



Centrum voor Wiskunde en Informatica

REPORTRAPPORT

X python reference manual

K.S. Mullender

Computer Science/Department of Algorithmics and Architecture

CS-R9541 1995

Report CS-R9541
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.

SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

Copyright © Stichting Mathematisch Centrum
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

X Python Reference Manual

Sjoerd Mullender

CWI

P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

sjoard@cwi.nl

Version 1.2 (31 May 1995)

Abstract

This document describes the built-in types, exceptions, and functions of the X windows extension to Python. It assumes basic knowledge about the Python language and access to the X windows documentation. For an informal introduction to the language, see the *Python Tutorial*. The *Python Reference Manual* gives a more formal definition of the language. The *Python Library Reference* describes the built-in and standard modules of Python. This document can be seen as an extension to that document.

CR Subject Classification (1991): D.3.2, D.3.3, D.1.5, D.2.2, I.3.4.

Keywords & Phrases: Object-oriented languages, Python, libraries, modules, Spanish Inquisition, SPAM.

Copyright © 1991-1995 by Stichting Mathematisch Centrum, Amsterdam, The Netherlands.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Contents

1	Introduction	1
1.1	Naming and coding conventions	1
1.2	Drawing	2
1.2.1	Drawing Images	2
1.2.2	Drawing Text	3
1.3	Non-default visuals	3
1.4	Unsupported features	4
2	Built-in module Xlib	5
2.1	Xlib functions and variables	5
2.2	Display object	5
3	Built-in module Xt	7
3.1	Xt functions and variables	7
3.2	Widget Class object	9
3.3	Widget object	9
3.4	Visual Info object	13
3.5	Image object	14
3.6	Graphics Context object	14
3.7	Colormap object	17
3.8	Pixmap object	18
3.9	Font object	19
4	Built-in module Xm	21
4.1	Xm functions and variables	21
4.2	Xm Widget Class object	23
4.3	Xm Widget object	23
4.3.1	Shell widget methods	27
4.3.2	VendorShell widget methods	27
4.3.3	XmCascadeButton widget methods	28
4.3.4	XmCommand widget methods	28
4.3.5	XmFileSelectionBox widget methods	29
4.3.6	XmList widget methods	29
4.3.7	XmMainWindow widget methods	31
4.3.8	XmMessageBox widget methods	31
4.3.9	XmRowColumn widget methods	31

4.3.10	XmScale widget methods	32
4.3.11	XmScrollBar widget methods	32
4.3.12	XmScrolledWindow widget methods	32
4.3.13	XmSelectionBox widget methods	32
4.3.14	XmText widget methods	33
4.3.15	XmTextField widget methods	35
4.3.16	XmToggleButton widget methods	36
4.3.17	XmToggleButtonGadget widget methods	36
5	Built-in module Xaw	38
5.1	Xaw functions and variables	38
6	Built-in module Glx	40
6.1	Glx functions and variables	40
7	Built-in module HTML	41
7.1	HTML functions and variables	41
7.2	HTML Widget object	41

Chapter 1

Introduction

This document describes the calling sequences and return values of all functions that have been implemented in the X extension of Python. If a function is not mentioned in this document, it was not implemented. In addition, the actual X function that is called internally is mentioned. For the precise semantics of the function, please refer to the X documentation.

In many cases, functions in the X libraries have been turned into methods of various objects. See the sections on the various objects for lists of defined methods. See the index to find of which object a function is a method.

As an appetizer, here is the Hello World program using Motif.

```
import sys, Xt, Xm

def main():
    toplevel = Xt.Initialize('XHello', [], sys.argv)
    hello = toplevel.CreateManagedWidget('hello', Xm.Label,
                                         {'labelString': 'Hello World'})
    toplevel.RealizeWidget()
    Xt.MainLoop()

main()
```

1.1 Naming and coding conventions

Prefixes, such as `Xt`, `Xm`, and `Glx` are dropped from a function name.

Only `Xt` functions that have no widget or widget class argument are defined directly in the `Xt` module.

`XtInitialize` must be called as `Xt.Initialize('classname', [], sys.argv)` or as `Xt.Initialize()`.

`Xt` and `Xm` functions with a widget parameter are changed into widget methods, with the widget parameter removed, even if it is not the first parameter.

Xt functions with a widget class parameter are changed into widget class methods.

Parameters, return values, and resources of type `XmString` are replaced by ordinary Python strings.

Parameters representing structures are replaced by Python tuples.

Some functions accept a partially filled struct with a flag describing which members are filled in. This is represented by a dictionary where the key is the name of the struct members to be filled in, and the value the value of these members. No flag is used.

Pairs of parameters describing arrays of strings or structures are replaced by a Python list of strings or tuples.

Output parameters are always replaced by Python function return values. If there is more than one output parameter, a tuple is constructed. If the functions also returns a value, it is the first element of the tuple.

Input parameters of type `ArgList`, such as those given to `XtCreateWidget` and `XtSetValues`, are replaced by a dictionary with the resource name as key and the resource value as value. Also, the return value of `XtGetValues` is such a dictionary.

Functions in Xlib that take both a `Display` and a `Window` argument are implemented as widget methods (if they are implemented at all). The `Display` and `Window` arguments are then removed from the argument list.

1.2 Drawing

Python Xt supports drawing operations, usable e.g. in Motif `DrawingArea` widgets and on off-screen `Pixmap`s. In general the X man pages should be consulted for the precise semantics.

An off-screen `Pixmap` can be created using the widget method `CreatePixmap`.

In order to draw in a widget or on a `Pixmap`, a `Graphics Context` object instance must be created. This can be done using the widget methods `CreateGC` and `GetGC`, or the `Pixmap` method `CreateGC`.

Once a `Graphics Context` object is instantiated, the Xlib drawing functions can be used. These drawing functions are methods of the `Graphics Context` object.

For example, in order to draw a rectangle in a window, the following code could be used.

```
values = {'line_style': X.LineOnOffDash}
gc = w.GetGC(values)
gc.DrawRectangle(x, y, width, height)
```

1.2.1 Drawing Images

An image (`pixmap`) can be drawn by using the `GC` method `PutImage`. This method requires an `Image` object which can be created using the `Visual Info` method `CreateImage`. The only way to get a `Visual Info` object is by using the widget method `GetVisualInfo` which returns a list of `Visual Info` objects.

For example (`picture` is a string object that contains an image in the correct format for the call to `CreateImage`, see the manual for more information):

```
visual = w.DefaultVisualOfScreen()
image = visual.CreateImage(depth, X.ZPixmap, 0, picture, im_w, im_h, pad, 0)
gc.PutImage(image, im_x, im_y, win_x, win_y, win_w, win_h)
```

1.2.2 Drawing Text

To draw text, a font must first be selected. A font object is acquired by using the widget method `LoadQueryFont`. This object can then be passed to the GC method `SetFont` or assigned to the GC instance variable `font`.

For example:

```
font = w.LoadQueryFont('-*-times-medium-r-*-*-100-*-*-*-iso8859-1')
gc.SetFont(font)
gc.DrawString(x, y, 'Hello World!')
```

1.3 Non-default visuals

First you must find the visual you want to use, then you must create a colormap using that visual. When creating a top-level shell widget, you must specify the visual, the colormap, and the depth of the visual.

Motif only allows you to change the visual of a shell widget, so the visual and colormap must be created before the shell widget is created.

When creating another shell widget, the new widget inherits the colormap and depth from the parent widget, but it inherits the visual from the parent window (the root window). Thus on every shell you create you must specify visual, colormap, and depth. Note that popup dialogs and menus are also shells.

Sample code that searches for a visual and creates a colormap which is then used by a shell widget is given here. (Note that `Xm` must be imported for the `CreateForm` method to be available.)

```
import Xt, Xm, X, sys
Xt.ToolkitInitialize()
dpy = Xt.OpenDisplay(None, None, None, [], sys.argv)
visuals = dpy.GetVisualInfo({'class': X.TrueColor})
if not visuals:
    print 'No TrueColor visuals available'
    sys.exit(1)
v_best = visuals[0]
for v in visuals:
    if v.depth > v_best.depth:
```



```

        v_best = v
    colormap = v_best.CreateColormap(X.AllocNone)
    toplevel = Xt.CreateApplicationShell('shell', Xt.ApplicationShell,
        {'visual': v_best, 'depth': v_best.depth, 'colormap': colormap})
    form = toplevel.CreateForm('form', {'width': 100, 'height': 100})
    form.ManageChild()
    toplevel.RealizeWidget()
    Xt.MainLoop()

```

1.4 Unsupported features

None of the `XtApp*` functions have been implemented, but most of the corresponding functions without `App` have been implemented, except where noted below. Thus you cannot use an other than the default application context. Other unimplemented functions are `XtCreateApplicationContext` and `XtDestroyApplicationContext`.

The functions that deal with application resources have not been implemented. The affected functions are: `XtGetApplicationResources`, `XtAddConverter`, and all `Xrm*` functions. A workaround is to use `void = os.system('xrdp -nocpp -merge <name>.resources')`.

The translation manager functions have not been implemented. The affected functions are `XtAddActions`, `XtParseTranslationsTable`, `XtAugmentTranslations`, and `XtOverrideTranslations`.

The geometry management functions `XtQueryGeometry` and `XtMakeResizeRequest` are not implemented.

Only a few functions with `Display`, `Screen`, or `Event` arguments or return types have been implemented.

The `XmClipboard*` functions have not been implemented.

The `XmCvt*` functions have not been implemented.

The Motif `Pixmap` and `Image` functions (`XmDestroyPixmap`, `XmInstallImage`, `XmUninstallImage`, and `XmGetPixmap`) have not been implemented.

The Motif drag and drop functions have not been implemented.

The `XmFontList*` functions have not been implemented.

Chapter 2

Built-in module Xlib

2.1 Xlib functions and variables

The function `getevent` is a hack. Some functions return a value that contains a pointer to an `XEvent` structure. The function `getevent` can be used to obtain the structure.

The module defines the following functions and variables:

`Error`

Exception generated by functions in this module.

`LookupString(event)`

Returns (string, keysym). This function corresponds with `XLookupString`.

`getevent(ptr)`

Returns event.

2.2 Display object

A display object can only be acquired by using the widget method `Display` or by the Xt function `OpenDisplay`.

Display objects (`display_objects`) have the following methods:

`CheckMaskEvent(event_mask)`

Returns None. This function corresponds with `XCheckMaskEvent`.

`ConnectionNumber()`

Returns `connection_number`. This function corresponds with `ConnectionNumber`.

`CreateFontCursor(shape)`

Returns cursor. This function corresponds with `XCreateFontCursor`.

`Flush()`

Returns None. This function corresponds with `XFlush`.

`GetVisualInfo(template_dict)`

Returns `vi_object_list`. This function corresponds with `XGetVisualInfo`.

`NextEvent()`

Returns `event`. This function corresponds with `XNextEvent`.

`PeekEvent()`

Returns `event`. This function corresponds with `XPeekEvent`.

`PutBackEvent(event)`

Returns `None`. This function corresponds with `XPutBackEvent`.

Chapter 3

Built-in module Xt

3.1 Xt functions and variables

The functions in the Xt library that have a widget argument are all turned into widget methods. The functions in the Xt library that have a widget class argument have been turned into widget_class methods. The remaining functions are given here.

The module defines the following functions and variables:

`AddInput(source, condition, proc, client_data)`

Returns `input_id`. This function corresponds with `XtAddInput`.

`AddTimeOut(interval, proc, client_data)`

Returns `interval_id`. This function corresponds with `XtAddTimeOut`.

`AddWorkProc(proc, client_data)`

Returns `workproc_id`. This function corresponds with `XtAddWorkProc`.

`ApplicationShell`

Widget class object.

`Composite`

Widget class object.

`Constraint`

Widget class object.

`CreateApplicationShell(application_name, widget_class, argdict)`

Returns `widget`. This function corresponds with `XtCreateApplicationShell`.

`DispatchEvent(event)`

Returns `None`. This function corresponds with `XtDispatchEvent`.

`Error`

Exception generated by functions in this module.

`Initialize([classname, option_list, argv])`

Returns widget. This function corresponds with XtInitialize.

MainLoop()
Returns None. This function corresponds with XtMainLoop.

ManageChildren(*widget_list*)
Returns None. This function corresponds with XtManageChildren.

NextEvent()
Returns event. This function corresponds with XtNextEvent.

OpenDisplay([*display*, *appname*, *classname*, *option_list*, *argv*])
Returns display_object. This function corresponds with XtOpenDisplay.

OverrideShell
Widget class object.

PeekEvent()
Returns event. This function corresponds with XtPeekEvent.

Pending()
Returns mask. This function corresponds with XtPending.

ProcessEvent(*mask*)
Returns None. This function corresponds with XtProcessEvent.

RemoveInput(*input_id*)
Returns None. This function corresponds with XtRemoveInput.

RemoveTimeOut(*interval_id*)
Returns None. This function corresponds with XtRemoveTimeOut.

RemoveWorkProc(*workproc_id*)
Returns None. This function corresponds with XtRemoveWorkProc.

Shell
Widget class object.

ToolkitInitialize()
Returns None. This function corresponds with XtToolkitInitialize.

TopLevelShell
Widget class object.

TransientShell
Widget class object.

VendorShell
Widget class object.

WMShell
Widget class object.

3.2 Widget Class object

Widget Class objects have the following method:

`GetConstraintResourceList()`

Returns `resource_list`. This function corresponds with `XtGetConstraintResourceList`.

`GetResourceList()`

Returns `resource_list`. This function corresponds with `XtGetResourceList`.

3.3 Widget object

Widget objects have the following methods:

`AddCallback(callback_name, callback, client_data)`

Returns `None`. This function corresponds with `XtAddCallback`.

`AddEventHandler(event_mask, nonmaskable, proc, client_data)`

Returns `None`. This function corresponds with `XtAddEventHandler`.

`AddRawEventHandler(event_mask, nonmaskable, proc, client_data)`

Returns `None`. This function corresponds with `XtAddRawEventHandler`.

`BuildEventMask()`

Returns `event_mask`. This function corresponds with `XtBuildEventMask`.

`CallCallbacks(callback_name, call_data)`

Returns `None`. This function corresponds with `XtCallCallbacks`.

`ChangeWindowAttributes(window_attribute_dict)`

Returns `None`. This function corresponds with `XChangeWindowAttributes`.

`Class()`

Returns `widget_class`. This function corresponds with `XtClass`.

`Convert(from_value, to_type)`

Returns `to_value`. This function corresponds with `XtConvertAndStore` or `XtConvert`.

`CreateColormap(vi_object, alloc)`

Returns `cm_object`. This function corresponds with `XCreateColormap`.

`CreateGC(gc_dict)`

Returns `gc_object`. This function corresponds with `XCreateGC`.

`CreateManagedWidget(name, widget_class, argdict)`

Returns `widget`. This function corresponds with `XtCreateManagedWidget`.

`CreatePixmap([width, height [, depth]])`

Returns `pixmap_object`. This function corresponds with `XCreatePixmap`.

`CreatePopupShell(name, widget_class, argdict)`
 Returns `widget`. This function corresponds with `XtCreatePopupShell`.

`CreateWidget(name, widget_class, argdict)`
 Returns `widget`. This function corresponds with `XtCreateWidget`.

`DefaultColormap()`
 Returns `cm_object`. This function corresponds with `DefaultColormapOfScreen`.

`DefaultColormapOfScreen()`
 Returns `cm_object`. This function corresponds with `DefaultColormapOfScreen`.

`DefaultVisual()`
 Returns `vi_object`. This function corresponds with `DefaultVisualOfScreen`.

`DefaultVisualOfScreen()`
 Returns `vi_object`. This function corresponds with `DefaultVisualOfScreen`.

`DefineCursor(cursor)`
 Returns `None`. This function corresponds with `XDefineCursor`.

`DestroyWidget()`
 Returns `None`. This function corresponds with `XtDestroyWidget`.

`Display()`
 Returns `display_object`. This function corresponds with `XtDisplay`.

`GetGC(gc_dict)`
 Returns `gc_object`. This function corresponds with `XtGetGC`.

`GetGeometry()`
 Returns (`root`, `x`, `y`, `width`, `height`, `border_width`, `depth`). This function corresponds with `XGetGeometry`.

`GetValues(name_list)`
 Returns `argdict`. This function corresponds with `XtGetValues`.

`GetVisualInfo(template_dict)`
 Returns `vi_object_list`. This function corresponds with `XGetVisualInfo`.

`GetWMNormalHints()`
 Returns `hintdict`. This function corresponds with `XGetWMNormalHints`.

`GrabButton(button, modifiers, owner_events, event_mask, pointer_mode, keyboard_mode, confine_to_widget)`
 Returns `None`. This function corresponds with `XGrabButton`.

`HasCallbacks(callback_name)`
 Returns `callback_status`. This function corresponds with `XtHasCallbacks`.

`HeightMMOfScreen()`
 Returns `height_mm`. This function corresponds with `HeightMMOfScreen`.

HeightOfScreen()
Returns height. This function corresponds with HeightOfScreen.

InstallAccelerators(*source*)
Returns None. This function corresponds with XtInstallAccelerators.

InstallAllAccelerators(*source*)
Returns None. This function corresponds with XtInstallAllAccelerators.

IsComposite()
Returns success. This function corresponds with XtIsComposite.

IsManaged()
Returns success. This function corresponds with XtIsManaged.

IsRealized()
Returns success. This function corresponds with XtIsRealized.

IsSensitive()
Returns success. This function corresponds with XtIsSensitive.

IsSubclass(*widget_class*)
Returns success. This function corresponds with XtIsSubclass.

ListFonts(*pattern*)
Returns font_list. This function corresponds with XListFonts.

LoadQueryFont(*name*)
Returns font_object. This function corresponds with XLoadQueryFont.

LowerWindow()
Returns None. This function corresponds with XLowerWindow.

ManageChild()
Returns None. This function corresponds with XtManageChild.

MapWidget()
Returns None. This function corresponds with XtMapWidget.

NameToWidget(*name*)
Returns widget. This function corresponds with XtNameToWidget.

Parent()
Returns parent_widget. This function corresponds with XtParent.

Popdown()
Returns None. This function corresponds with XtPopdown.

Popup(*grab_kind*)
Returns None. This function corresponds with XtPopup.

QueryPointer()
Returns (root, child, root_x, root_y, win_x, win_y, mask). This function corresponds with XQueryPointer.

RaiseWindow()
Returns None. This function corresponds with XRaiseWindow.

`ReadBitmapFile(file)`
 Returns (width, height, pixmap_object, hotspot_x, hotspot_y). This function corresponds with `XReadBitmapFile`.

`RealizeWidget()`
 Returns None. This function corresponds with `XtRealizeWidget`.

`RemoveAllCallbacks(callback_name)`
 Returns None. This function corresponds with `XtRemoveAllCallbacks`.

`RemoveCallback(callback_name, callback, client_data)`
 Returns None. This function corresponds with `XtRemoveAllCallbacks`.

`RemoveEventHandler(event_mask, nonmaskable, proc, client_data)`
 Returns None. This function corresponds with `XtRemoveEventHandler`.

`RemoveRawEventHandler(event_mask, nonmaskable, proc, client_data)`
 Returns None. This function corresponds with `XtRemoveRawEventHandler`.

`SetMappedWhenManaged(map_when_managed)`
 Returns None. This function corresponds with `XtSetMappedWhenManaged`.

`SetSensitive(sensitive)`
 Returns None. This function corresponds with `XtSetSensitive`.

`SetValues(argdict)`
 Returns None. This function corresponds with `XtSetValues`.

`SetWMNormalHints(hintdict)`
 Returns None. This function corresponds with `XSetWMNormalHints`.

`SetWindowColormap(cm_object)`
 Returns None. This function corresponds with `XSetWindowColormap`.

`Superclass()`
 Returns `widget_class`. This function corresponds with `XtSuperclass`.

`TranslateCoords(x, y)`
 Returns (`rootx`, `rooty`). This function corresponds with `XtTranslateCoords`.

`UndefineCursor()`
 Returns None. This function corresponds with `XUndefineCursor`.

`UngrabButton(button, modifiers)`
 Returns None. This function corresponds with `XUngrabButton`.

`UnmanageChild()`
 Returns None. This function corresponds with `XtUnmanageChild`.

`UnmapWidget()`
 Returns None. This function corresponds with `XtUnmapWidget`.

`UnrealizeWidget()`
 Returns None. This function corresponds with `XtUnrealizeWidget`.

WidthMMOfScreen()

Returns `width_mm`. This function corresponds with `WidthMMOfScreen`.

WidthOfScreen()

Returns `width`. This function corresponds with `WidthOfScreen`.

Window()

Returns `window`. This function corresponds with `XtWindow`.

3.4 Visual Info object

Visual Info objects are used to represent a `XVisualInfo` structure. The widget method `GetVisualInfo` returns a list of Visual Info objects; the widget method `DefaultVisualOfScreen` returns a single Visual Info object. Visual Info objects can be used to create a colormap using the widget method `CreateColormap` and to create an Image object using the Visual Info method `CreateImage`.

Visual Info objects (`vi_objects`) have the following method and variables:

`CreateColormap(alloc)`

Returns `cm_object`. This function corresponds with `XCreateColormap`.

`CreateImage(depth, format, offset, data, width, height, bitmap_pad, bytes_per_line)`

Returns `im_object`. This function corresponds with `XCreateImage`.

`bits_per_rgb`

Field in `XVisualInfo` structure.

`blue_mask`

Field in `XVisualInfo` structure.

`c_class`

Field in `XVisualInfo` structure. Alternative name for the `class` field.

`class`

Field in `XVisualInfo` structure.

`colormap_size`

Field in `XVisualInfo` structure.

`depth`

Field in `XVisualInfo` structure.

`green_mask`

Field in `XVisualInfo` structure.

`red_mask`

Field in `XVisualInfo` structure.

`visual`

Field in `XVisualInfo` structure.

`visualid`

Field in XVisualInfo structure.

3.5 Image object

An Image object is used to represent an XImage structure. An Image object is created using the CreateImage method of a Visual Info object. The image can be copied to a window or a pixmap using the Graphics Context method PutImage. Image objects (im_objects) have no methods or variables.

3.6 Graphics Context object

All Graphics Context objects have methods that perform drawing operations. They are listed here, with the argument types they expect. XPoint is a pair of integers; XRectangle is a tuple of four integers; XSegment is a tuple of four integers; XArc is a tuple of six integers; <type>[] stands for a list of <type> objects.

All GC objects have data attributes representing the individual values. These are gc.line_width, gc.line_style, etc. (The dash attribute is not supported.) These attributes can be assigned to if the GC object is writable.

The Pixmap arguments are Pixmap objects; the Font arguments are Font objects; the Image arguments are Image objects.

Graphics Context objects (gc_objects) have the following methods and variables:

ChangeGC(*gc_dict*)

Returns None. This function corresponds with XChangeGC.

DrawArc(*x, y, width, height, angle1, angle2*)

Returns None. This function corresponds with XDrawArc.

DrawArcs(*XArc[]*)

Returns None. This function corresponds with XDrawArcs.

DrawImageString(*x, y, string*)

Returns None. This function corresponds with XDrawImageString.

DrawLine(*x1, y1, x2, y2*)

Returns None. This function corresponds with XDrawLine.

DrawLines(*XPoint[], mode*)

Returns None. This function corresponds with XDrawLines.

DrawPoint(*x, y*)

Returns None. This function corresponds with XDrawPoint.

DrawPoints(*XPoint[], mode*)

Returns None. This function corresponds with XDrawPoints.

`DrawRectangle(x, y, width, height)`
Returns None. This function corresponds with XDrawRectangle.

`DrawRectangles(XRectangle[])`
Returns None. This function corresponds with XDrawRectangles.

`DrawSegments(XSegment[])`
Returns None. This function corresponds with XDrawSegments.

`DrawString(x, y, string)`
Returns None. This function corresponds with XDrawString.

`FillArc(x, y, width, height, angle1, angle2)`
Returns None. This function corresponds with XFillArc.

`FillArcs(XArc[])`
Returns None. This function corresponds with XFillArcs.

`FillPolygon(XPoint[], shape, mode)`
Returns None. This function corresponds with XFillPolygon.

`FillRectangle(x, y, width, height)`
Returns None. This function corresponds with XFillRectangle.

`FillRectangles(XRectangle[])`
Returns None. This function corresponds with XFillRectangles.

`PutImage(Image, src_x, src_y, dest_x, dest_y, width, height)`
Returns None. This function corresponds with XPutImage.

`SetArcMode(arc_mode)`
Returns None. This function corresponds with XSetArcMode.

`SetBackground(background)`
Returns None. This function corresponds with XSetBackground.

`SetClipMask(Pixmap)`
Returns None. This function corresponds with XSetClipMask.

`SetClipOrigin(clip_x_origin, clip_y_origin)`
Returns None. This function corresponds with XSetClipOrigin.

`SetClipRectangles(clip_x_origin, clip_y_origin, rectangle_list, ordering)`
Returns None. This function corresponds with XSetClipRectangles.

`SetDashes(dash_offset, dash_list)`
Returns None. This function corresponds with XSetDashes.

`SetFillRule(fill_rule)`
Returns None. This function corresponds with XSetFillRule.

`SetFillStyle(fill_style)`
Returns None. This function corresponds with XSetFillStyle.

`SetFont(font_object)`
Returns None. This function corresponds with XSetFont.

`SetForeground(foreground)`
 Returns None. This function corresponds with `XSetForeground`.

`SetFunction(function)`
 Returns None. This function corresponds with `XSetFunction`.

`SetGraphicsExposures(graphics_exposure)`
 Returns None. This function corresponds with `XSetGraphicsExposures`.

`SetLineAttributes(line_width, line_style, cap_style, join_style)`
 Returns None. This function corresponds with `XSetLineAttributes`.

`SetPlaneMask(plane_mask)`
 Returns None. This function corresponds with `XSetPlaneMask`.

`SetState((foreground, background, function, plane_mask))`
 Returns None. This function corresponds with `XSetState`.

`SetStipple(stipple)`
 Returns None. This function corresponds with `XSetStipple`.

`SetSubwindowMode(subwindow_mode)`
 Returns None. This function corresponds with `XSetSubwindowMode`.

`SetTSOrigin(ts_x_origin, ts_y_origin)`
 Returns None. This function corresponds with `XSetTSOrigin`.

`SetTitle(tile)`
 Returns None. This function corresponds with `XSetTitle`.

`arc_mode`
 Field in `XGCValues` structure.

`background`
 Field in `XGCValues` structure.

`cap_style`
 Field in `XGCValues` structure.

`clip_mask`
 Field in `XGCValues` structure.

`clip_x_origin`
 Field in `XGCValues` structure.

`clip_y_origin`
 Field in `XGCValues` structure.

`dash_offset`
 Field in `XGCValues` structure.

`file_style`
 Field in `XGCValues` structure.

`fill_rule`
 Field in `XGCValues` structure.

`font`

Field in XGCValues structure.

foreground
Field in XGCValues structure.

function
Field in XGCValues structure.

graphics_exposures
Field in XGCValues structure.

join_style
Field in XGCValues structure.

line_style
Field in XGCValues structure.

line_width
Field in XGCValues structure.

plane_mask
Field in XGCValues structure.

stipple
Field in XGCValues structure.

subwindow_mode
Field in XGCValues structure.

tile
Field in XGCValues structure.

ts_x_origin
Field in XGCValues structure.

ts_y_origin
Field in XGCValues structure.

3.7 Colormap object

Colormap objects are used to represent the Colormap type. A Colormap object can be obtained using the widget method `CreateColormap` (which also needs a Visual Info object) or the widget method `DefaultColormapOfScreen`. A Colormap object can be used in the widget method `SetWindowColormap` and as the colormap resource of widgets.

Colormap objects (`cm_objects`) have the following methods:

`AllocColor(red, green, blue)`

Returns (`pixel, red, green, blue, flags`). This function corresponds with `XAllocColor`.

`AllocColorCells(contig, nplanes, npixels)`

Returns (`plane_masks_list, pixels_list`). This function corresponds with

XAllocColorCells.

AllocNamedColor(*color_name*)
 Returns
 ((screen_def.pixel, screen_def.red, screen_def.green, screen_def.blue, screen_def.flags),
 This function corresponds with XAllocNamedColor.

FreeColors(*pixel_list*, *planes*)
 Returns None. This function corresponds with XFreeColors.

Install()
 Returns None. This function corresponds with XInstallColormap.

InstallColormap()
 Returns None. This function corresponds with XInstallColormap.

LookupColor(*color_name*)
 Returns
 ((exact_def.pixel, exact_def.red, exact_def.green, exact_def.blue, exact_def.flags), (scr
 This function corresponds with XLookupColor.

ParseColor(*color_name*)
 Returns (pixel, red, green, blue, flags). This function corresponds
 with XParseColor.

QueryColor(*pixel*)
 Returns (pixel, red, green, blue, flags). This function corresponds
 with XQueryColor.

QueryColors(*pixel_list*)
 Returns XColor_list. This function corresponds with XQueryColors.

StoreColors(*color_list*)
 Returns None. This function corresponds with XStoreColors.

3.8 Pixmap object

Pixmap objects are used to represent the Pixmap type. A Pixmap object can be obtained using the widget method CreatePixmap.

A Pixmap is an offscreen piece of memory in which the application can draw. Drawing in the pixmap can be done using a Graphics Context object which can be created using the CreateGC method. The pixmap can be copied to a window or another pixmap using the CopyArea and CopyPlane methods.

Pixmap objects (pixmap_objects) have the following methods:

CopyArea(*destobject*, *gc_object*, *src_x*, *src_y*, *width*, *height*, *dest_x*, *dest_y*)
 Returns None. This function corresponds with XCopyArea.

CopyPlane(*destobject*, *gc_object*, *src_x*, *src_y*, *width*, *height*, *dest_x*, *dest_y*, *plane*)
 Returns None. This function corresponds with XCopyPlane.

CreateGC(*gc_dict*)

Returns `gc_object`. This function corresponds with `XCreateGC`.

`ReadBitmapFile(file)`

Returns (`width`, `height`, `pixmap_object`, `hotspot_x`, `hotspot_y`). This function corresponds with `XReadBitmapFile`.

3.9 Font object

The Font object attributes are read-only. The Font ID (`fid`) is fairly useless. Everywhere a Font ID is required in the library, a Font object must be given.

Font objects (`font_objects`) have the following methods and variables:

`TextExtents(string)`

Returns

(`direction`, `font_ascent`, `font_descent`, (`lbearing`, `rbearing`, `width`, `ascent`, `descent`, `attri`

This function corresponds with `XTextExtents`.

`TextWidth(string)`

Returns `width`. This function corresponds with `XTextWidth`.

`all_chars_exist`

Field in `XFontStruct` structure.

`ascent`

Field in `XFontStruct` structure.

`default_char`

Field in `XFontStruct` structure.

`descent`

Field in `XFontStruct` structure.

`direction`

Field in `XFontStruct` structure.

`fid`

Field in `XFontStruct` structure.

`max_bounds`

Field in `XFontStruct` structure.

`max_byte1`

Field in `XFontStruct` structure.

`max_char_or_byte2`

Field in `XFontStruct` structure.

`min_bounds`

Field in `XFontStruct` structure.

`min_byte1`

Field in `XFontStruct` structure.

`min_char_or_byte2`
Field in `XFontStruct` structure.

`n_properties`
Field in `XFontStruct` structure.

`per_char`
Field in `XFontStruct` structure.

`properties`
Field in `XFontStruct` structure.

Chapter 4

Built-in module Xm

Which of the variables and methods are available depends on the version of the Motif libraries. The variable `TextField` is only available in Motif versions 1.1 and higher, the variable `Display` is only available in Motif version 1.2 and higher. To check dynamically which Motif version is being used, one can check whether these variables exist in the `Xm` module.

4.1 Xm functions and variables

`ArrowButtonGadget`

Widget class object.

`BulletinBoard`

Widget class object.

`CascadeButton`

Widget class object.

`CascadeButtonGadget`

Widget class object.

`Command`

Widget class object.

`DialogShell`

Widget class object.

`Display`

Widget class object.

`DrawingArea`

Widget class object.

`DrawnButton`

Widget class object.

FileSelectionBox
Widget class object.

Form
Widget class object.

Frame
Widget class object.

Gadget
Widget class object.

Label
Widget class object.

LabelGadget
Widget class object.

List
Widget class object.

MainWindow
Widget class object.

MenuShell
Widget class object.

MessageBox
Widget class object.

PanedWindow
Widget class object.

PushButton
Widget class object.

PushButtonGadget
Widget class object.

RowColumn
Widget class object.

Scale
Widget class object.

ScrollBar
Widget class object.

ScrolledWindow
Widget class object.

SelectionBox
Widget class object.

Separator
Widget class object.

SeparatorGadget
Widget class object.

Text
Widget class object.

TextField
Widget class object.

ToggleButton
Widget class object.

ToggleButtonGadget
Widget class object.

4.2 Xm Widget Class object

Motif Widget Class objects have the following method in addition to the standard ones:

GetSecondaryResourceData()
Returns `resource_list_list`. This function corresponds with `XmGetSecondaryResourceData`.

4.3 Xm Widget object

If Motif is included all widget objects have the following methods in addition to the normal widget methods. These methods are only available after Xm was imported at least once somewhere in the program.

AddTabGroup()
Returns `None`. This function corresponds with `XmAddTabGroup`.

ChangeColor(*background*)
Returns `None`. This function corresponds with `XmChangeColor`.

ConvertUnits(*orientation, from_unit_type, from_value, to_unit_type*)
Returns `None`. This function corresponds with `XmConvertUnits`.

CreateArrowButton(*name, argdict*)
Returns `widget`. This function corresponds with `XmCreateArrowButton`.

CreateArrowButtonGadget(*name, argdict*)
Returns `widget`. This function corresponds with `XmCreateArrowButtonGadget`.

CreateBulletinBoard(*name, argdict*)
Returns `widget`. This function corresponds with `XmCreateBulletinBoard`.

CreateBulletinBoardDialog(*name, argdict*)
Returns `widget`. This function corresponds with

XmCreateBulletinBoardDialog.

CreateCascadeButton(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateCascadeButton.

CreateCascadeButtonGadget(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateCascadeButtonGadget.

CreateCommand(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateCommand.

CreateDialogShell(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateDialogShell.

CreateDrawingArea(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateDrawingArea.

CreateDrawnButton(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateDrawnButton.

CreateErrorDialog(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateErrorDialog.

CreateFileSelectionBox(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateFileSelectionBox.

CreateFileSelectionDialog(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateFileSelectionDialog.

CreateForm(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateForm.

CreateFormDialog(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateFormDialog.

CreateFrame(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateFrame.

CreateInformationDialog(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateInformationDialog.

CreateLabel(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateLabel.

CreateLabelGadget(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateLabelGadget.

CreateList(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateList.

CreateMainWindow(*name*, *argdict*)
Returns widget. This function corresponds with XmCreateMainWindow.

`CreateMenuBar(name, argdict)`
Returns widget. This function corresponds with `XmCreateMenuBar`.

`CreateMenuShell(name, argdict)`
Returns widget. This function corresponds with `XmCreateMenuShell`.

`CreateMessageBox(name, argdict)`
Returns widget. This function corresponds with `XmCreateMessageBox`.

`CreateMessageDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreateMessageDialog`.

`CreateOptionMenu(name, argdict)`
Returns widget. This function corresponds with `XmCreateOptionMenu`.

`CreatePanedWindow(name, argdict)`
Returns widget. This function corresponds with `XmCreatePanedWindow`.

`CreatePopupMenu(name, argdict)`
Returns widget. This function corresponds with `XmCreatePopupMenu`.

`CreatePromptDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreatePromptDialog`.

`CreatePulldownMenu(name, argdict)`
Returns widget. This function corresponds with `XmCreatePulldownMenu`.

`CreatePushButton(name, argdict)`
Returns widget. This function corresponds with `XmCreatePushButton`.

`CreatePushButtonGadget(name, argdict)`
Returns widget. This function corresponds with `XmCreatePushButtonGadget`.

`CreateQuestionDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreateQuestionDialog`.

`CreateRadioBox(name, argdict)`
Returns widget. This function corresponds with `XmCreateRadioBox`.

`CreateRowColumn(name, argdict)`
Returns widget. This function corresponds with `XmCreateRowColumn`.

`CreateScale(name, argdict)`
Returns widget. This function corresponds with `XmCreateScale`.

`CreateScrollBar(name, argdict)`
Returns widget. This function corresponds with `XmCreateScrollBar`.

`CreateScrolledList(name, argdict)`
Returns widget. This function corresponds with `XmCreateScrolledList`.

`CreateScrolledText(name, argdict)`
Returns widget. This function corresponds with `XmCreateScrolledText`.

`CreateScrolledWindow(name, argdict)`
Returns widget. This function corresponds with `XmCreateScrolledWindow`.

`CreateSelectionBox(name, argdict)`
Returns widget. This function corresponds with `XmCreateSelectionBox`.

`CreateSelectionDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreateSelectionDialog`.

`CreateSeparator(name, argdict)`
Returns widget. This function corresponds with `XmCreateSeparator`.

`CreateSeparatorGadget(name, argdict)`
Returns widget. This function corresponds with `XmCreateSeparatorGadget`.

`CreateSimpleCheckBox(name, argdict)`
Returns widget. This function corresponds with `XmCreateSimpleCheckBox`.

`CreateSimpleMenuBar(name, argdict)`
Returns widget. This function corresponds with `XmCreateSimpleMenuBar`.

`CreateSimpleOptionMenu(name, argdict)`
Returns widget. This function corresponds with `XmCreateSimpleOptionMenu`.

`CreateSimplePopupMenu(name, argdict)`
Returns widget. This function corresponds with `XmCreateSimplePopupMenu`.

`CreateSimplePulldownMenu(name, argdict)`
Returns widget. This function corresponds with `XmCreateSimplePulldownMenu`.

`CreateSimpleRadioBox(name, argdict)`
Returns widget. This function corresponds with `XmCreateSimpleRadioBox`.

`CreateTemplateDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreateTemplateDialog`.

`CreateText(name, argdict)`
Returns widget. This function corresponds with `XmCreateText`.

`CreateTextField(name, argdict)`
Returns widget. This function corresponds with `XmCreateTextField`.

`CreateToggleButton(name, argdict)`
Returns widget. This function corresponds with `XmCreateToggleButton`.

`CreateToggleButtonGadget(name, argdict)`
Returns widget. This function corresponds with `XmCreateToggleButtonGadget`.

`CreateWarningDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreateWarningDialog`.

`CreateWorkArea(name, argdict)`
Returns widget. This function corresponds with `XmCreateWorkArea`.

`CreateWorkingDialog(name, argdict)`
Returns widget. This function corresponds with `XmCreateWorkingDialog`.

GetDisplay()
 Returns `display_widget`. This function corresponds with `XmGetDisplay`.

GetFocusWidget()
 Returns `widget`. This function corresponds with `XmGetFocusWidget`.

GetTabGroup()
 Returns `widget`. This function corresponds with `XmGetTabGroup`.

GetVisibility()
 Returns `visibility`. This function corresponds with `XmGetVisibility`.

IsTraversable()
 Returns `success`. This function corresponds with `XmIsTraversable`.

ProcessTraversal(*direction*)
 Returns `success`. This function corresponds with `XmProcessTraversal`.

RemoveTabGroup()
 Returns `None`. This function corresponds with `XmRemoveTabGroup`.

TrackingEvent(*cursor, confine_to*)
 Returns `(widget, event)`. This function corresponds with `XmTrackingEvent`.

TrackingLocate(*cursor, confine_to*)
 Returns `widget`. This function corresponds with `XmTrackingLocate`.

UpdateDisplay()
 Returns `None`. This function corresponds with `XmUpdateDisplay`.

4.3.1 Shell widget methods

Widget objects whose widget class is subclassed from `Shell` have the following additional method:

IsMotifWMRunning()
 Returns `success`. This function corresponds with `XmIsMotifWMRunning`.

4.3.2 VendorShell widget methods

Widget objects whose widget class is subclassed from `VendorShell` have the following additional methods (note that `VendorShell` is a subclass of `Shell`):

ActivateProtocol(*property, protocol*)
 Returns `None`. This function corresponds with `XmActivateProtocol`.

ActivateWMProtocol(*protocol*)
 Returns `None`. This function corresponds with `XmActivateWMProtocol`.

AddProtocolCallback(*property, protocol, cbfunc, cbarg*)
 Returns `None`. This function corresponds with `XmAddProtocolCallback`.

`AddProtocols(property, protocol[])`
 Returns None. This function corresponds with `XmAddProtocols`.

`AddWMPProtocolCallback(protocol, cbfunc, cbarg)`
 Returns None. This function corresponds with `XmAddWMPProtocolCallback`.

`AddWMPProtocols(protocol[])`
 Returns None. This function corresponds with `XmAddWMPProtocols`.

`DeactivateProtocol(property, protocol)`
 Returns None. This function corresponds with `XmDeactivateProtocol`.

`DeactivateWMPProtocol(protocol)`
 Returns None. This function corresponds with `XmDeactivateWMPProtocol`.

`RemoveProtocolCallback(property, protocol, callback, closure)`
 Returns None. This function corresponds with `XmRemoveProtocolCallback`.

`RemoveWMPProtocolCallback(protocol, callback, closure)`
 Returns None. This function corresponds with `XmRemoveWMPProtocolCallback`.

`SetProtocolHooks(property, protocol, prehook, pre_closure, posthook, post_closure)`
 Returns None. This function corresponds with `XmSetProtocolHooks`.

`SetWMPProtocolHooks(protocol, prehook, pre_closure, posthook, post_closure)`
 Returns None. This function corresponds with `XmSetWMPProtocolHooks`.

4.3.3 XmCascadeButton widget methods

Widget objects whose widget class is subclassed from `XmCascadeButton` or `XmCascadeButtonGadget` have the following additional method:

`CascadeButtonHighlight(highlight)`
 Returns None. This function corresponds with `XmCascadeButtonHighlight`.

4.3.4 XmCommand widget methods

Widget objects whose widget class is subclassed from `XmCommand` have the following additional methods:

`CommandAppendValue(command)`
 Returns None. This function corresponds with `XmCommandAppendValue`.

`CommandError(error)`
 Returns None. This function corresponds with `XmCommandError`.

`CommandGetChild(child)`
 Returns `child_widget`. This function corresponds with `XmCommandGetChild`.

`CommandSetValue(command)`
 Returns None. This function corresponds with `XmCommandSetValue`.

4.3.5 XmFileSelectionBox widget methods

Widget objects whose widget class is subclassed from XmFileSelectionBox have the following additional methods (note that XmFileSelectionBox is a subclass of XmSelectionBox):

`FileSelectionBoxGetChild(child)`
Returns `child_widget`. This function corresponds with `XmFileSelectionBoxGetChild`.

`FileSelectionDoSearch(dirmask)`
Returns `None`. This function corresponds with `XmFileSelectionDoSearch`.

4.3.6 XmList widget methods

Widget objects whose widget class is subclassed from XmList have the following additional methods:

`ListAddItem(item, position)`
Returns `None`. This function corresponds with `XmListAddItem`.

`ListAddItemUnselected(item, position)`
Returns `None`. This function corresponds with `XmListAddItemUnselected`.

`ListAddItems(item_list, position)`
Returns `None`. This function corresponds with `XmListAddItems`.

`ListAddItemsUnselected(item_list, position)`
Returns `None`. This function corresponds with `XmListAddItemsUnselected`.

`ListDeleteAllItems()`
Returns `None`. This function corresponds with `XmListDeleteAllItems`.

`ListDeleteItem(item)`
Returns `None`. This function corresponds with `XmListDeleteItem`.

`ListDeleteItems(item_list)`
Returns `None`. This function corresponds with `XmListDeleteItems`.

`ListDeleteItemsPos(item_count, position)`
Returns `None`. This function corresponds with `XmListDeleteItemsPos`.

`ListDeletePos(position)`
Returns `None`. This function corresponds with `XmListDeletePos`.

`ListDeletePositions(positionlist)`
Returns `None`. This function corresponds with `XmListDeletePositions`.

`ListDeselectAllItems()`
Returns `None`. This function corresponds with `XmListDeselectAllItems`.

`ListDeselectItem(item)`
Returns `None`. This function corresponds with `XmListDeselectItem`.

`ListDeselectPos(position)`
Returns None. This function corresponds with `XmListDeselectPos`.

`ListGetKbdItemPos()`
Returns position. This function corresponds with `XmListGetKbdItemPos`.

`ListGetSelectedPos()`
Returns `position_list` or None. This function corresponds with `XmListGetSelectedPos`.

`ListItemExists(item)`
Returns success. This function corresponds with `XmListItemExists`.

`ListItemPos(item)`
Returns position. This function corresponds with `XmListItemPos`.

`ListPosSelected(position)`
Returns selected. This function corresponds with `XmListPosSelected`.

`ListPosToBounds(position)`
Returns (x, y, width, height) or None. This function corresponds with `XmListPosToBounds`.

`ListReplaceItems(old_items_list, new_items_list)`
Returns None. This function corresponds with `XmListReplaceItems`.

`ListReplaceItemsPos(new_item_list, position)`
Returns None. This function corresponds with `XmListReplaceItemsPos`.

`ListReplaceItemsPosUnselected(new_item_list, position)`
Returns None. This function corresponds with `XmListReplaceItemsPosUnselected`.

`ListReplaceItemsUnselected(old_items_list, new_items_list)`
Returns None. This function corresponds with `XmListReplaceItemsUnselected`.

`ListReplacePositions(position_list, item_list)`
Returns None. This function corresponds with `XmListReplacePositions`.

`ListSelectItem(item, notify)`
Returns None. This function corresponds with `XmListSelectItem`.

`ListSelectPos(position, notify)`
Returns None. This function corresponds with `XmListSelectPos`.

`ListSetAddMode(state)`
Returns None. This function corresponds with `XmListSetAddMode`.

`ListSetBottomItem(item)`
Returns None. This function corresponds with `XmListSetBottomItem`.

`ListSetBottomPos(position)`
Returns None. This function corresponds with `XmListSetBottomPos`.

ListSetHorizPos(*position*)

Returns None. This function corresponds with XmListSetHorizPos.

ListSetItem(*item*)

Returns None. This function corresponds with XmListSetItem.

ListSetKbdItemPos(*position*)

Returns success. This function corresponds with XmListSetKbdItemPos.

ListSetPos(*position*)

Returns None. This function corresponds with XmListSetPos.

ListUpdateSelectedList()

Returns None. This function corresponds with XmListUpdateSelectedList.

ListYToPos(*y*)

Returns position. This function corresponds with XmListYToPos.

4.3.7 XmMainWindow widget methods

Widget objects whose widget class is subclassed from XmMainWindow have the following additional methods:

MainWindowSep1()

Returns widget. This function corresponds with XmMainWindowSep1.

MainWindowSep2()

Returns widget. This function corresponds with XmMainWindowSep2.

MainWindowSep3()

Returns widget. This function corresponds with XmMainWindowSep3.

MainWindowSetAreas(*menu_bar*, *command_window*, *horizontal_scrollbar*, *vertical_scrollbar*, *work_region*)

Returns None. This function corresponds with XmMainWindowSetAreas.

4.3.8 XmMessageBox widget methods

Widget objects whose widget class is subclassed from XmMessageBox have the following additional method:

MessageBoxGetChild(*child*)

Returns child_widget. This function corresponds with XmMessageBoxGetChild.

4.3.9 XmRowColumn widget methods

Widget objects whose widget class is subclassed from XmRowColumn have the following additional methods:

GetPostedFromWidget()

Returns widget. This function corresponds with XmGetPostedFromWidget.

`GetTearOffControl()`
Returns widget. This function corresponds with `XmGetTearOffControl`.

`MenuPosition(event)`
Returns None. This function corresponds with `XmMenuPosition`.

`OptionButtonGadget()`
Returns widget. This function corresponds with `XmOptionButtonGadget`.

`OptionLabelGadget()`
Returns widget. This function corresponds with `XmOptionLabelGadget`.

4.3.10 XmScale widget methods

Widget objects whose widget class is subclassed from `XmScale` have the following additional methods:

`ScaleGetValue()`
Returns value. This function corresponds with `XmScaleGetValue`.

`ScaleSetValue(value)`
Returns None. This function corresponds with `XmScaleSetValue`.

4.3.11 XmScrollBar widget methods

Widget objects whose widget class is subclassed from `XmScrollBar` have the following additional methods:

`ScrollBarGetValues()`
Returns (value, slider_size, increment, page_increment). This function corresponds with `XmScrollBarGetValues`.

`ScrollBarSetValues(value, slider_size, increment, page_increment, notify)`
Returns None. This function corresponds with `XmScrollBarSetValues`.

4.3.12 XmScrolledWindow widget methods

Widget objects whose widget class is subclassed from `XmScrolledWindow` have the following additional methods:

`ScrollVisible(widget, left_right_margin, top_bottom_margin)`
Returns None. This function corresponds with `XmScrollVisible`.

`ScrolledWindowSetAreas(horizontal_scrollbar, vertical_scrollbar, work_region)`
Returns None. This function corresponds with `XmScrolledWindowSetAreas`.

4.3.13 XmSelectionBox widget methods

Widget objects whose widget class is subclassed from `XmSelectionBox` have the following additional method:

SelectionBoxGetChild(*child*)

Returns *child_widget*. This function corresponds with `XmSelectionBoxGetChild`.

4.3.14 XmText widget methods

Widget objects whose widget class is subclassed from `XmText` have the following additional methods:

TextClearSelection(*time*)

Returns `None`. This function corresponds with `XmTextClearSelection`.

TextCopy(*time*)

Returns `None`. This function corresponds with `XmTextCopy`.

TextCut(*time*)

Returns `None`. This function corresponds with `XmTextCut`.

TextDisableRedisplay()

Returns `None`. This function corresponds with `XmTextDisableRedisplay`.

TextEnableRedisplay()

Returns `None`. This function corresponds with `XmTextEnableRedisplay`.

TextFindString(*start, string, direction*)

Returns position or `None`. This function corresponds with `XmTextFindString`.

TextGetBaseline()

Returns *baseline*. This function corresponds with `XmTextGetBaseline`.

TextGetEditable()

Returns *state*. This function corresponds with `XmTextGetEditable`.

TextGetInsertionPosition()

Returns *position*. This function corresponds with `XmTextGetInsertionPosition`.

TextGetLastPosition()

Returns *position*. This function corresponds with `XmTextGetLastPosition`.

TextGetMaxLength()

Returns *length*. This function corresponds with `XmTextGetMaxLength`.

TextGetSelection()

Returns *text*. This function corresponds with `XmTextGetSelection`.

TextGetSelectionPosition()

Returns (*owner, left, right*). This function corresponds with `XmTextGetSelectionPosition`.

TextGetSource()

Returns *source*. This function corresponds with `XmTextGetSource`.

TextGetString()
Returns text. This function corresponds with XmTextGetString.

TextGetTopCharacter()
Returns position. This function corresponds with XmTextGetTopCharacter.

TextInsert(*position*, *value*)
Returns None. This function corresponds with XmTextInsert.

TextPaste()
Returns *is_owner*. This function corresponds with XmTextPaste.

TextPosToXY(*position*)
Returns (*x*, *y*) or None. This function corresponds with XmTextPosToXY.

TextRemove()
Returns success. This function corresponds with XmTextRemove.

TextReplace(*from_pos*, *to_pos*, *value*)
Returns None. This function corresponds with XmTextReplace.

TextScroll(*lines*)
Returns None. This function corresponds with XmTextScroll.

TextSetAddMode(*state*)
Returns None. This function corresponds with XmTextSetAddMode.

TextSetEditable(*editable*)
Returns None. This function corresponds with XmTextSetEditable.

TextSetHighlight(*left*, *right*, *mode*)
Returns None. This function corresponds with XmTextSetHighlight.

TextSetInsertionPosition(*position*)
Returns None. This function corresponds with
XmTextSetInsertionPosition.

TextSetMaxLength(*max_length*)
Returns None. This function corresponds with XmTextSetMaxLength.

TextSetSelection(*first*, *last*, *time*)
Returns None. This function corresponds with XmTextSetSelection.

TextSetSource(*source*, *top_character*, *cursor_position*)
Returns None. This function corresponds with XmTextSetSource.

TextSetString(*value*)
Returns None. This function corresponds with XmTextSetString.

TextSetTopCharacter(*top_character*)
Returns None. This function corresponds with XmTextSetTopCharacter.

TextShowPosition(*position*)
Returns None. This function corresponds with XmTextShowPosition.

TextXYToPos(*x*, *y*)
Returns None. This function corresponds with XmTextXYToPos.

4.3.15 XmTextField widget methods

Widget objects whose widget class is subclassed from XmTextField have the following additional methods:

TextFieldClearSelection(*time*)

Returns None. This function corresponds with XmTextFieldClearSelection.

TextFieldCopy(*time*)

Returns None. This function corresponds with XmTextFieldCopy.

TextFieldCut(*time*)

Returns None. This function corresponds with XmTextFieldCut.

TextFieldGetBaseline()

Returns baseline. This function corresponds with XmTextFieldGetBaseline.

TextFieldGetEditable()

Returns state. This function corresponds with XmTextFieldGetEditable.

TextFieldGetInsertionPosition()

Returns position. This function corresponds with XmTextFieldGetInsertionPosition.

TextFieldGetLastPosition()

Returns position. This function corresponds with XmTextFieldGetLastPosition.

TextFieldGetMaxLength()

Returns length. This function corresponds with XmTextFieldGetMaxLength.

TextFieldGetSelection()

Returns text. This function corresponds with XmTextFieldGetSelection.

TextFieldGetSelectionPosition()

Returns (owner, left, right). This function corresponds with XmTextFieldGetSelectionPosition.

TextFieldGetString()

Returns text. This function corresponds with XmTextFieldGetString.

TextFieldInsert(*position*, *value*)

Returns None. This function corresponds with XmTextFieldInsert.

TextFieldPaste()

Returns is_owner. This function corresponds with XmTextFieldPaste.

TextFieldPosToXY(*position*)

Returns (x, y) or None. This function corresponds with XmTextFieldPosToXY.

TextFieldRemove()

Returns success. This function corresponds with XmTextFieldRemove.

`TextFieldReplace(from_pos, to_pos, value)`
 Returns None. This function corresponds with `XmTextFieldReplace`.

`TextFieldSetAddMode(state)`
 Returns None. This function corresponds with `XmTextFieldSetAddMode`.

`TextFieldSetEditable(editable)`
 Returns None. This function corresponds with `XmTextFieldSetEditable`.

`TextFieldSetHighlight(left, right, mode)`
 Returns None. This function corresponds with `XmTextFieldSetHighlight`.

`TextFieldSetInsertionPosition(position)`
 Returns None. This function corresponds with `XmTextFieldSetInsertionPosition`.

`TextFieldSetMaxLength(max_length)`
 Returns None. This function corresponds with `XmTextFieldSetMaxLength`.

`TextFieldSetSelection(first, last, time)`
 Returns None. This function corresponds with `XmTextFieldSetSelection`.

`TextFieldSetString(value)`
 Returns None. This function corresponds with `XmTextFieldSetString`.

`TextFieldShowPosition(position)`
 Returns None. This function corresponds with `XmTextFieldShowPosition`.

`TextFieldXYToPos(x, y)`
 Returns None. This function corresponds with `XmTextFieldXYToPos`.

4.3.16 XmToggleButton widget methods

Widget objects whose widget class is subclassed from `XmToggleButton` have the following additional methods:

`ToggleButtonGetState()`
 Returns state. This function corresponds with `XmToggleButtonGetState`.

`ToggleButtonSetState(state, notify)`
 Returns None. This function corresponds with `XmToggleButtonSetState`.

4.3.17 XmToggleButtonGadget widget methods

Widget objects whose widget class is subclassed from `XmToggleButtonGadget` have the following additional methods:

`ToggleButtonGadgetGetState()`
 Returns state. This function corresponds with `XmToggleButtonGadgetGetState`.

`ToggleButtonGadgetSetState(state, notify)`
 Returns None. This function corresponds with `XmToggleButtonGadgetSetState`.

XmToggleButtonGadgetSetState.

Chapter 5

Built-in module Xaw

5.1 Xaw functions and variables

The module defines the following variables:

Clock
Widget class object.

Command
Widget class object.

Dialog
Widget class object.

Form
Widget class object.

Grip
Widget class object.

Label
Widget class object.

List
Widget class object.

Logo
Widget class object.

Mailbox
Widget class object.

MenuButton
Widget class object.

Paned
Widget class object.

Panner
Widget class object.

Porthole
Widget class object.

Repeater
Widget class object.

Scrollbar
Widget class object.

Simple
Widget class object.

SimpleMenu
Widget class object.

StripChart
Widget class object.

Text
Widget class object.

Toggle
Widget class object.

Tree
Widget class object.

Viewport
Widget class object.

Chapter 6

Built-in module Glx

6.1 Glx functions and variables

The module defines the following functions and variables:

`Draw`

Widget class object

`makeconfig(list)`

Returns int.

`winset(glx_draw_widget)`

Returns None. This function corresponds with GLXwinset.

Chapter 7

Built-in module HTML

7.1 HTML functions and variables

The module defines the following functions and variables:

`anchor_cbarg(WbAnchorCallbackData)`
Returns (event, element_id, text, href).

`form_cbarg(WbFormCallbackData)`
Returns (event, href, method, enctype, attr_list).

`html`
Widget class object.

7.2 HTML Widget object

Widget objects whose widget class is subclassed from `html` have the following additional methods:

`AnchorPosition(name)`
Returns (success, x, y). This function corresponds with `HTMLAnchorToPosition`.

`AnchorToId(name)`
Returns id. This function corresponds with `HTMLAnchorToId`.

`ClearSelection()`
Returns None. This function corresponds with `HTMLClearSelection`.

`FreeImageInfo()`
Returns None. This function corresponds with `HTMLFreeImageInfo`.

`GetHRefs()`
Returns href_list. This function corresponds with `HTMLGetHRefs`.

`GetImageSrcs()`
Returns srcs_list. This function corresponds with `HTMLGetImageSrcs`.

`GetText(pretty)`
Returns `text`. This function corresponds with `HTMLGetText`.

`GetTextAndSelection()`
Returns `(text, start, end, insert)`. This function corresponds with `HTMLGetTextAndSelection`.

`GotoId(element_id)`
Returns `None`. This function corresponds with `HTMLGotoId`.

`IdToPosition(element_id)`
Returns `(success, x, y)`. This function corresponds with `HTMLIdToPosition`.

`PositionToId(x, y)`
Returns `element_id`. This function corresponds with `HTMLPositionToId`.

`SearchText(pattern, start, backward, caseless)`
Returns `(page, (start_id, start_pos), (end_id, end_pos))`. This function corresponds with `HTMLSearchText`.

`SetSelection((start_id, start_pos), (end_id, end_pos))`
Returns `None`. This function corresponds with `HTMLSetSelection`.

`SetText(text, header_text, footer_text)`
Returns `None`. This function corresponds with `HTMLSetText`.

Index

- ActivateProtocol (VendorShell widget method), 27
- ActivateWMPProtocol (VendorShell widget method), 27
- AddCallback (Widget object attribute), 9
- AddEventHandler (Widget object attribute), 9
- AddInput (in module Xt), 7
- AddProtocolCallback (VendorShell widget method), 27
- AddProtocols (VendorShell widget method), 28
- AddRawEventHandler (Widget object attribute), 9
- AddTabGroup (Xm Widget object attribute), 23
- AddTimeOut (in module Xt), 7
- AddWMPProtocolCallback (VendorShell widget method), 28
- AddWMPProtocols (VendorShell widget method), 28
- AddWorkProc (in module Xt), 7
- all_chars_exist (Font object attribute), 19
- AllocColor (Colormap object attribute), 17
- AllocColorCells (Colormap object attribute), 17
- AllocNamedColor (Colormap object attribute), 18
- anchor_cbarg (in module HTML), 41
- AnchorPosition (HTML Widget object attribute), 41
- AnchorToId (HTML Widget object attribute), 41
- ApplicationShell (in module Xt), 7
- arc_mode (Graphics Context object attribute), 16
- ArrowButtonGadget (in module Xm), 21
- ascent (Font object attribute), 19
- background (Graphics Context object attribute), 16
- bits_per_rgb (Visual Info object attribute), 13
- blue_mask (Visual Info object attribute), 13
- BuildEventMask (Widget object attribute), 9
- BulletinBoard (in module Xm), 21
- c_class (Visual Info object attribute), 13
- CallCallbacks (Widget object attribute), 9
- cap_style (Graphics Context object attribute), 16
- CascadeButton (in module Xm), 21
- CascadeButtonGadget (in module Xm), 21
- CascadeButtonHighlight (XmCascadeButton widget method), 28
- ChangeColor (Xm Widget object attribute), 23
- ChangeGC (Graphics Context object attribute), 14
- ChangeWindowAttributes (Widget object attribute), 9
- CheckMaskEvent (Display object attribute), 5
- Class (Widget object attribute), 9
- class (Visual Info object attribute), 13
- ClearSelection (HTML Widget object attribute), 41

clip_mask (Graphics Context object attribute), 16
clip_x_origin (Graphics Context object attribute), 16
clip_y_origin (Graphics Context object attribute), 16
Clock (in module Xaw), 38
colormap_size (Visual Info object attribute), 13
Command (in module Xaw), 38
Command (in module Xm), 21
CommandAppendValue (XmCommand widget method), 28
CommandError (XmCommand widget method), 28
CommandGetChild (XmCommand widget method), 28
CommandSetValue (XmCommand widget method), 28
Composite (in module Xt), 7
ConnectionNumber (Display object attribute), 5
Constraint (in module Xt), 7
Convert (Widget object attribute), 9
ConvertUnits (Xm Widget object attribute), 23
CopyArea (Pixmap object attribute), 18
CopyPlane (Pixmap object attribute), 18
CreateApplicationShell (in module Xt), 7
CreateArrowButton (Xm Widget object attribute), 23
CreateArrowButtonGadget (Xm Widget object attribute), 23
CreateBulletinBoard (Xm Widget object attribute), 23
CreateBulletinBoardDialog (Xm Widget object attribute), 23
CreateCascadeButton (Xm Widget object attribute), 24
CreateCascadeButtonGadget (Xm Widget object attribute), 24
CreateColormap (Visual Info object attribute), 13
CreateColormap (Widget object attribute), 9
CreateCommand (Xm Widget object attribute), 24
CreateDialogShell (Xm Widget object attribute), 24
CreateDrawingArea (Xm Widget object attribute), 24
CreateDrawnButton (Xm Widget object attribute), 24
CreateErrorDialog (Xm Widget object attribute), 24
CreateFileSelectionBox (Xm Widget object attribute), 24
CreateFileSelectionDialog (Xm Widget object attribute), 24
CreateFontCursor (Display object attribute), 5
CreateForm (Xm Widget object attribute), 24
CreateFormDialog (Xm Widget object attribute), 24
CreateFrame (Xm Widget object attribute), 24
CreateGC (Pixmap object attribute), 18
CreateGC (Widget object attribute), 9
CreateImage (Visual Info object attribute), 13
CreateInformationDialog (Xm Widget object attribute), 24
CreateLabel (Xm Widget object attribute), 24
CreateLabelGadget (Xm Widget object attribute), 24
CreateList (Xm Widget object attribute), 24
CreateMainWindow (Xm Widget object attribute), 24
CreateManagedWidget (Widget object attribute), 9
CreateMenuBar (Xm Widget object attribute), 24
CreateMenuShell (Xm Widget object attribute), 25

CreateMessageBox (Xm Widget object attribute), 25
 CreateMessageDialog (Xm Widget object attribute), 25
 CreateOptionMenu (Xm Widget object attribute), 25
 CreatePanedWindow (Xm Widget object attribute), 25
 CreatePixmap (Widget object attribute), 9
 CreatePopupMenu (Xm Widget object attribute), 25
 CreatePopupShell (Widget object attribute), 10
 CreatePromptDialog (Xm Widget object attribute), 25
 CreatePulldownMenu (Xm Widget object attribute), 25
 CreatePushButton (Xm Widget object attribute), 25
 CreatePushButtonGadget (Xm Widget object attribute), 25
 CreateQuestionDialog (Xm Widget object attribute), 25
 CreateRadioBox (Xm Widget object attribute), 25
 CreateRowColumn (Xm Widget object attribute), 25
 CreateScale (Xm Widget object attribute), 25
 CreateScrollBar (Xm Widget object attribute), 25
 CreateScrolledList (Xm Widget object attribute), 25
 CreateScrolledText (Xm Widget object attribute), 25
 CreateScrolledWindow (Xm Widget object attribute), 25
 CreateSelectionBox (Xm Widget object attribute), 25
 CreateSelectionDialog (Xm Widget object attribute), 26
 CreateSeparator (Xm Widget object attribute), 26
 CreateSeparatorGadget (Xm Widget object attribute), 26
 CreateSimpleCheckBox (Xm Widget object attribute), 26
 CreateSimpleMenuBar (Xm Widget object attribute), 26
 CreateSimpleOptionMenu (Xm Widget object attribute), 26
 CreateSimplePopupMenu (Xm Widget object attribute), 26
 CreateSimplePulldownMenu (Xm Widget object attribute), 26
 CreateSimpleRadioBox (Xm Widget object attribute), 26
 CreateTemplateDialog (Xm Widget object attribute), 26
 CreateText (Xm Widget object attribute), 26
 CreateTextField (Xm Widget object attribute), 26
 CreateToggleButton (Xm Widget object attribute), 26
 CreateToggleButtonGadget (Xm Widget object attribute), 26
 CreateWarningDialog (Xm Widget object attribute), 26
 CreateWidget (Widget object attribute), 10
 CreateWorkArea (Xm Widget object attribute), 26
 CreateWorkingDialog (Xm Widget object attribute), 26
 dash_offset (Graphics Context object attribute), 16
 DeactivateProtocol (VendorShell widget method), 28
 DeactivateWMPProtocol (VendorShell widget method), 28
 default_char (Font object attribute), 19
 DefaultColormap (Widget object attribute), 10
 DefaultColormapOfScreen (Widget object attribute), 10
 DefaultVisual (Widget object attribute), 10
 DefaultVisualOfScreen (Widget object attribute), 10

DefineCursor (Widget object attribute), 10
 depth (Visual Info object attribute), 13
 descent (Font object attribute), 19
 DestroyWidget (Widget object attribute), 10
 Dialog (in module Xaw), 38
 DialogShell (in module Xm), 21
 direction (Font object attribute), 19
 DispatchEvent (in module Xt), 7
 Display (Widget object attribute), 10
 Display (in module Xm), 21
 Draw (in module Glx), 40
 DrawArc (Graphics Context object attribute), 14
 DrawArcs (Graphics Context object attribute), 14
 DrawImageString (Graphics Context object attribute), 14
 DrawingArea (in module Xm), 21
 DrawLine (Graphics Context object attribute), 14
 DrawLines (Graphics Context object attribute), 14
 DrawnButton (in module Xm), 21
 DrawPoint (Graphics Context object attribute), 14
 DrawPoints (Graphics Context object attribute), 14
 DrawRectangle (Graphics Context object attribute), 15
 DrawRectangles (Graphics Context object attribute), 15
 DrawSegments (Graphics Context object attribute), 15
 DrawString (Graphics Context object attribute), 15

 Error (in module Xlib), 5
 Error (in module Xt), 7

 fid (Font object attribute), 19
 file_style (Graphics Context object attribute), 16

 FileSelectionBox (in module Xm), 21
 FileSelectionBoxGetChild (XmFileSelectionBox widget method), 29
 FileSelectionDoSearch (XmFileSelectionBox widget method), 29
 fill_rule (Graphics Context object attribute), 16
 FillArc (Graphics Context object attribute), 15
 FillArcs (Graphics Context object attribute), 15
 FillPolygon (Graphics Context object attribute), 15
 FillRectangle (Graphics Context object attribute), 15
 FillRectangles (Graphics Context object attribute), 15
 Flush (Display object attribute), 5
 font (Graphics Context object attribute), 16
 foreground (Graphics Context object attribute), 17
 Form (in module Xaw), 38
 Form (in module Xm), 22
 form_cbarg (in module HTML), 41
 Frame (in module Xm), 22
 FreeColors (Colormap object attribute), 18
 FreeImageInfo (HTML Widget object attribute), 41
 function (Graphics Context object attribute), 17

 Gadget (in module Xm), 22
 GetConstraintResourceList (Widget Class object attribute), 9
 GetDisplay (Xm Widget object attribute), 26
 getevent (in module Xlib), 5
 GetFocusWidget (Xm Widget object attribute), 27
 GetGC (Widget object attribute), 10
 GetGeometry (Widget object attribute), 10

GetHRefs (HTML Widget object attribute), 41
 GetImageSrcs (HTML Widget object attribute), 41
 GetPostedFromWidget (XmRowColumn widget method), 31
 GetResourceList (Widget Class object attribute), 9
 GetSecondaryResourceData (Xm Widget Class object attribute), 23
 GetTabGroup (Xm Widget object attribute), 27
 GetTearOffControl (XmRowColumn widget method), 32
 GetText (HTML Widget object attribute), 42
 GetTextAndSelection (HTML Widget object attribute), 42
 GetValues (Widget object attribute), 10
 GetVisibility (Xm Widget object attribute), 27
 GetVisualInfo (Display object attribute), 6
 GetVisualInfo (Widget object attribute), 10
 GetWMNormalHints (Widget object attribute), 10
 Glx (built-in module), 40
 GotoId (HTML Widget object attribute), 42
 GrabButton (Widget object attribute), 10
 graphics_exposures (Graphics Context object attribute), 17
 green_mask (Visual Info object attribute), 13
 Grip (in module Xaw), 38
 HasCallbacks (Widget object attribute), 10
 HeightMMOfScreen (Widget object attribute), 10
 HeightOfScreen (Widget object attribute), 10
 HTML (built-in module), 41
 html (in module HTML), 41
 IdToPosition (HTML Widget object attribute), 42
 Initialize (in module Xt), 7
 Install (Colormap object attribute), 18
 InstallAccelerators (Widget object attribute), 11
 InstallAllAccelerators (Widget object attribute), 11
 InstallColormap (Colormap object attribute), 18
 IsComposite (Widget object attribute), 11
 IsManaged (Widget object attribute), 11
 IsMotifWMRunning (Shell widget method), 27
 IsRealized (Widget object attribute), 11
 IsSensitive (Widget object attribute), 11
 IsSubclass (Widget object attribute), 11
 IsTraversable (Xm Widget object attribute), 27
 join_style (Graphics Context object attribute), 17
 Label (in module Xaw), 38
 Label (in module Xm), 22
 LabelGadget (in module Xm), 22
 line_style (Graphics Context object attribute), 17
 line_width (Graphics Context object attribute), 17
 List (in module Xaw), 38
 List (in module Xm), 22
 ListAddItem (XmList widget method), 29
 ListAddItems (XmList widget method), 29
 ListAddItemsUnselected (XmList widget method), 29
 ListAddItemUnselected (XmList widget method), 29

ListDeleteAllItems (XmList widget method), 29
ListDeleteItem (XmList widget method), 29
ListDeleteItems (XmList widget method), 29
ListDeleteItemsPos (XmList widget method), 29
ListDeletePos (XmList widget method), 29
ListDeletePositions (XmList widget method), 29
ListDeselectAllItems (XmList widget method), 29
ListDeselectItem (XmList widget method), 29
ListDeselectPos (XmList widget method), 30
ListFonts (Widget object attribute), 11
ListGetKbdItemPos (XmList widget method), 30
ListGetSelectedPos (XmList widget method), 30
ListItemExists (XmList widget method), 30
ListItemPos (XmList widget method), 30
ListPosSelected (XmList widget method), 30
ListPosToBounds (XmList widget method), 30
ListReplaceItems (XmList widget method), 30
ListReplaceItemsPos (XmList widget method), 30
ListReplaceItemsPosUnselected (XmList widget method), 30
ListReplaceItemsUnselected (XmList widget method), 30
ListReplacePositions (XmList widget method), 30
ListSelectItem (XmList widget method), 30
ListSelectPos (XmList widget method), 30
ListSetAddMode (XmList widget method), 30
ListSetBottomItem (XmList widget method), 30
ListSetBottomPos (XmList widget method), 30
ListSetHorizPos (XmList widget method), 30
ListSetItem (XmList widget method), 31
ListSetKbdItemPos (XmList widget method), 31
ListSetPos (XmList widget method), 31
ListUpdateSelectedList (XmList widget method), 31
ListYToPos (XmList widget method), 31
LoadQueryFont (Widget object attribute), 11
Logo (in module Xaw), 38
LookupColor (Colormap object attribute), 18
LookupString (in module Xlib), 5
LowerWindow (Widget object attribute), 11
Mailbox (in module Xaw), 38
MainLoop (in module Xt), 8
MainWindow (in module Xm), 22
MainWindowSep1 (XmMainWindow widget method), 31
MainWindowSep2 (XmMainWindow widget method), 31
MainWindowSep3 (XmMainWindow widget method), 31
MainWindowSetAreas (XmMainWindow widget method), 31
makeconfig (in module Glx), 40
ManageChild (Widget object attribute), 11
ManageChildren (in module Xt), 8
MapWidget (Widget object attribute), 11
max_bounds (Font object attribute), 19

max_byte1 (Font object attribute),
19
max_char_or_byte2 (Font object
attribute), 19
MenuButton (in module Xaw), 38
MenuPosition (XmRowColumn widget
method), 32
MenuShell (in module Xm), 22
MessageBox (in module Xm), 22
MessageBoxGetChild (XmMessageBox
widget method), 31
min_bounds (Font object attribute),
19
min_byte1 (Font object attribute),
19
min_char_or_byte2 (Font object
attribute), 20

n_properties (Font object
attribute), 20
NameToWidget (Widget object
attribute), 11
NextEvent (Display object
attribute), 6
NextEvent (in module Xt), 8

OpenDisplay (in module Xt), 8
OptionButtonGadget (XmRowColumn
widget method), 32
OptionLabelGadget (XmRowColumn
widget method), 32
OverrideShell (in module Xt), 8

Paned (in module Xaw), 38
PanedWindow (in module Xm), 22
Panner (in module Xaw), 38
Parent (Widget object attribute),
11
ParseColor (Colormap object
attribute), 18
PeekEvent (Display object
attribute), 6
PeekEvent (in module Xt), 8
Pending (in module Xt), 8
per_char (Font object attribute),
20
plane_mask (Graphics Context object
attribute), 17

Popdown (Widget object attribute),
11
Popup (Widget object attribute), 11
Porthole (in module Xaw), 39
PositionToId (HTML Widget object
attribute), 42
ProcessEvent (in module Xt), 8
ProcessTraversal (Xm Widget object
attribute), 27
properties (Font object attribute),
20
PushButton (in module Xm), 22
PushButtonGadget (in module Xm), 22
PutBackEvent (Display object
attribute), 6
PutImage (Graphics Context object
attribute), 15

QueryColor (Colormap object
attribute), 18
QueryColors (Colormap object
attribute), 18
QueryPointer (Widget object
attribute), 11

RaiseWindow (Widget object
attribute), 11
ReadBitmapFile (Pixmap object
attribute), 19
ReadBitmapFile (Widget object
attribute), 11
RealizeWidget (Widget object
attribute), 12
red_mask (Visual Info object
attribute), 13
RemoveAllCallbacks (Widget object
attribute), 12
RemoveCallback (Widget object
attribute), 12
RemoveEventHandler (Widget object
attribute), 12
RemoveInput (in module Xt), 8
RemoveProtocolCallback (VendorShell
widget method), 28
RemoveRawEventHandler (Widget
object attribute), 12

RemoveTabGroup (Xm Widget object attribute), 27
 RemoveTimeOut (in module Xt), 8
 RemoveWMPProtocolCallback (VendorShell widget method), 28
 RemoveWorkProc (in module Xt), 8
 Repeater (in module Xaw), 39
 RowColumn (in module Xm), 22

 Scale (in module Xm), 22
 ScaleGetValue (XmScale widget method), 32
 ScaleSetValue (XmScale widget method), 32
 ScrollBar (in module Xm), 22
 Scrollbar (in module Xaw), 39
 ScrollBarGetValues (XmScrollBar widget method), 32
 ScrollBarSetValues (XmScrollBar widget method), 32
 ScrolledWindow (in module Xm), 22
 ScrolledWindowSetAreas (XmScrolledWindow widget method), 32
 ScrollVisible (XmScrolledWindow widget method), 32
 SearchText (HTML Widget object attribute), 42
 SelectionBox (in module Xm), 22
 SelectionBoxGetChild (XmSelectionBox widget method), 33
 Separator (in module Xm), 22
 SeparatorGadget (in module Xm), 22
 SetArcMode (Graphics Context object attribute), 15
 SetBackground (Graphics Context object attribute), 15
 SetClipMask (Graphics Context object attribute), 15
 SetClipOrigin (Graphics Context object attribute), 15
 SetClipRectangles (Graphics Context object attribute), 15
 SetDashes (Graphics Context object attribute), 15
 SetFillRule (Graphics Context object attribute), 15
 SetFillStyle (Graphics Context object attribute), 15
 SetFont (Graphics Context object attribute), 15
 SetForeground (Graphics Context object attribute), 16
 SetFunction (Graphics Context object attribute), 16
 SetGraphicsExposures (Graphics Context object attribute), 16
 SetLineAttributes (Graphics Context object attribute), 16
 SetMappedWhenManaged (Widget object attribute), 12
 SetPlaneMask (Graphics Context object attribute), 16
 SetProtocolHooks (VendorShell widget method), 28
 SetSelection (HTML Widget object attribute), 42
 SetSensitive (Widget object attribute), 12
 SetState (Graphics Context object attribute), 16
 SetStipple (Graphics Context object attribute), 16
 SetSubwindowMode (Graphics Context object attribute), 16
 SetText (HTML Widget object attribute), 42
 SetTile (Graphics Context object attribute), 16
 SetTSOrigin (Graphics Context object attribute), 16
 SetValues (Widget object attribute), 12
 SetWindowColormap (Widget object attribute), 12
 SetWMNormalHints (Widget object attribute), 12
 SetWMPProtocolHooks (VendorShell widget method), 28
 Shell (in module Xt), 8
 Simple (in module Xaw), 39

SimpleMenu (in module Xaw), 39

stipple (Graphics Context object attribute), 17

StoreColors (Colormap object attribute), 18

StripChart (in module Xaw), 39

subwindow_mode (Graphics Context object attribute), 17

Superclass (Widget object attribute), 12

Text (in module Xaw), 39

Text (in module Xm), 23

TextClearSelection (XmText widget method), 33

TextCopy (XmText widget method), 33

TextCut (XmText widget method), 33

TextDisableRedisplay (XmText widget method), 33

TextEnableRedisplay (XmText widget method), 33

TextExtents (Font object attribute), 19

TextField (in module Xm), 23

TextFieldClearSelection (XmTextField widget method), 35

TextFieldCopy (XmTextField widget method), 35

TextFieldCut (XmTextField widget method), 35

TextFieldGetBaseline (XmTextField widget method), 35

TextFieldGetEditable (XmTextField widget method), 35

TextFieldGetInsertionPosition (XmTextField widget method), 35

TextFieldGetLastPosition (XmTextField widget method), 35

TextFieldGetMaxLength (XmTextField widget method), 35

TextFieldGetSelection (XmTextField widget method), 35

TextFieldGetSelectionPosition (XmTextField widget method), 35

TextFieldGetString (XmTextField widget method), 35

TextFieldInsert (XmTextField widget method), 35

TextFieldPaste (XmTextField widget method), 35

TextFieldPosToXY (XmTextField widget method), 35

TextFieldRemove (XmTextField widget method), 35

TextFieldReplace (XmTextField widget method), 35

TextFieldSetAddMode (XmTextField widget method), 36

TextFieldSetEditable (XmTextField widget method), 36

TextFieldSetHighlight (XmTextField widget method), 36

TextFieldSetInsertionPosition (XmTextField widget method), 36

TextFieldSetMaxLength (XmTextField widget method), 36

TextFieldSetSelection (XmTextField widget method), 36

TextFieldSetString (XmTextField widget method), 36

TextFieldShowPosition (XmTextField widget method), 36

TextFieldXYToPos (XmTextField widget method), 36

TextFindString (XmText widget method), 33

TextGetBaseline (XmText widget method), 33

TextGetEditable (XmText widget method), 33

TextGetInsertionPosition (XmText widget method), 33

TextGetLastPosition (XmText widget method), 33

TextGetMaxLength (XmText widget method), 33

TextGetSelection (XmText widget method), 33

TextGetSelectionPosition (XmText widget method), 33
 TextGetSource (XmText widget method), 33
 TextGetString (XmText widget method), 34
 TextGetTopCharacter (XmText widget method), 34
 TextInsert (XmText widget method), 34
 TextPaste (XmText widget method), 34
 TextPosToXY (XmText widget method), 34
 TextRemove (XmText widget method), 34
 TextReplace (XmText widget method), 34
 TextScroll (XmText widget method), 34
 TextSetAddMode (XmText widget method), 34
 TextSetEditable (XmText widget method), 34
 TextSetHighlight (XmText widget method), 34
 TextSetInsertionPosition (XmText widget method), 34
 TextSetMaxLength (XmText widget method), 34
 TextSetSelection (XmText widget method), 34
 TextSetSource (XmText widget method), 34
 TextSetString (XmText widget method), 34
 TextSetTopCharacter (XmText widget method), 34
 TextShowPosition (XmText widget method), 34
 TextWidth (Font object attribute), 19
 TextXYToPos (XmText widget method), 34
 tile (Graphics Context object attribute), 17
 Toggle (in module Xaw), 39
 ToggleButton (in module Xm), 23
 ToggleButtonGadget (in module Xm), 23
 ToggleButtonGadgetGetState (XmToggleButtonGadget widget method), 36
 ToggleButtonGadgetSetState (XmToggleButtonGadget widget method), 36
 ToggleButtonGetState (XmToggleButton widget method), 36
 ToggleButtonSetState (XmToggleButton widget method), 36
 ToolkitInitialize (in module Xt), 8
 TopLevelShell (in module Xt), 8
 TrackingEvent (Xm Widget object attribute), 27
 TrackingLocate (Xm Widget object attribute), 27
 TransientShell (in module Xt), 8
 TranslateCoords (Widget object attribute), 12
 Tree (in module Xaw), 39
 ts_x_origin (Graphics Context object attribute), 17
 ts_y_origin (Graphics Context object attribute), 17
 UndefineCursor (Widget object attribute), 12
 UngrabButton (Widget object attribute), 12
 UnmanageChild (Widget object attribute), 12
 UnmapWidget (Widget object attribute), 12
 UnrealizeWidget (Widget object attribute), 12
 UpdateDisplay (Xm Widget object attribute), 27
 VendorShell (in module Xt), 8
 Viewport (in module Xaw), 39
 visual (Visual Info object attribute), 13

visualid (Visual Info object
attribute), 13

WidthMMOfScreen (Widget object
attribute), 12

WidthOfScreen (Widget object
attribute), 13

Window (Widget object attribute),
13

winset (in module Glx), 40

WMShell (in module Xt), 8

Xaw (built-in module), 38

Xlib (built-in module), 5

Xm (built-in module), 21

Xt (built-in module), 7