*;
    int **valuators*;
    int *first_valuator*;
    int *num_valuators*;

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

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

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

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

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

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

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

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

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

char * XSetICValues(*ic, ... *)
    XIC *ic*;

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

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

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

int(*XSetIOErrorHandler(*handler*))()
int(**handler*)

void 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*;

char *XSetLocaleModifiers(*modifier_list*)
char **modifier_list*;

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

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

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

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

void XSetRGBColormaps(*display, w, std_colormap, count, property*)
Display **display*;
Window *w*;
XStandardColormap **std_colormap*;
int *count*;
Atom *property*;

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

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

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

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

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

void XSetTextProperty(*display, w, text_prop, property*)
Display **display*;
Window *w*;
XTextProperty **text_prop*;
Atom *property*;

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

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

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

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

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

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

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

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

void XSetWindowColormap(*display, w, colormap*)
Display **display*;
Window *w*;
Colormap *colormap*;

void XSetWMClientMachine(*display, w, text_prop*)
Display **display*;
Window *w*;
XTextProperty **text_prop*;

Status XSetWMColormapWindows(*display, w, colormap_windows, count*)
Display **display*;
Window *w, *colormap_windows*;
int *count*;

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

void XSetWMIconName(*display, w, text_prop*)
Display **display*;
Window *w*;
XTextProperty **text_prop*;

void XSetWMName(*display, w, text_prop*)
Display **display*;
Window *w*;
XTextProperty **text_prop*;

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

void XSetWMProperties(*display, w, window_name, icon_name, argv, argc, normal_hints, wm_hints, class_hints*)
Display **display*;
Window *w*;
XTextProperty **window_name, *icon_name*;
char ***argv*;
int *argc*;
XSize_hints **normal_hints*;
XWMHints **wm_hints*;
XClassHint **class_hints*;

Status XSetWMProtocols(*display, w, protocols, count*)
Display **display*;
Window *w*;
Atom **protocols*;
int *count*;

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

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

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

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

void XStoreColor(*display, colormap, color*)
    Display **display*;
    Colormap *colormap*;
    XColor **color*;

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

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

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

Status XStringListToTextProperty(*list, count, text_prop_return*)
    char ***list*;
    int *count*;
    XTextProperty **text_prop_return*;

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*;

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

Bool XSupportsLocale()

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

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

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

void 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*;

Status XTextPropertyToStringList(*text_prop, list_return, count_return*)
XTextProperty **text_prop*;
char ****list_return*;
int **count_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*;
int *src_x, src_y, *dest_x_return, *dest_y_return*;
Window **child_return*;

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

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

int XUngrabDevice(*display, device, time*)
Display **display*;
XDevice **device*;
Time *time*;

void XUngrabDeviceButton(*display, device, button, modifiers, modfier_device, grab_window*)
Display **display*;
XDevice **device, *modifier_device*;
unsigned int *button, modifiers*;
Window *grab_window*;

int XUngrabDeviceKey(*display, device, Key, modifiers, modifier_device, grab_window*)
Display **display*;
XDevice **device, *modifier_device*;
unsigned int *Key, modifiers*;
Window *grab_window*;

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

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

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

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

void XUninstallColormap(*display, colormap*)
Display **display*;
Colormap *colormap*;

void XUnionRectWithRegion(*rectangle, src_region, dest_region_return*)
XRectangle **rectangle*;
Region *src_region, dest_region_return*;

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

XContext XUniqueContext()

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

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

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

void XUnsetICFocus(*ic*)
XIC *ic*;

XVaNestedList XVaCreateNestedList(*dummy, ...*) int *dummy*;

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

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

void 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*;
unsigned int *src_width, src_height*;
int *dest_x, dest_y*;

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

void XwcDrawString(*display, d, font_set, gc, x, y, string, num_wchars*)
Display **display*;
Drawable *d*;
XFontSet *font_set*;
GC *gc*;
int *x, y, num_wchars*;
wchar_t **string*;

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

void **XwcFreeStringList**(*list*)
wchar_t ***list*;

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

wchar_t * **XwcResetIC**(*ic*)
XIC *ic*;

int **XwcTextEscapement**(*font_set, string, num_wchars*)
XFontSet *font_set*;
wchar_t **string*;
int *num_wchars*;

int **XwcTextExtents**(*font_set, string, num_wchars, overall_return*)
XFontSet *font_set*;
wchar_t **string*;
int *num_wchars*;
XRectangle **overall_ink_return*;
XRectangle **overall_logical_return*;

int **XwcTextListToTextProperty**(*display, list, count, style, text_prop_return*)
Display **display*;
wchar_t ***list*;
int *count*;
XICCEncodingStyle *style*;
XTextProperty **text_prop_return*;

Status **XwcTextPerCharExtents**(*font_set, string, num_wchars, ink_array_return, logical_array_return, array_size, num_chars_return, overall_return*)
XFontSet *font_set*;
wchar_t **string*;
int *num_wchars*;
XRectangle **ink_array_return, *logical_array_return*;
int *array_size, *num_chars_return*;
XRectangle **overall_ink_return, *overall_logical_return*;

int **XwcTextPropertyToTextList**(*display, text_prop, list_return, count_return*)
Display **display*;
XTextProperty **text_prop*;
wchar_t ****list_return*;
int **count_return*;

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

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

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

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

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

Status **XWithdrawWindow**(*display, w, screen_number*)
Display **display*;
Window *w*;
int *screen_number*;

int **XWMGeometry**(*display, screen, user_geom, def_geom, bwidth, hints, x_return, y_return, width_return, height_return, gravity_return*)
Display **display*;
int *screen*;
char **user_geom, *def_geom*;
unsigned int *bwidth*;
XSizeHints **hints*;
int **x_return, *y_return, *width_return*;
int **height_return, *gravity_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

X11 R5 Xlib Macros

Header files for these macros can be found in `Xcms.h`, `Xlib.h`, or `xutil.h`.

`AllPlanes`

`BitmapBitOrder`(*display*)
    Display **display*;

`BitmapPad`(*display*)
    Display **display*;

`BitmapUnit`(*display*)
    Display **display*;

`BlackPixel`(*display, screen_number*)
    Display **display*;
    int *screen_number*;

`BlackPixelOfScreen`(*screen*)
    Screen **screen*;

`CellsOfScreen`(*screen*)
    Screen **screen*;

`ClientWhitePointOfCCC`(*ccc*)
    XcmsCCC *ccc*;

`ConnectionNumber`(*display*)
    Display **display*;

`DefaultColormap`(*display, screen_number*)
    Display **display*;
    int *screen_number*;

`DefaultColormapOfScreen`(*screen*)
    Screen **screen*;

`DefaultDepth`(*display, screen_number*)
    Display **display*;
    int *screen_number*;

`DefaultDepthOfScreen`(*screen*)
    Screen **screen*;

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

DefaultGCOfScreen(*screen*)
    Screen **screen*;

DefaultRootWindow(*display*)
    Display **display*;

DefaultScreen(*display*)
    Display **display*;

DefaultScreenOfDisplay(*display*)
    Display **display*;

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

DefaultVisualOfScreen(*screen*)
    Screen **screen*;

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

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

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

DisplayOfCCC(*ccc*)
    XcmsCCC *ccc*;

DisplayOfScreen(*screen*)
    Screen **screen*;

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

DisplayString(*display*)
    Display **display*;

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

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

DoesBackingStore(*screen*)
    Screen **screen*;

DoesSaveUnders(*screen*)
    Screen **screen*;

EventMaskOfScreen(*screen*)
    Screen **screen*;

FunctionSetOfCCC(*ccc*)
    XcmsCCC *ccc*;

HeightMMOfScreen(*screen*)
    Screen **screen*;

HeightOfScreen(*screen*)
    Screen **screen*;

ImageByteOrder(*display*)
    Display **display*;

IsCursorKey(*keysym*)
    KeySym *keysym*;

IsFunctionKey(*keysym*)
    KeySym *keysym*;

IsKeypadKey(*keysym*)
    KeySym *keysym*;

IsMiscFunctionKey(*keysym*)
    KeySym *keysym*;

IsModifierKey(*keysym*)
    KeySym *keysym*;

IsPFKey(*keysym*)
    KeySym *keysym*;

LastKnownRequestProcessed(*display*)
    Display **display*;

MaxCmapsOfScreen(*screen*)
    Screen **screen*;

MinCmapsOfScreen(*screen*)
    Screen **screen*;

NextRequest(*display*)
    Display **display*;

PlanesOfScreen(*screen*)
    Screen **screen*;

ProtocolRevision(*display*)
    Display **display*;

ProtocolVersion(*display*)
    Display **display*;

QLength(*display*)
    Display **display*;

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

RootWindowOfScreen(*screen*)
    Screen **screen*;

ScreenCount(*display*)
    Display **display*;

ScreenNumberOfCCC(*ccc*)
    XcmsCCC *ccc*;

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

ScreenWhitePointOfCCC(*ccc*)
    XcmsCCC *ccc*;

ServerVendor(*display*)
    Display **display*;

VendorRelease(*display*)
    Display **display*;

VisualOfCCC(*ccc*)
    XcmsCCC *ccc*;

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

WhitePixelOfScreen(*screen*)
    Screen **screen*;

WidthMMOfScreen(*screen*)
    Screen **screen*;

WidthOfScreen(*screen*)
    Screen **screen*;

4

OSF/Motif 1.2 Functions

Cardinal `MrmCloseHierarchy`(*hierarchy_id*)
MrmHierarchy *hierarchy_id*;

Cardinal `MrmFetchBitmapLiteral`(*hierarchy_id, index, screen, display, pixmap_return, width, height*)
MrmHierarchy *hierarchy_id*;
String *index*;
Screen **screen*;
Display **display*;
Pixmap **pixmap_return*;
Dimension **width* **height*;

int `MrmFetchColorLiteral`(*hierarchy_id, index, display, colormap_id, pixel*)
MrmHierarchy *hierarchy_id*;
String *index*;
Display **display*;
Colormap *colormap_id*;
Pixel **pixel*;

int `MrmFetchIconLiteral`(*hierarchy_id, index, screen, display, fgpix, bgpix, pixmap*)
MrmHierarchy *hierarchy_id*;
String *index*;
Screen **screen*;
Display **display*;
Pixel *fgpix, bgpix*;
Pixmap **pixmap*;

int `MrmFetchLiteral`(*hierarchy_id, index, display, value, type*)
MrmHierarchy *hierarchy_id*;
String *index*;
Display **display*;
XtPointer **value*;
MrmCode **type*;

Cardinal **MrmFetchSetValues**(*hierarchy_id, widget, args, num_args*)
MrmHierarchy *hierarchy_id*;
Widget *widget*;
ArgList *args*;
Cardinal *num_args*;

Cardinal **MrmFetchWidget**(*hierarchy_id, index, parent_widget, widget, class*)
MrmHierarchy *hierarchy_id*;
String *index*;
Widget *parent_widget, *widget*;
MrmType *class*;

Cardinal **MrmFetchWidgetOverride**(*hierarchy_id, index, parent_widget, override_name, override_args, override_num_args, widget, class*)
MrmHierarchy *hierarchy_id*;
String *index, override_name*;
Widget *parent_widget, *widget*;
ArgList *override_args*;
Cardinal *override_num_args*;
MrmType *class*;

void MrmInitialize()

Cardinal **MrmOpenHierarchy**(*num_files, file_names_list, ancillary_structures_list, hierarchy_id*)
MrmCount *num_files*;
String *file_names_list []*;
MrmOsOpenParamPtr *ancillary_structures_list*;
MrmHierarchy *hierarchy_id*;

Cardinal **MrmOpenHierarchyPerDisplay** (*display, num_files, file_names_list, ancillary_structures_list, hierarchy_id*)
Display *display*;
MrmCount *num_files*;
String *file_names_list*[];
MrmOsOpenParamPtr *ancillary_structures_list*;
MrmHierarchy *hierarchy_id*;

Cardinal **MrmRegisterClass**(*class_code, class_name, create_name, create_proc, class_record*)
MrmType *class_code*;
String *class_name, create_name*;
Widget *(*create_proc) ()*;
WidgetClass *class_record*);

Cardinal **MrmRegisterNames**(*register_list, register_count*)
MrmRegisterArglist *register_list*;
MrmCount *register_count*;

Cardinal **MrmRegisterNamesInHierarchy**(*hierarchy_id, register_list, register_count*)
MrmHierarchy *hierarchy_id*;
MrmRegisterArglist *register_list*;
MrmCount *register_count*;

Uil_status_type **Uil**(*command_desc, compile_desc, message_cb, message_data, status_cb, status_data*)
Uil_command_type **command_desc*;
Uil_compile_desc_type **compile_desc*;
Uil_continue_type *(*message_cb) ()*;
char **message_data*;
Uil_continue_type *(status_cb) ()*;
char **status_data*;

void **UilDumpSymbolTable**(*root_ptr*)
sym_entry_type **root_ptr*;

void **XmActivateProtocol**(*shell, property, protocol*)
Widget *shell*;
Atom *property, protocol*;

void **XmActivateWMProtocol**(*shell, protocol*)
Widget *shell*;
Atom *protocol*;

void **XmAddProtocolCallback**(*shell, property, protocol, callback, closure*)
Widget *shell*;
Atom *property, protocol*;
XtCallbackProc *callback*;
XtPointer *closure*;

void **XmAddProtocols**(*shell, property, protocols, num_protocols*)
Widget *shell*;
Atom *property, *protocols*;
Cardinal *num_protocols*;

void **XmAddTabGroup**(*tab_group*)
Widget *tab_group*;

void **XmAddWMProtocolCallback**(*shell, protocol, callback, closure*)
Widget *shell*;
Atom *protocol*;
XtCallbackProc *callback*;
XtPointer *closure*;

void **XmAddWMProtocols**(*shell, protocols, num_protocols*)
Widget *shell*;
Atom **protocols*;
Cardinal *num_protocols*;

void XmCascadeButtonGadgetHighlight(*cascadeButtonGadget, highlight*)
Widget *cascadeButtonGadget*;
Boolean *highlight*;

void XmCascadeButtonHighlight(*cascadeButton, highlight*)
Widget *cascadeButton*;
Boolean *highlight*;

void XmChangeColor (*widget, background*)
Widget *widget*;
Pixel *background*;

int XmClipboardCancelCopy(*display, window, item_id*)
Display **display*;
Window *window*;
long *item_id*;

int XmClipboardCopy(*display, window, item_id, format_name, buffer, length, private_id, data_id*)
Display **display*;
Window *window*;
long *item_id*;
char **format_name*;
XtPointer **buffer*;
unsigned long *length*;
long *private_id* **data_id*;

int XmClipboardCopyByName(*display, window, data_id, buffer, length, private_id*)
Display **display*;
Window *window*;
long *data_id, private_id*;
XtPointer *buffer*;
unsigned long *length*;

int XmClipboardEndCopy(*display, window, item_id*)
Display **display*;
Window *window*;
long *item_id*;

int XmClipboardEndRetrieve(*display, window*)
Display **display*;
Window *window*;

int XmClipboardInquireCount(*display, window, count, max_format_name_length*)
Display **display*;
Window *window*;
int **count*;
unsigned long **max_format_name_length*;

int XmClipboardInquireFormat(*display, window, index, format_name_buf, buffer_len, copied_len*)
Display **display*;
Window *window*;
int *index*;
XtPointer *format_name_buf*;
unsigned long *buffer_len, *copied_len*;

int XmClipboardInquireLength(*display, window, format_name, length*)
Display **display*;
Window *window*;
char **format_name*;
unsigned long **length*;

int XmClipboardInquirePendingItems(*display, window, format_name, item_list, count*)
Display **display*;
Window *window*;
char **format_name*;
XmClipboardPendingList **item_list*;
unsigned long **count*;

int XmClipboardLock(*display, window*)
Display **display*;
Window *window*;

int XmClipboardRegisterFormat(*display, format_name, format_length*)
Display **display*;
char **format_name*;
int *format_length*;

int XmClipboardRetrieve(*display, window, format_name, buffer, length, num_bytes, private_id*)
Display **display*;
Window *window*;
char **format_name*;
XtPointer *buffer*;
unsigned long *length, *num_bytes*;
int **private_id*;

int **XmClipboardStartCopy**(*display, window, clip_label, timestamp, widget, callback, item_id*)
Display **display*;
Window *window*;
XmString *clip_label*;
Time *timestamp*;
Widget *widget*;
XmCutPasteProc *callback*;
long **item_id*;

int **XmClipboardStartRetrieve**(*display, window, timestamp*)
Display **display*;
Window *window*;
Time *timestamp*;

int **XmClipboardUndoCopy**(*display, window*)
Display **display*;
Window *window*;

int **XmClipboardUnlock**(*display, window, remove_all_locks*)
Display **display*;
Window *window*;
Boolean *remove_all_locks*;

int **XmClipboardWithdrawFormat**(*display, window, data_id*)
Display **display*;
Window *window*;
int *data_id*;

void **XmCommandAppendValue**(*widget, command*)
Widget *widget*;
XmString *command*;

void **XmCommandError**(*widget, error*)
Widget *widget*;
XmString *error*;

Widget **XmCommandGetChild**(*widget, child*)
Widget *widget*;
unsigned char *child*;

void **XmCommandSetValue**(*widget, command*)
Widget *widget*;
XmString *command*;

int **XmConvertUnits**(*widget, orientation, from_unit_type, from_value, to_unit_type*)
Widget *widget*;
int *orientation, from_unit_type, from_value, to_unit_type*;

Widget XmCreateArrowButton(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateArrowButtonGadget(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateBulletinBoard(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateBulletinBoardDialog(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateCascadeButton(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateCascadeButtonGadget(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateCommand(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateDialogShell(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateDragIcon (*widget, name, arglist, argcount*)
Widget *widget*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

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

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

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

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

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

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

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

Widget XmCreateFrame(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateInformationDialog(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateLabel(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateLabelGadget(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateList(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateMainWindow(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateMenuBar(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateMenuShell(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateMessageBox(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateMessageDialog(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateOptionMenu(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreatePanedWindow(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreatePopupMenu(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreatePromptDialog(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreatePulldownMenu(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreatePushButton(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreatePushButtonGadget(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateQuestionDialog(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateRadioBox(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateRowColumn(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateScale(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateScrollBar(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateScrolledList(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateScrolledText(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateScrolledWindow(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSelectionBox(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSelectionDialog(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSeparator(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSeparatorGadget(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSimpleCheckBox(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSimpleMenuBar(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

Widget XmCreateSimpleOptionMenu(*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

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

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

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

Widget XmCreateTemplateDialog (*parent, name, arglist, argcount*)
Widget *parent*;
String *name*;
ArgList *arglist*;
Cardinal *argcount*;

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

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

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

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

Widget **XmCreateWarningDialog**(*parent, name, arglist, argcount*)
    Widget *parent*;
    String *name*;
    ArgList *arglist*;
    Cardinal *argcount*;

Widget **XmCreateWorkArea**(*parent, name, arglist, argcount*)
    Widget *parent*;
    String *name*;
    ArgList *arglist*;
    Cardinal *argcount*;

Widget **XmCreateWorkingDialog**(*parent, name, arglist, argcount*)
    Widget *parent*;
    String *name*;
    ArgList *arglist*;
    Cardinal *argcount*;

XmString **XmCvtCTToXmString**(*text*)
    char **text*;

void **XmCvtStringToUnitType**(*args, num_args, from_val, to_val*)
    XrmValuePtr *args*;
    Cardinal **num_args*;
    XrmValue **from_val, *to_val*;

char ***XmCvtXmStringToCT**(*string*)
    XmString *string*;

void **XmDeactivateProtocol**(*shell, property, protocol*)
    Widget *shell*;
    Atom *property, protocol*;

void **XmDeactivateWMProtocol**(*shell, protocol*)
    Widget *shell*;
    Atom *protocol*;

Boolean **XmDestroyPixmap**(*screen, pixmap*)
    Screen **screen*;
    Pixmap *pixmap*;

void **XmDragCancel** (*dragcontext*)
    Widget *dragcontext*;

Widget **XmDragStart** (*widget, event, arglist, argcount*)
    Widget *widget*;
    XEvent **event*;
    ArgList *arglist*;
    Cardinal *argcount*;

void **XmDropSiteConfigureStackingOrder** (*widget, sibling, stack_mode*)
Widget *widget, sibling*;
Cardinal *stack_mode*;

void **XmDropSiteEndUpdate** (*widget*)
Widget *widget*;

Status **XmDropSiteQueryStackingOrder** (*widget, parent_return, child_returns, num_child_returns*)
Widget *widget, *parent_return, **child_returns*;
Cardinal **num_child_returns*;

void **XmDropSiteRegister** (*widget, arglist, argcount*)
Widget *widget*;
ArgList *arglist*;
Cardinal *argcount*;

void **XmDropSiteRetrieve** (*widget, arglist, argcount*)
Widget *widget*;
ArgList *arglist*;
Cardinal *argcount*;

void **XmDropSiteStartUpdate** (*widget*)
Widget *widget*;

void **XmDropSiteUnregister** (*widget*)
Widget *widget*;

void **XmDropSiteUpdate** (*widget, arglist, argcount*)
Widget *widget*;
ArgList *arglist*;
Cardinal *argcount*;

void **XmDropTransferAdd** (*drop_transfer, transfers, num_transfers*)
Widget *drop_transfer*;
XmDropTransferEntryRec **transfers*;
Cardinal *num_transfers*;

Widget **XmDropTransferStart** (*widget, arglist, argcount*)
Widget *widget*;
ArgList *arglist*;
Cardinal *argcount*;

Widget **XmFileSelectionBoxGetChild**(*widget, child*)
Widget *widget*;
unsigned char *child*;

void **XmFileSelectionDoSearch**(*widget, dirmask*)
Widget *widget*;
XmString *dirmask*;

XmFontList **XmFontListAdd**(*oldlist, font, charset*)
XmFontList *oldlist*;
XFontStruct **font*;
XmStringCharSet *charset*;

XmFontList **XmFontListAppendEntry** (*oldlist, entry*)
XmFontList *oldlist*;
XmFontListEntry *entry*;

XmFontList **XmFontListCopy**(*fontlist*)
XmFontList *fontlist*;

XmFontList **XmFontListCreate**(*font, charset*)
XFontStruct **font*;
XmStringCharSet *charset*;

XmFontListEntry **XmFontListEntryCreate** (*tag, type, font*)
char **tag*;
XmFontType *type*;
XtPointer *font*;

void **XmFontListEntryFree** (*entry*)
XmFontListEntry **entry*;

XtPointer **XmFontListEntryGetFont** (*entry, type_return*)
XmFontListEntry *entry*;
XmFontType **type_return*;

char* **XmFontListEntryGetTag** (*entry*)
XmFontListEntry *entry*;

XmFontListEntry **XmFontListEntryLoad** (*display, font_name, type, tag*)
Display **display*;
char **font_name, *tag*;
XmFontType *type*;

void **XmFontListFree**(*list*)
XmFontList *list*;

void **XmFontListFreeFontContext**(*context*)
XmFontContext *context*;

Boolean **XmFontListGetNextFont**(*context, charset, font*)
XmFontContext *context*;
XmStringCharSet **charset*;
XFontStruct ***font*;

Boolean **XmFontListInitFontContext**(*context, fontlist*)
XmFontContext **context*;
XmFontList *fontlist*;

XmFontListEntry **XmFontListNextEntry** (*context*)
XmFontContext *context*;

XmFontList **XmFontListRemoveEntry** (*oldlist, entry*)
XmFontList *oldlist*;
XmFontListEntry *entry*;

String **XmGetAtomName**(*display, atom*)
Display **display*;
Atom *atom*;

XmColorProc **XmGetColorCalculation**()

void **XmGetColors**(*screen, colormap, background, foreground, top_shadow, bottom_shadow, select*)
Screen **screen*;
Colormap *colormap*;
Pixel **background, *foreground, *top_shadow, *bottom_shadow*;
Pixel **select*;

Widget **XmGetDestination**(*display*)
Display **display*;

Widget **XmGetDragContext** (*refwidget, timestamp*)
Widget *refwidget*;
Time *timestamp*;

Widget **XmGetFocusWidget** (*widget*)
Widget *widget*;

Cursor **XmGetMenuCursor**(*display*)
Display **display*;

Pixmap **XmGetPixmap**(*screen, image_name, foreground, background*)
Screen **screen*;
char **image_name*;
Pixel *foreground, background*;

Pixmap **XmGetPixmapByDepth** (*screen, image_name,foreground, background, depth*)
Screen **screen*;
char **image_name*;
Pixel *foreground, background*;
int *depth*;

Widget **XmGetPostedFromWidget**(*menu*)
Widget *menu*;

Cardinal **XmGetSecondaryResourceData** (*widget_class, secondary_data_return*)
WidgetClass *widget_class*;
XmSecondaryResourceData ***secondary_data_return*;

Widget **XmGetTabGroup** (*widget*)
Widget *widget*;

Widget **XmGetTearOffControl** (*menu*)
    Widget *menu*;

XmVisibility **XmGetVisibility** (*widget*)
    Widget *widget*;

Widget **XmGetXmDisplay** (*display*)
    Display **display*;

Widget **XmGetXmScreen** (*screen*)
    Screen **screen*;

XIM **XmImGetXIM**(widget)
    Widget *widget*;

int **XmImMbLookupString** (*widget, event, buffer_return, bytes_buffer, keysym_return, status_return*)
    Widget *widget*;
    XKeyPressedEvent **event*;
    char **buffer_return*;
    int *bytes_buffer*;
    KeySym **keysym_return*;
    Status **status_return*;

void **XmImRegister** (*widget*)
    Widget *widget*;

void **XmImSetFocusValues** (*widget, arglist, argcount*)
    Widget *widget*;
    ArgList *arglist*;
    Cardinal *argcount*;

void **XmImSetValues** (*widget, arglist, argcount*)
    Widget *widget*;
    ArgList *arglist*;
    Cardinal *argcount*;

void **XmImUnregister** (*widget*)
    Widget *widget*;

void **XmImUnsetFocus** (*widget*)
    Widget *widget*;

void **XmImVaSetFocusValues** (*widget, arg ...*)
    Widget *widget*;

void **XmImVaSetValues** (*widget, arg ...*)
    Widget *widget*;

Boolean **XmInstallImage**(*image, image_name*)
    XImage **image*;
    char **image_name*;

Atom XmInternAtom(*display, name, only_if_exists*)
Display **display*;
String *name*;
Boolean *only_if_exists*;

Boolean XmIsMotifWMRunning(*shell*)
Widget *shell*;

Boolean XmIsTraversable (*widget*)
Widget *widget*;

void XmListAddItem(*widget, item, position*)
Widget *widget*;
XmString *item*;
int *position*;

void XmListAddItems(*widget, items, item_count, position*)
Widget *widget*;
XmString **items*;
int *item_count, position*;

void XmListAddItemsUnselected(*widget, item, position*)
Widget *widget*;
XmString *item*;
int *position*;

void XmListDeleteAllItems(*widget*)
Widget *widget*;

void XmListDeleteItem(*widget, item*)
Widget *widget*;
XmString *item*;

void XmListDeleteItems(*widget, items, item_count*)
Widget *widget*;
XmString **items*;
int *item_count*;

void XmListDeleteItemsPos(*widget, item_count, position*)
Widget *widget*;
int *item_count*;
int *position*;

void XmListDeletePos(*widget, position*)
Widget *widget*;
int *position*;

void XmListDeletePositions (*widget, position_list, position_count*)
Widget *widget*;
int **position_list, position_count*;

void **XmListDeselectAllItems**(*widget*)
Widget *widget*;

void **XmListDeselectItem**(*widget, item*)
Widget *widget*;
XmString *item*;

void **XmListDeselectPos**(*widget, position*)
Widget *widget*;
int *position*;

int **XmListGetKbdItemPos** (*widget*)
Widget *widget*;

Boolean **XmListGetMatchPos**(*widget, item, position_list, position_count*)
Widget *widget*;
XmString *item*;
int ***position_list, *position_count*;

Boolean **XmListGetSelectedPos**(*widget, position_list, position_count*)
Widget *widget*;
int ***position_list, *position_count*;

Boolean **XmListItemExists**(*widget, item*)
Widget *widget*;
XmString *item*;

int **XmListItemPos**(*widget, item*)
Widget *widget*;
XmString *item*;

Boolean **XmListPosSelected** (*widget, position*)
Widget *widget*;
int *position*;

Boolean **XmListPosToBounds** (*widget, position, x, y, width, height*)
Widget *widget*;
int *position*;
Position **x, *y*;
Dimension **width, *height*;

void **XmListReplaceItems**(*widget, old_items, item_count, new_items*)
Widget *widget*;
XmString **old_items*;
int *item_count*;
XmString **new_items*;

void **XmListReplaceItemsPos**(*widget, new_items, item_count, position*)
Widget *widget*;
XmString **new_items*;
int *item_count, position*;

void **XmListReplaceItemsPosUnselected** (*widget, new_items, item_count, position*)
Widget *widget*;
XmString **new_items*;
int *item_count, position*;

void **XmListReplaceItemsUnselected** (*widget, old_items, item_count, new_items*)
Widget *widget*;
XmString **old_items, *new_items*;
int *item_count*;

void **XmListReplacePositions** (*widget, position_list, item_list, item_count*)
Widget *widget*;
int **position_list, item_count*;
XmString **item_list*;

void **XmListSelectItem**(*widget, item, notify*)
Widget *widget*;
XmString *item*;
Boolean *notify*;

void **XmListSelectPos**(*widget, position, notify*)
Widget *widget*;
int *position*;
Boolean *notify*;

void **XmListSetAddMode**(*widget, mode*)
Widget *widget*;
Boolean *mode*;

void **XmListSetBottomItem**(*widget, item*)
Widget *widget*;
XmString *item*;

void **XmListSetBottomPos**(*widget, position*)
Widget *widget*;
int *position*;

void **XmListSetHorizPos**(*widget, position*)
Widget *widget*;
int *position*;

void **XmListSetItem**(*widget, item*)
Widget *widget*;
XmString *item*;

Boolean **XmListSetKbdItemPos** (*widget, position*)
Widget *widget*;
int *position*;

void **XmListSetPos**(*widget, position*)
Widget *widget*;
int *position*;

void **XmListUpdateSelectedList** (*widget*)
Widget *widget*;

int **XmListYToPos** (*widget, y*)
Widget *widget*;
Position *y*;

Widget **XmMainWindowSep1**(*widget*)
Widget *widget*;

Widget **XmMainWindowSep2**(*widget*)
Widget *widget*;

Widget **XmMainWindowSep3**(*widget*)
Widget *widget*;

void **XmMainWindowSetAreas**(*widget, menu_bar, command_window, horizontal_scrollbar, vertical_scrollbar, work_region*)
Widget *widget, menu_bar, command_window*;
Widget *horizontal_scrollbar, vertical_scrollbar*;
Widget *work_region*;

char * **XmMapSegmentEncoding** (*fontlist_tag*)
char **fontlist_tag*;

void **XmMenuPosition**(*menu, event*)
Widget *menu*;
XButtonPressedEvent **event*;

Widget **XmMessageBoxGetChild**(*widget, child*)
Widget *widget*;
unsigned char *child*;

Widget **XmOptionButtonGadget**(*option_menu*)
Widget *option_menu*;

Widget **XmOptionLabelGadget**(*option_menu*)
Widget *option_menu*;

Boolean **XmProcessTraversal**(*widget, direction*)
Widget *widget*;
XmTraversalDirection *direction*;

char * XmRegisterSegmentEncoding (*fontlist_tag, ct_encoding*)
    char **fontlist_tag*;
    char **ct_encoding*;

void XmRemoveProtocolCallback(*shell, property, protocol, callback, closure*)
    Widget *shell*;
    Atom *property, protocol*;
    XtCallbackProc *callback*;
    XtPointer *closure*;

void XmRemoveProtocols (*shell, property, protocols, num_protocols*)
    Widget *shell*;
    Atom *property, * protocols*;
    Cardinal *num_protocols*;

void XmRemoveTabGroup(*tab_group*)
    Widget *tab_group*;

void XmRemoveWMProtocolCallback(*shell, protocol, callback, closure*)
    Widget *shell*;
    Atom *protocol*;
    XtCallbackProc *callback*;
    XtPointer *closure*;

void XmRemoveWMProtocols (*shell, protocols, num_protocols*)
    Widget *shell*;
    Atom ** protocols*;
    Cardinal *num_protocols*;

void XmRepTypeAddReverse (*rep_type_id*)
    XmRepTypeId *rep_type_id*;

XmRepTypeId XmRepTypeGetId (*rep_type*)
    String *rep_type*;

String * XmRepTypeGetNameList (*rep_type_id, use_uppercase_format*)
    XmRepTypeId *rep_type_id*;
    Boolean *use_uppercase_format*;

XmRepTypeEntry XmRepTypeGetRecord (*rep_type_id*)
    XmRepTypeId *rep_type_id*;

XmRepTypeList XmRepTypeGetRegistered ()

void XmRepTypeInstallTearOffModelConverter ()

XmRepTypeId XmRepTypeRegister (*rep_type, value_names, values, num_values*)
    String *rep_type, *value_names*;
    unsigned char **values, num_values*;

Boolean **XmRepTypeValidValue** (*rep_type_id, test_value, enable_default_warning*)
XmRepTypeId *rep_type_id*;
unsigned char *test_value*;
Widget *enable_default_warning*;

void **XmResolveAllPartOffsets**(*widget_class, offset, constraint_offset*)
WidgetClass *widget_class*;
XmOffsetPtr **offset, **constraint_offset*;

void **XmResolvePartOffsets**(*widget_class, offset*)
WidgetClass *widget_class*;
XmOffsetPtr **offset*;

void **XmScaleGetValue**(*widget, value_return*)
Widget *widget*;
int **value_return*;

void **XmScaleSetValue**(*widget, value*)
Widget *widget*;
int *value*;

void **XmScrollBarGetValues**(*widget, value_return, slider_size_return, increment_return, page_increment_return*)
Widget *widget*;
int **value_return, *slider_size_return*;
int **increment_return, *page_increment_return*;

void **XmScrollBarSetValues**(*widget, value, slider_size, increment, page_increment, notify*)
Widget *widget*;
int *value, slider_size*;
int *increment, page_increment*;
Boolean *notify*;

void **XmScrollVisible** (*scrollw_widget, widget, left_right_margin, top_bottom_margin*)
Widget *scrollw_widget, widget*;
Dimension *left_right_margin, top_bottom_margin*;

void **XmScrolledWindowSetAreas**(*widget, horizontal_scrollbar, vertical_scrollbar, work_region*)
Widget *widget, horizontal_scrollbar*;
Widget *vertical_scrollbar, work_region*;

Widget **XmSelectionBoxGetChild**(*widget, child*)
Widget *widget*;
unsigned char *child*;

XmColorProc **XmSetColorCalculation**(*color_proc*)
XmColorProc *color_proc*;

void XmSetFontUnit(*display, font_unit_value*)
Display **display*;
int *font_unit_value*;

void XmSetFontUnits(*display, h_value, v_value*)
Display **display*;
int *h_value, v_value*;

void XmSetMenuCursor(*display, cursorId*)
Display **display*;
Cursor *cursorId*;

void XmSetProtocolHooks(*shell, property, protocol, prehook, pre_closure, posthook, post_closure*)
Widget *shell*;
Atom *property, protocol*;
XtCallbackProc *prehook, posthook*;
XtPointer *pre_closure, post_closure*;

void XmSetWMProtocolHooks(*shell, protocol, prehook, pre_closure, posthook, post_closure*)
Widget *shell*;
Atom *protocol*;
XtCallbackProc *prehook, posthook*;
XtPointer *pre_closure, post_closure*;

Dimension XmStringBaseline(*fontlist, string*)
XmFontList *fontlist*;
XmString *string*;

Boolean XmStringByteCompare(*s1, s2*)
XmString *s1, s2*;

Boolean XmStringCompare(*s1, s2*)
XmString *s1, s2*;

XmString XmStringConcat(*s1, s2*)
XmString *s1, s2*;

XmString XmStringCopy(*s1*)
XmString *s1*;

XmString XmStringCreate(*text, tag*)
char **text, *tag*;

XmString XmStringCreateLocalized (*text*)
char **text*;

XmString XmStringCreateLtoR(*text, tag*)
char **text, *tag*;

XmString XmStringCreateSimple(*text*)
char **text*;

XmString **XmStringDirectionCreate**(*direction*)
XmStringDirection *direction*;

void **XmStringDraw**(*d, w, fontlist, string, gc, x, y, width, alignment, layout_direction, clip*)
Display **d*;
Window *w*;
XmFontList *fontlist*;
XmString *string*;
GC *gc*;
Position *x, y*;
Dimension *width*;
unsigned char *alignment, layout_direction*;
XRectangle **clip*;

void **XmStringDrawImage**(*d, w, fontlist, string, gc, x, y, width, alignment, layout_direction, clip*)
Display **d*;
Window *w*;
XmFontList *fontlist*;
XmString *string*;
GC *gc*;
Position *x, y*;
Dimension *width*;
unsigned char *alignment, layout_direction*;
XRectangle **clip*;

void **XmStringDrawUnderline**(*d, w, fontlist, string, gc, x, y, width, alignment, layout_direction, clip, underline*)
Display **d*;
Window *w*;
XmFontList *fontlist*;
XmString *string*;
GC *gc*;
Position *x, y*;
Dimension *width*;
unsigned char *alignment, layout_direction*;
XRectangle **clip*;
XmString *underline*;

Boolean **XmStringEmpty**(*s1*)
XmString *s1*;

void **XmStringExtent**(*fontlist, string, width, height*)
XmFontList *fontlist*;
XmString *string*;
Dimension **width, *height*;

void **XmStringFree**(*string*)
XmString *string*;

void **XmStringFreeContext**(*context*)
XmStringContext *context*;

Boolean **XmStringGetLtoR**(*string, tag , text*)
XmString *string*;
XmStringCharSet *tag;
char **text*;

XmStringComponentType **XmStringGetNextComponent**(*context, text, tag, direction, unknown_tag, unknown_length, unknown_value*)
XmStringContext *context*;
char **text*;
XmStringCharSet *tag;
XmStringDirection *direction;
XmStringComponentType *unknown_tag;
unsigned short *unknown_length;
unsigned char **unknown_value*;

Boolean **XmStringGetNextSegment**(*context, text, tag, direction, separator*)
XmStringContext *context*;
char **text*;
XmStringCharSet *tag;
XmStringDirection *direction;
Boolean *separator;

Boolean **XmStringHasSubstring**(*string, substring*)
XmString *string*;
XmString *substring*;

Dimension **XmStringHeight**(*fontlist, string*)
XmFontList *fontlist*;
XmString *string*;

Boolean **XmStringInitContext**(*context, string*)
XmStringContext *context*;
XmString *string*;

int **XmStringLength**(*s1*)
XmString *s1*;

int **XmStringLineCount**(*string*)
XmString *string*;

XmString **XmStringNConcat**(*s1, s2, num_bytes*)
XmString *s1, s2*;
int *num_bytes*;

XmString **XmStringNCopy**(*s1, num_bytes*)
    XmString *s1*;
    int *num_bytes*;

XmStringComponentType **XmStringPeekNextComponent**(*context*)
    XmStringContext *context*;

XmString **XmStringSegmentCreate**(*text, tag, direction, separator*)
    char **text*;
    XmStringCharSet *tag*;
    XmStringDirection *direction*;
    Boolean *separator*;

XmString **XmStringSeparatorCreate**()

Dimension **XmStringWidth**(*fontlist, string*)
    XmFontList *fontlist*;
    XmString *string*;

Boolean **XmTargetsAreCompatible** (*display, export_targets, num_export_targets, import_targets, num_import_targets*)
    Display **display*;
    Atom **export_targets, *import_targets*;
    Cardinal *num_export_targets, num_import_targets*;

void **XmTextClearSelection**(*widget, time*)
    Widget *widget*;
    Time *time*;

Boolean **XmTextCopy**(*widget, time*)
    Widget *widget*;
    Time *time*;

Boolean **XmTextCut**(*widget, time*)
    Widget *widget*;
    Time *time*;

void **XmTextDisableRedisplay** (*widget*)
    Widget *widget*;

void **XmTextEnableRedisplay** (*widget*)
    Widget *widget*;

void **XmTextFieldClearSelection**(*widget, time*)
    Widget *widget*;
    Time *time*;

Boolean **XmTextFieldCopy**(*widget, time*)
    Widget *widget*;
    Time *time*;

Boolean XmTextFieldCut(*widget, time*)
Widget *widget*;
Time *time*;

int XmTextFieldGetBaseline(*widget*)
Widget *widget*;

Boolean XmTextFieldGetEditable(*widget*)
Widget *widget*;

XmTextPosition XmTextFieldGetInsertionPosition(*widget*)
Widget *widget*;

XmTextPosition XmTextFieldGetLastPosition(*widget*)
Widget *widget*;

int XmTextFieldGetMaxLength(*widget*)
Widget *widget*;

char *XmTextFieldGetSelection(*widget*)
Widget *widget*;

Boolean XmTextFieldGetSelectionPosition(*widget, left, right*)
Widget *widget*;
XmTextPosition **left, *right*;

wchar_t * XmTextFieldGetSelectionWcs (*widget*)
Widget *widget*;

char *XmTextFieldGetString(*widget*)
Widget *widget*;

wchar_t * XmTextFieldGetStringWcs (*widget*)
Widget *widget*;

int XmTextFieldGetSubstring (*widget, start, num_chars, buffer_size, buffer*)
Widget *widget*;
XmTextPosition *start*;
int *num_chars, buffer_size*;
char **buffer*;

int XmTextFieldGetSubstringWcs (*widget, start, num_chars, buffer_size, buffer*)
Widget *widget*;
XmTextPosition *start*;
int *num_chars, buffer_size*;
wchar_t **buffer*;

void **XmTextFieldInsert**(*widget, position, value*)
    Widget *widget*;
    XmTextPosition *position*;
    char **value*;

void **XmTextFieldInsertWcs** (*widget, position, wcstring*)
    Widget *widget*;
    XmTextPosition *position*;
    wchar_t **wcstring*;

Boolean **XmTextFieldPaste**(*widget*)
    Widget *widget*;

Boolean **XmTextFieldPosToXY**(*widget, position, x, y*)
    Widget *widget*;
    XmTextPosition *position*;
    Position **x, *y*;

Boolean **XmTextFieldRemove**(*widget*)
    Widget *widget*;

void **XmTextFieldReplace**(*widget, from_pos, to_pos, value*)
    Widget *widget*;
    XmTextPosition *from_pos, to_pos*;
    char **value*;

void **XmTextFieldReplaceWcs** (*widget, from_pos, to_pos, wcstring*)
    Widget *widget*;
    XmTextPosition *from_pos*;
    XmTextPosition *to_pos*;
    wchar_t **wcstring*;

void **XmTextFieldSetAddMode**(*widget, state*)
    Widget *widget*;
    Boolean *state*;

void **XmTextFieldSetEditable**(*widget, editable*)
    Widget *widget*;
    Boolean *editable*;

void **XmTextFieldSetHighlight**(*widget, left, right, mode*)
    Widget *widget*;
    XmTextPosition *left, right*;
    XmHighlightMode *mode*;

void **XmTextFieldSetInsertionPosition**(*widget, position*)
    Widget *widget*;
    XmTextPosition *position*;

void **XmTextFieldSetMaxLength**(*widget, max_length*)
    Widget *widget*;
    int *max_length*;

void **XmTextFieldSetSelection**(*widget, first, last, time*)
Widget *widget*;
XmTextPosition *first, last*;
Time *time*;

void **XmTextFieldSetString**(*widget, value*)
Widget *widget*;
char **value*;

void **XmTextFieldSetStringWcs** (*widget, wcstring*)
Widget *widget*;
wchar_t **wcstring*;

void **XmTextFieldShowPosition**(*widget, position*)
Widget *widget*;
XmTextPosition *position*;

XmTextPosition **XmTextFieldXYToPos**(*widget, x, y*)
Widget *widget*;
Position *x, y*;

Boolean **XmTextFindString** (*widget, start, string, direction, position*)
Widget *widget*;
XmTextPosition *start*;
char **string*;
XmTextDirection *direction*;
XmTextPosition **position*;

Boolean **XmTextFindStringWcs** (*widget, start, wcstring, direction, position*)
Widget *widget*;
XmTextPosition *start*;
wchar_t **wcstring*;
XmTextDirection *direction*;
XmTextPosition **position*;

int **XmTextGetBaseline**(*widget*)
Widget *widget*;

Boolean **XmTextGetEditable**(*widget*)
Widget *widget*;

XmTextPosition **XmTextGetInsertionPosition**(*widget*)
Widget *widget*;

XmTextPosition **XmTextGetLastPosition**(*widget*)
Widget *widget*;

int **XmTextGetMaxLength**(*widget*)
Widget *widget*;

char ***XmTextGetSelection**(*widget*)
Widget *widget*;

Boolean **XmTextGetSelectionPosition**(*widget, left, right*)
Widget *widget*;
XmTextPosition **left, **right*;

wchar_t ***XmTextGetSelectionWcs** (*widget*)
Widget *widget*;

XmTextSource **XmTextGetSource**(*widget*)
Widget *widget*;

char ***XmTextGetString**(*widget*)
Widget *widget*;

wchar_t * **XmTextGetStringWcs** (*widget*)
Widget *widget*;

int **XmTextGetSubstring** (*widget, start, num_chars, buffer_size, buffer*)
Widget *widget*;
XmTextPosition *start*;
int *num_chars, buffer_size*;
char **buffer*;

int **XmTextGetSubstringWcs** (*widget, start, num_chars, buffer_size, buffer*)
Widget *widget*;
XmTextPosition *start*;
int *num_chars, buffer_size*;
wchar_t **buffer*;

XmTextPosition **XmTextGetTopCharacter**(*widget*)
Widget *widget*;

void **XmTextInsert**(*widget, position, value*)
Widget *widget*;
XmTextPosition *position*;
char **value*;

void **XmTextInsertWcs** (*widget, position, wcstring*)
Widget *widget*;
XmTextPosition *position*;
wchar_t **wcstring*;

Boolean **XmTextPaste**(*widget*)
Widget *widget*;

Boolean **XmTextPosToXY**(*widget, position, x, y*)
Widget *widget*;
XmTextPosition *position*;
Position **x, *y*;

Boolean **XmTextRemove**(*widget*)
Widget *widget*;

void **XmTextReplace**(*widget, from_pos, to_pos, value*)
    Widget *widget*;
    XmTextPosition *from_pos, to_pos*;
    char **value*;

void **XmTextReplaceWcs** (*widget, from_pos, to_pos, wcstring*)
    Widget *widget*;
    XmTextPosition *from_pos*;
    XmTextPosition *to_pos*;
    wchar_t **wcstring*;

void **XmTextScroll**(*widget, lines*)
    Widget *widget*;
    int *lines*;

void **XmTextSetAddMode**(*widget, state*)
    Widget *widget*;
    Boolean *state*;

void **XmTextSetEditable**(*widget, editable*)
    Widget *widget*;
    Boolean *editable*;

void **XmTextSetHighlight**(*widget, left, right, mode*)
    Widget *widget*;
    XmTextPosition *left, right*;
    XmHighlightMode *mode*;

void **XmTextSetInsertionPosition**(*widget, position*)
    Widget *widget*;
    XmTextPosition *position*;

void **XmTextSetMaxLength**(*widget, max_length*)
    Widget *widget*;
    int *max_length*;

void **XmTextSetSelection**(*widget, first, last, time*)
    Widget *widget*;
    XmTextPosition *first, last*;
    Time *time*;

void **XmTextSetSource**(*widget, source, top_character, cursor_position*)
    Widget *widget*;
    XmTextSource *source*;
    XmTextPosition *top_character, cursor_position*;

void **XmTextSetString**(*widget, value*)
    Widget *widget*;
    char **value*;

void **XmTextSetStringWcs** (*widget, wcstring*)
Widget *widget*;
wchar_t **wcstring*;

void **XmTextSetTopCharacter**(*widget, top_character*)
Widget *widget*;
XmTextPosition *top_character*;

void **XmTextShowPosition**(*widget, position*)
Widget *widget*;
XmTextPosition *position*;

XmTextPosition **XmTextXYToPos**(*widget, x, y*)
Widget *widget*;
Position *x, y*;

Boolean **XmToggleButtonGadgetGetState**(*widget*)
Widget *widget*;

void **XmToggleButtonGadgetSetState**(*widget, state, notify*)
Widget *widget*;
Boolean *state, notify*;

Boolean **XmToggleButtonGetState**(*widget*)
Widget *widget*;

void **XmToggleButtonSetState**(*widget, state, notify*)
Widget *widget*;
Boolean *state, notify*;

Widget **XmTrackingEvent** (*widget, cursor, confine_to, event_return*)
Widget *widget*;
Cursor *cursor*;
Boolean *confine_to*;
XEvent **event_return*;

Widget **XmTrackingLocate**(*widget, cursor, confine_to*)
Widget *widget*;
Cursor *cursor*;
Boolean *confine_to*;

void **XmTranslateKey** (*display, keycode, modifiers, modifiers_return, keysym_return*)
Display **display*;
KeyCode *keycode*;
Modifiers *modifiers*, **modifiers_return*;
KeySym **keysym_return*;

Boolean **XmUninstallImage**(*image*)
XImage **image*;

void **XmUpdateDisplay**(*widget*)
Widget *widget*;

Widget **XmVaCreateSimpleCheckBox**(*parent, name, callback, arg ...*)
Widget *parent*;
String *name*;
XtCallbackProc *callback*;

Widget **XmVaCreateSimpleMenuBar**(*parent, name, arg ...*)
Widget *parent*;
String *name*;

Widget **XmVaCreateSimpleOptionMenu**(*parent, name, option_label, option_mnemonic, button_set, callback, arg ...*)
Widget *parent*;
String *name*;
XmString *option_label*;
KeySym *option_mnemonic*;
int *button_set*;
XtCallbackProc *callback*;

Widget **XmVaCreateSimplePopupMenu**(*parent, name, callback, arg ...*)
Widget *parent*;
String *name*;
XtCallbackProc *callback*;

Widget **XmVaCreateSimplePulldownMenu**(*parent, name, post_from_button, callback, arg ...*)
Widget *parent*;
String *name*;
int *post_from_button*;
XtCallbackProc *callback*;

Widget **XmVaCreateSimpleRadioBox**(*parent, name, button_set, callback, arg ...*)
Widget *parent*;
String *name*;
int *button_set*;
XtCallbackProc *callback*;

Boolean **XmWidgetGetBaselines** (*widget, baselines, line_count*)
Widget *widget*;
Dimension ***baselines*;
int **line_count*;

Boolean **XmWidgetGetDisplayRect** (*widget, displayrect*)
Widget *widget*;
XRectangle **displayrect*;