

Appendix A - Function Summaries

This quick reference is intended to help you find and use the right function for a particular task. It organises the Section 1 and Section 3 reference pages into two lists:

- List of functions and macros by groups.
- Alphabetical list of functions and macros

The first column indicates which section to find the routines in; the required Section is given in brackets after the function name.

A.1 Group Listing with Brief Descriptions

Atoms

XmGetAtomName(1)	Get the string representation of an Atom
XmInternAtom(1)	Return an Atom for a given property name string

CascadeButton

XmCascadeButtonGadgetHighlight(1)	Set the highlight state of a CascadeButtonGadget
XmCascadeButtonHighlight(1)	Set the highlight state of a CascadeButton

Clipboard

XmClipboardBeginCopy(1)	Set up storage for a clipboard copy operation
XmClipboardCancelCopy(1)	Cancel a copy operation to the clipboard
XmClipboardCopy(1)	Copy a data item to temporary storage for later copying to the clipboard
XmClipboardCopyByName(1)	Copy as data item passed by name
XmClipboardEndCopy(1)	End a copy operation to the clipboard
XmClipboardEndRetrieve(1)	End a copy operation from the clipboard
XmClipboardInquireCount(1)	Get the number of data item formats available on the clipboard
XmClipboardInquireFormat(1)	Get the specified clipboard data format name
XmClipboardInquireLength(1)	Get the length of the data item on the clipboard

Appendix A: Function Summaries

XmClipboardInquirePendingItems(1)	Get a list of pending data ID/private ID pairs
XmClipboardLock(1)	Lock the clipboard
XmClipboardRegisterFormat(1)	Register a new format for clipboard data items
XmClipboardRetrieve(1)	Retrieve a data item from the clipboard
XmClipboardStartCopy(1)	Set up storage for a clipboard copy operation
XmClipboardStartRetrieve(1)	Start a clipboard retrieval operation
XmClipboardUndoCopy(1)	Remove the last item copied to the clipboard
XmClipboardUnlock(1)	Unlock the clipboard
XmClipboardWithdrawFormat(1)	Indicate that an application does not want to supply a data item any longer

Colors

XmChangeColor(1)	Update the colors of a widget
XmGetColorCalculation(1)	Get the procedure that calculates default widget colors
XmGetColors(1)	Get the foreground, select, and shadow colors for a widget
XmSetColorCalculation(1)	Set the procedure that calculates default widget colors

ComboBox

XmComboBoxAddItem(1)	Add a compound string to the ComboBox List
XmComboBoxDeletePos(1)	Delete an item at the specified position from a ComboBox List
XmComboBoxSelectItem(1)	Select an item from a ComboBox List
XmComboBoxSetItem(1)	Select and make visible an item from a ComboBox List
XmComboBoxUpdate(1)	Update the ComboBox List after changes to component widgets

Command

XmCommandAppendValue(1)	Append a compound string to a Command widget
XmCommandError(1)	Display an error message in a Command widget

Appendix A: Function Summaries

XmCommandGetChild(1)	Get the specified child of a Command widget
XmCommandSetValue(1)	Replace the Command string

Compound Strings

XmCvtByteStreamToXmString()	Convert a byte stream to a compound string
XmCvtCTToXmString(1)	Convert compound text to a compound string
XmCvtTextPropertyToXmStringTable(1)	Convert an XTextProperty to a compound string table
XmCvtXmStringTableToTextProperty(1)	Convert a compound string table to an XTextProperty
XmCvtXmStringToByteStream(1)	Convert a compound string to byte stream format
XmCvtXmStringToCT(1)	Convert a compound string to compound text
XmMapSegmentEncoding(1)	Get the compound text encoding format for a font list element tag
XmRegisterSegmentEncoding(1)	Register a compound text encoding format for a font list element tag
XmStringBaseline(1)	Get the baseline spacing for a compound string
XmStringByteCompare(1)	Compare two compound strings byte-by-byte
XmStringByteStreamLength(1)	Calculate the length of a byte stream
XmStringCompare(1)	Compare two compound strings
XmStringComponentCreate(1)	Create a compound string component
XmStringConcat(1)	Concatenate two compound strings
XmStringConcatAndFree(1)	Create a new compound string formed by concatenating two compound strings, freeing the two strings afterwards
XmStringCopy(1)	Copy a compound string
XmStringCreate(1)	Create a compound string
XmStringCreateLocalized(1)	Create a compound string in the current locale
XmStringCreateLtoR(1)	Create a compound string
XmStringCreateSimple(1)	Create a compound string in the current language environment
XmStringDirectionCreate(1)	Create a compound string containing a direction component

Appendix A: Function Summaries

XmStringDirectionToDirection(1)	Convert a string direction to a direction
XmStringDraw(1)	Draw a compound string
XmStringDrawImage(1)	Draw a compound string
XmStringDrawUnderline(1)	Draw a compound string with an underlined sub-string
XmStringEmpty(1)	Determine whether there are any text components in a compound string
XmStringExtent(1)	Get the smallest rectangle enclosing a compound string
XmStringFree(1)	Free the memory used by a compound string
XmStringFreeContext(1)	Free a compound string context
XmStringGenerate(1)	Generate a compound string
XmStringGetLtoR(1)	Get a text segment from a compound string
XmStringGetNextComponent(1)	Retrieves information about the next compound string component
XmStringGetNextSegment(1)	Retrieves information about the next compound string component
XmStringGetNextTriple(1)	Retrieve information about the next compound string segment
XmStringHasSubstring(1)	Determine whether a compound string contains a sub-string
XmStringHeight(1)	Get the line height of a compound string
XmStringInitContext(1)	Create a compound string context
XmStringIsVoid(1)	Determine whether there are valid segments in a compound string
XmStringLength(1)	Get the length of a compound string
XmStringLineCount(1)	Get the number of lines in a compound string
XmStringNConcat(1)	Concatenate a specified portion of a compound string to another compound string
XmStringNCopy(1)	Copy a specified portion of a compound string
XmStringParseText(1)	Convert a string to a compound string
XmStringPeekNextComponent(1)	Returns the type of the next compound string component
XmStringPutRendition(1)	Add rendition components to a compound string
XmStringSegmentCreate(1)	Create a compound string segment

Appendix A: Function Summaries

XmStringSeparatorCreate(1)	Create a compound string containing a separator component
XmStringTableParseStringArray(1)	Convert an array of strings into a compound string table
XmStringTableProposeTabList(1)	Create a Tab list for a compound string
XmStringTableToXmString(1)	Convert a compound string table to a compound string
XmStringTableUnparse(1)	Convert a compound string table to a string
XmStringToXmStringTable(1)	Convert a compound string to a compound string table
XmStringUnparse(1)	Convert a compound string to a string
XmStringWidth(1)	Return the width of a compound string

Container

XmContainerCopy(1)	Copy the Container primary selection to the clipboard
XmContainerCopyLink(1)	Copy links to the Container primary selection into the clipboard
XmContainerCut(1)	Cuts the Container primary selection onto the clipboard
XmContainerGetItemChild(1)	Find the children of a Container item
XmContainerPaste(1)	Pastes the clipboard selection into a Container
XmContainerPasteLink(1)	Copies links from the clipboard selection into a Container
XmContainerRelayout(1)	Force relayout of a Container widget
XmContainerReorder(1)	Reorder children of a Container

Cursors

XmGetMenuCursor(1)	Get the current menu cursor
XmSetMenuCursor(1)	Set the current menu cursor

DataField

XmDataFieldCopy	Copy the primary selection to the clipboard
XmDataFieldCut	Copy the primary selection to the clipboard and remove the selected text
XmDataFieldGetSelection	Get the value of the primary selection
XmDataFieldGetSelectionPosition	Get the position of the primary selection

Appendix A: Function Summaries

XmDataFieldGetString	Set the text string
XmDataFieldGetStringWcs	Set the current wide-character text string
XmDataFieldSetAddMode	Set the add mode state
XmDataFieldSetEditable	Set the edit permission state
XmDataFieldSetInsertionPosition	Set the position of the insertion cursor
XmDataFieldSetHighlight	Specify the DataField widget
XmDataFieldSetSelection	Set the value of the primary selection
XmDataFieldSetString	Set the text string
XmDataFieldShowPosition	Display the text at a specified position
XmDataFieldXYToPos	Get the character position for an x, y position

Direction

XmDirectionMatch(1)	Compare two directions
XmDirectionMatchPartial(1)	Partially compare two directions
XmDirectionToStringDirection(1)	Convert a direction to a string direction
XmStringDirectionToDirection(1)	Convert a string direction to a direction

Display

XmGetXmDisplay(1)	Get the XmDisplay object for the Display
XmUpdateDisplay(1)	Update the display

Drag and Drop

XmDragCancel(1)	Cancel a drag operation
XmDragStart(1)	Start a drag operation
XmDropSiteConfigureStackingOrder(1)	Change the stacking order of a drop site
XmDropSiteEndUpdate(1)	End an update of multiple drop site operations
XmDropSiteQueryStackingOrder(1)	Get the stacking order of a drop site
XmDropSiteRegister(1)	Register a widget as a drop site
XmDropSiteRetrieve(1)	Get the resource values of a drop site
XmDropSiteStartUpdate(1)	Start an update of multiple drop site operations
XmDropSiteUnregister(1)	Remove a drop site
XmDropSiteUpdate(1)	Change the resource values for a drop site
XmDropTransferAdd(1)	Add drop transfer entries to a drop operation
XmDropTransferStart(1)	Start a drop operation

Appendix A: Function Summaries

XmGetDragContext(1)	Get information about a drag-and-drop operation
XmTargetsAreCompatible(1)	Determine whether or not the target types of a drag source and a drop site match

DropDown

XmDropDownGetArrow(1)	Return the “arrow” child of the XmDropDown
XmDropDownGetChild(1)	Get the specified child of a DropDown widget
XmDropDownGetLabel(1)	Return the “label” child of the XmDropDown
XmDropDownGetList(1)	Return the “list” child of the XmDropDown
XmDropDownGetText(1)	Return the “arrow” child of the XmDropDown
XmDropDownGetValue(1)	Retrieve the value from DropDown.
XmIconBoxIsCellEmpty(1)	Determine whether a cell in the IconBox is empty

FileSelectionBox

XmFileSelectionBoxGetChild(1)	Get a specified child of a FileSelection-Box widget
XmFileSelectionDoSearch(1)	Start a directory search

Font Lists

XmFontListAdd(1)	Create a new font list
XmFontListAppendEntry(1)	Append a font entry to a font list
XmFontListCopy(1)	Copy a font list
XmFontListCreate(1)	Create a new font list
XmFontListCreate_r(1)	Create a new font list in a thread-safe manner
XmFontListEntryCreate(1)	Create a new font list entry
XmFontListEntryCreate_r(1)	Create a new font list entry in a thread-safe manner
XmFontListEntryFree(1)	Free the memory used by a font list entry
XmFontListEntryGetFont(1)	Get the font information from a font list entry

Appendix A: Function Summaries

XmFontListEntryGetTag(1)	Get the tag of a font list entry
XmFontListEntryLoad(1)	Load a font or create a font set, and then create a font list entry
XmFontListFree(1)	Free the memory used by a font list
XmFontListFreeFontContext(1)	Free a font context
XmFontListGetNextFont(1)	Retrieve information about the next font list element
XmFontListInitFontContext(1)	Create a font context
XmFontListNextEntry(1)	Retrieve the next font list entry in a font list
XmFontListRemoveEntry(1)	Remove a font list entry from a font list
XmSetFontUnit(1)	Set the font unit values
XmSetFontUnits(1)	Set the font unit values

Hierarchy

XmHierarchyGetChildNodes(1)	Return the list of “node children” of the node widget
XmHieararchyOpenAllAncestors(1)	Open all ancestors of a given node

Keyboard Handling

XmTranslateKey(1)	Convert a KeyCode to a KeySym using the default translator
-------------------	--

Keyboard Traversal

XmAddTabGroup(1)	Add a widget to a tab group list
XmGetDestination(1)	Get the current destination widget
XmGetFocusWidget(1)	Get the widget that has the keyboard focus
XmGetTabGroup(1)	Get the tab group for a widget
XmGetVisibility(1)	Determine whether or not a widget is visible
XmIsTraversable(1)	Determine whether or not a widget can receive the keyboard focus
XmProcessTraversal(1)	Set the widget which has the keyboard focus
XmRemoveTabGroup(1)	Remove a widget from a tab group list

Input Methods

XmImCloseXIM(1)	Close all input contexts associated with the input method
-----------------	---

Appendix A: Function Summaries

XmImFreeXIC(1)	Free an input context
XmImGetXIC(1)	Create an input context for a widget
XmImGetXIM(1)	Retrieve the input method for a widget
XmImMbLookupString(1)	Retrieve a compound string from an input method
XmImMbResetIC(1)	Reset an input context
XmImRegister(1)	Register an input context for a widget
XmImSetFocusValues(1)	Set the values and focus for an input context
XmImSetValues(1)	Set the values for an input context
XmImSetXIC(1)	Register a widget with an existing input context
XmImUnregister(1)	Unregister the input context for a widget
XmImUnsetFocus(1)	Unset focus for an input context
XmImVaSetFocusValues(1)	Set the values and focus for an input context
XmImVaSetValues(1)	Set the values for an input context

List

XmListAddItem(1)	Add an item to a List
XmListAddItems(1)	Add items to a List
XmListAddItemUnselected(1)	Add an item to a List, without selecting the item
XmListAddItemsUnselected(1)	Add items to a List, without selecting the items
XmListDeleteAllItems(1)	Delete all of the items from a List
XmListDeleteItem(1)	Delete an item from a List
XmListDeleteItems(1)	Delete items from a List
XmListDeleteItemsPos(1)	Delete items starting at a specified position from a List
XmListDeletePos(1)	Delete an item at a specified position from a List
XmListDeletePositions(1)	Delete items at specified positions from a List
XmListDeselectAllItems(1)	Deselect all items in a List
XmListDeselectPos(1)	Deselect an item at a specified position in a List
XmListGetKbdItemPos(1)	Get the position of the item that has the location cursor in a List
XmListGetMatchPos(1)	Get all occurrences of an item in a List
XmListItemExists(1)	Determine if a specified item is in a List

Appendix A: Function Summaries

<code>XmListItemPos(1)</code>	Determine the position of an item in a List
<code>XmListPosSelected(1)</code>	Check if the item at the specified position in the List is selected
<code>XmListPosToBounds(1)</code>	Return the bounding box of the item at the specified position in a List
<code>XmListReplaceItems(1)</code>	Replace specified items in a List
<code>XmListReplaceItemsPos(1)</code>	Replace items at specified positions in a List
<code>XmListReplaceItemsPosUnselected(1)</code>	Replace items at specified positions in a List, without selecting the replacement items
<code>XmListReplaceItemsUnselected(1)</code>	Replace items in the List, without selecting the replacement items
<code>XmListReplacePositions(1)</code>	Replace items at the specified positions in a List
<code>XmListSelectedPos(1)</code>	Get the positions of the selected items in a List
<code>XmListSelectItem(1)</code>	Select a specified item in a List
<code>XmListSelectPos(1)</code>	Select the item at the specified position in a List
<code>XmListSetAddMode(1)</code>	Set add mode in a List
<code>XmListSetBottomItem(1)</code>	Set the last visible item in a List
<code>XmListSetBottomPos(1)</code>	Set the last visible position in a List
<code>XmListSetHorizPos(1)</code>	Set the horizontal position of a List
<code>XmListSetItem(1)</code>	Set the first visible item in a List
<code>XmListSetKbdItemPos(1)</code>	Set the position of the location cursor in a List
<code>XmListSetPos(1)</code>	Set the first visible position in a List
<code>XmListUpdateSelectedList(1)</code>	Update the set of selected items in a List
<code>XmListYToPos(1)</code>	Get the position of the item at the specified y-coordinate in a List

MainWindow

<code>XmMainWindowSep1(1)</code>	Get the widget ID of the first MainWindow Separator
<code>XmMainWindowSep2(1)</code>	Get the widget ID of the second MainWindow Separator
<code>XmMainWindowSep3(1)</code>	Get the widget ID of the third MainWindow Separator
<code>XmMainWindowSetAreas(1)</code>	Specify the children for a MainWindow

Appendix A: Function Summaries

Menus

XmGetPostedFromWidget(1)	Get the widget that posted a menu
XmGetTearOffControl(1)	Get the tear-off control for a Menu
XmMenuPosition(1)	Position a popup menu
XmOptionButtonGadget(1)	Get the CascadeButtonGadget in an OptionMenu
XmOptionLabelGadget(1)	Get the LabelGadget in an OptionMenu
XmRemoveFromPostFromList(1)	Make a menu inaccessible from a widget

MessageBox

XmMessageBoxGetChild(1)	Retrieve a specified child of a Message-Box widget
-------------------------	--

Mrm Functions

MrmCloseHierarchy(3)	Close an Mrm hierarchy
MrmFetchBitmapLiteral(3)	Retrieve an exported bitmap from an Mrm hierarchy
MrmFetchColorLiteral(3)	Retrieve an exported color value from an Mrm hierarchy
MrmFetchIconLiteral(3)	Retrieve an exported icon from an Mrm hierarchy
MrmFetchLiteral(3)	Retrieve an exported value from an Mrm hierarchy
MrmFetchSetValues(3)	Set widget resources to values retrieved from an Mrm hierarchy
MrmFetchWidget(3)	Create the widget tree rooted at a named widget
MrmFetchWidgetOverride(3)	Create the widget tree rooted at a named widget, and override the resources set in the UID file
MrmInitialize(3)	Prepare the Mrm library for use
MrmOpenHierarchy(3)	Open an Mrm hierarchy
MrmOpenHierarchyFromBuffer(3)	Open an Mrm hierarchy
MrmOpenHierarchyPerDisplay(3)	Open an Mrm hierarchy
MrmRegisterClass(3)	Register a widget creation routine for a non-Motif component
MrmRegisterNames(3)	Register application-defined values and procedures

Appendix A: Function Summaries

MrmRegisterNamesInHierarchy(3)	Register application-defined values and procedures for use in a specific UIL hierarchy
--------------------------------	--

MultiList

XmMultiListDeselectItems(1)	Clear the selection state of items of a MultiList
XmMultiListDeselectRow(1)	Clear the selection state of a specified row of a MultiList widget
XmMultiListGetSelectedRowArray(1)	Returns NULL-terminated array of pointers to selected rows of MultiList
XmMultiListGetSelectedRows(1)	Find the rows that are currently selected
XmMultiListMakeRowVisible(1)	Make row of a MultiList widget visible
XmMultiListSelectAllItems(1)	Set the selection state on all rows of a MultiList
XmMultiListSelectItems(1)	Set the selection state of items of a MultiList
XmMultiListSelectRow(1)	Set the selection state of a specified row of a MultiList widget
XmMultiListToggleRow(1)	Toggle the selection state of a specified row of a MultiList widget.
XmMultiListUnselectAllItems(1)	Deselects all rows of the list
XmMultiListUnselectItem(1)	Deselects the specified item of the list

Notebook

XmNotebookGetPageInfo(1)	Return information about a Notebook page
--------------------------	--

Paned

XmPanedGetPanels(1)	Retrieve the panes in the widget
---------------------	----------------------------------

Parse Mapping

XmParseMappingCreate(1)	Create a parse mapping
XmParseMappingFree(1)	Free the memory used by a parse mapping
XmParseMappingGetValues(1)	Fetch resources from a parse mapping object
XmParseMappingSetValues(1)	Set resources for a parse mapping object
XmParseTableFree(1)	Free the memory used by a parse table

Appendix A: Function Summaries

Pixmaps

XmDestroyPixmap(1)	Remove a pixmap from the pixmap cache
XmGetPixmap(1)	Create and return a pixmap
XmGetPixmapByDepth(1)	Create and return a pixmap of the specified depth
XmGetScaledPixmap(1)	Create and return a scaled pixmap
XmInstallImage(1)	Install an image in the image cache
XmUninstallImage(1)	Remove an image from the image cache

Print

XmPrintPopupPDM(1)	Notify the Print Display Manager
XmPrintSetup(1)	Create a PrintShell widget
XmPrintToFile(1)	Save X print Server data to file
XmRedisplayWidget(1)	Force widget exposure for printing

Protocols

XmActivateProtocol(1)	Activate a protocol
XmActivateWMProtocol(1)	Activate the XA_WM_PROTOCOLS protocol
XmAddProtocolCallback(1)	Add client callbacks to a protocol
XmAddProtocols(1)	Add protocols to the protocol manager
XmAddWMProtocolCallback(1)	Add client callbacks to an XA_WM_PROTOCOLS protocol
XmAddWMProtocols(1)	Add the XA_WM_PROTOCOLS protocol to the protocol manager
XmDeactivateProtocol(1)	Deactivate a protocol
XmDeactivateWMProtocol(1)	Deactivate the XA_WM_PROTOCOLS protocol
XmRemoveProtocolCallback(1)	Remove client callback from a protocol
XmRemoveProtocols(1)	Remove protocols from the protocol manager
XmRemoveWMProtocolCallback(1)	Remove client callbacks from the XA_WM_PROTOCOLS protocol
XmRemoveWMProtocols(1)	Remove the XA_WM_PROTOCOLS protocol from the protocol manager
XmSetProtocolHooks(1)	Set prehooks and posthooks for a protocol
XmSetWMProtocolHooks(1)	Set prehooks and posthooks for the XA_WM_PROTOCOLS protocol

Render Tables and Renditions

XmRenderTableAddRenditions(1)	Add renditions to a render table
XmRenderTableCopy(1)	Copy a render table
XmRenderTableCvtFromProp(1)	Convert from a string representation into a render table
XmRenderTableCvtToProp(1)	Convert a render table into a string representation
XmRenderTableFree(1)	Free the memory used by a render table
XmRenderTableGetRendition(1)	Search a render table for a matching rendition
XmRenderTableGetRenditions(1)	Search a render table for a matching renditions
XmRenderTableGetTags(1)	Fetch the list of rendition tags from a render table
XmRenderTableRemoveRenditions(1)	Copy a render table, excluding specified renditions
XmRenditionCreate(1)	Create a rendition object
XmRenditionFree(1)	Free the memory used by a rendition object
XmRenditionRetrieve(1)	Fetch rendition object resources
XmRenditionUpdate(1)	Set rendition object resources

Resolution Independence

XmConvertUnits(1)	Convert a value to a specified unit type
-------------------	--

Resource Conversion

XmConvertStringToUnits(1)	Convert a string to an integer, optionally translating the units
XmCvtStringToUnitType(1)	Convert a string to a unit-type value
XmRepTypeAddReverse(1)	Install the reverse converter for a representation type
XmRepTypeGetId(1)	Get the ID number of a representation type
XmRepTypeGetNameList(1)	Get the list of value names for a representation type
XmRepTypeGetRecord(1)	Get the information about a representation type
XmRepTypeGetRegistered(1)	Get the registered representation types
XmRepTypeInstallTearOffModelConverter(1)	

Appendix A: Function Summaries

XmRepTypeRegister(1)
XmRepTypeValidValue(1)

Install the resource converter for
XmNtearOffModel
Register a representation type resource
Determine the validity of a numerical
value for a representation type

Scale

XmScaleGetValue(1)
XmScaleSetTicks(1)
XmScaleSetValue(1)

Get the slider value for a Scale widget
Set tick marks for a Scale widget
Set the slider value for a Scale widget

Screen

XmGetXmScreen(1)

Get the XmScreen object for a screen

ScrollBar

XmScrollBarGetValues(1)
XmScrollBarSetValues(1)

Get information about the current state
of a ScrollBar
Set the current state of a ScrollBar
widget

ScrolledWindow

XmScrolledWindowSetAreas(1)
XmScrollVisible(1)

Set the children for a ScrolledWindow
Make an obscured child of a Scrolled-
Window visible

SelectionBox

XmSelectionBoxGetChild(1)

Get the specified child of a Selection-
Box widget

SpinBox

XmSimpleSpinBoxAddItem(1)
XmSimpleSpinBoxDeletePos(1)

XmSimpleSpinBoxSetItem(1)
XmSpinBoxValidatePosition(1)

Add an item to a SimpleSpinBox
Delete an item at a specified position
from a SimpleSpinBox
Set an item in a SimpleSpinBox
Validate the current value of a SpinBox

Tab Lists

XmTabCreate(1)

Create a tab (XmTab) object

Appendix A: Function Summaries

XmTabFree(1)	Free the memory used by an XmTab object
XmTabGetValues(1)	Fetch the value of an XmTab object
XmTabListCopy(1) ¹	Copy a tab list object
XmTabListFree(1)	Free the memory used by a tab list
XmTabListGetTab(1)	Retrieve a tab from a tab list
XmTabListInsertTabs(1)	Insert tabs into a tab list
XmTabListRemoveTabs(1)	Copy a tab list, excluding specified tabs
XmTabListReplacePositions(1)	Copy a tab list, replacing tabs at specified positions
XmTabListTabCount(1)	Count the number of tabs in a tab list
XmTabSetValue(1)	Set the value of a tab object

TabStack

XmTabStackGetSelectedTab(1)	Return the widget ID of the currently selected tab
XmTabStackIndexToWidget(1)	Return the widget ID of the tab in the TabStack
XmTabStackSelectTab(1)	Set the currently displayed child of the TabStack
XmTabStackXYToWidget(1)	Convert a pixel coordinate in the TabStack's window to the widget of the tab.

Text

XmTextClearSelection(1)	Clear the primary selection
XmTextCopy(1)	Copy the primary selection to the clipboard
XmTextCopyLink(1)	Copy links from the primary selection to the clipboard
XmTextCut(1)	Copy the primary selection to the clipboard, and remove the selected text
XmTextDisableRedisplay(1)	Prevent visual update
XmTextEnableRedisplay(1)	Allow visual update
XmTextFindString(1)	Find the beginning position of a text string
XmTextFindStringWcs(1)	Find the beginning position of a wide-character text string
XmTextGetBaseline(1)	Get the position of the baseline

1. Erroneously given as XmTabListTabCopy() in 2nd edition

Appendix A: Function Summaries

XmTextGetCenterline(1)	Get the height of vertical text
XmTextGetCursorPosition(1)	Get the position of the location cursor
XmTextGetEditable(1)	Get the edit permission state
XmTextGetInsertionPosition(1)	Get the position of the insertion cursor
XmTextGetLastPosition(1)	Get the position of the last character
XmTextGetMaxLength(1)	Get the maximum possible length of a text string
XmTextGetSelection(1)	Get the value of the primary selection
XmTextGetSelectionPosition(1)	Get the position of the primary selection
XmTextGetSelectionWcs(1)	Get the wide-character value of the primary selection
XmTextGetSource(1)	Get the text source
XmTextGetString(1)	Get the text string
XmTextGetStringWcs(1)	Get the wide-character text string
XmTextGetSubstring(1)	Get a copy of part of the text string
XmTextGetSubstringWcs(1)	Get a copy of part of the wide-character text string
XmTextGetTopCharacter(1)	Get the position of the first visible character
XmTextInsert(1)	Insert a string into the text string
XmTextInsertWcs(1)	Insert a wide-character string into the text string
XmTextPaste(1)	Insert the clipboard selection
XmTextPasteLink(1)	Insert links from the clipboard selection
XmTextPosToXY(1)	Get the x, y coordinates of a character position
XmTextRermove(1)	Delete the primary selection
XmTextReplace(1)	Replace part of the text string
XmTextReplaceWcs(1)	Replace part of the wide-character text string
XmTextScroll(1)	Scroll the text
XmTextSetAddMode(1)	Set the add mode state
XmTextSetCursorPosition(1)	Set the position of the location cursor
XmTextSetEditable(1)	Set the edit permission state
XmTextSetHighlight(1)	Highlight part of the text string
XmTextSetInsertionPosition(1)	Set the position of the insertion cursor
XmTextSetMaxLength(1)	Set the maximum possible length of a text string
XmTextSetSelection(1)	Set the value of the primary selection
XmTextSetSource(1)	Set the text source
XmTextSetString(1)	Set the text string
XmTextSetStringWcs(1)	Set the wide-character text string

<code>XmTextSetTopCharacter(1)</code>	Set the position of the first visible character
<code>XmTextShowPosition(1)</code>	Scroll the text so that a specified position is visible
<code>XmTextXYToPos(1)</code>	Get the character position for a given x, y coordinate

TextField

<code>XmTextFieldClearSelection(1)</code>	Clear the primary selection
<code>XmTextFieldCopy(1)</code>	Copy the primary selection to the clipboard
<code>XmTextFieldCopyLink(1)</code>	Copy links from the primary selection to the clipboard
<code>XmTextFieldCut(1)</code>	Copy the primary selection to the clipboard, and remove the selected text
<code>XmTextFieldGetBaseline(1)</code>	Get the position of the baseline
<code>XmTextFieldGetCursorPosition(1)</code>	Get the position of the location cursor
<code>XmTextFieldGetEditable(1)</code>	Get the edit permission state
<code>XmTextFieldGetInsertionPosition(1)</code>	Get the position of the insertion cursor
<code>XmTextFieldGetLastPosition(1)</code>	Get the position of the last character
<code>XmTextFieldGetMaxLength(1)</code>	Get the maximum possible length of a text string
<code>XmTextFieldGetSelection(1)</code>	Get the value of the primary selection
<code>XmTextFieldGetSelectionPosition(1)</code>	Get the position of the primary selection
<code>XmTextFieldGetSelectionWcs(1)</code>	Get the wide-character value of the primary selection
<code>XmTextFieldGetString(1)</code>	Get the text string
<code>XmTextFieldGetStringWcs(1)</code>	Get the wide-character text string
<code>XmTextFieldGetSubstring(1)</code>	Get a copy of part of the text string
<code>XmTextFieldGetSubstringWcs(1)</code>	Get a copy of part of the wide-character text string
<code>XmTextFieldGetTopCharacter(1)</code>	Get the position of the first visible character
<code>XmTextFieldInsert(1)</code>	Insert a string into the text string
<code>XmTextFieldInsertWcs(1)</code>	Insert a wide-character string into the text string
<code>XmTextFieldPaste(1)</code>	Insert the clipboard selection
<code>XmTextFieldPasteLink(1)</code>	Insert links from the clipboard selection
<code>XmTextFieldPosToXY(1)</code>	Get the x, y coordinates of a character position
<code>XmTextFieldRremove(1)</code>	Delete the primary selection

Appendix A: Function Summaries

XmTextFieldReplace(1)	Replace part of the text string
XmTextFieldReplaceWcs(1)	Replace part of the wide-character text string
XmTextFieldSetAddMode(1)	Set the add mode state
XmTextFieldSetCursorPosition(1)	Set the position of the location cursor
XmTextFieldSetEditable(1)	Set the edit permission state
XmTextFieldSetHighlight(1)	Highlight part of the text string
XmTextFieldSetInsertionPosition(1)	Set the position of the insertion cursor
XmTextFieldSetMaxLength(1)	Set the maximum possible length of a text string
XmTextFieldSetSelection(1)	Set the value of the primary selection
XmTextFieldSetSource(1)	Set the text source
XmTextFieldSetString(1)	Set the text string
XmTextFieldSetStringWcs(1)	Set the wide-character text string
XmTextFieldShowPosition(1)	Scroll the text so that a specified position is visible
XmTextFieldXYToPos(1)	Get the character position for a given x, y coordinate

ToggleButton

XmToggleButtonGadgetGetState(1)	Get the state of a ToggleButtonGadget
XmToggleButtonGadgetSetState(1)	Set the state of a ToggleButtonGadget
XmToggleButtonGadgetSetValue(1)	Set the value of a ToggleButtonGadget
XmToggleButtonGetState(1)	Get the state of a ToggleButton
XmToggleButtonSetState(1)	Set the state of a ToggleButton
XmToggleButtonSetValue(1)	Set the value of a ToggleButton

Uniform Transfer Model

XmTransferDone(1)	Complete a data transfer operation
XmTransferSendRequest(1)	Send a multiple transfer request
XmTransferSetParameters(1)	Set parameters for the next transfer operation
XmTransferStartRequest(1)	Initiate a multiple transfer batch operation
XmTransferValue(1)	Transfer data to a destination

Widget Class

XmIsObject(1)	Determine whether a widget is a subclass of a given class
---------------	---

Widget Creation

<code>XmCreateObject(1)</code>	Create an instance of a particular widget class or compound object
<code>XmPrintSetup(1)</code>	Create a <code>PrintShell</code> widget
<code>XmVaCreateSimpleCheckBox(1)</code>	Create a <code>CheckBox</code> compound object
<code>XmVaCreateSimpleMenuBar(1)</code>	Create a <code>MenuBar</code> compound object
<code>XmVaCreateSimpleOptionMenu(1)</code>	Create an <code>OptionMenu</code> compound object
<code>XmVaCreateSimplePopupMenu(1)</code>	Create a <code>PopupMenu</code> compound object as the child of a <code>MenuShell</code>
<code>XmVaCreateSimplePulldownMenu(1)</code>	Create a <code>PulldownMenu</code> compound object as the child of a <code>MenuShell</code>
<code>XmVaCreateSimpleRadioBox(1)</code>	Create a <code>RadioBox</code> compound object

Widget Internals

<code>XmGetSecondaryResourceData(1)</code>	Retrieve secondary widget resource data
<code>XmResolveAllPartOffset(1)</code>	Ensure upward-compatible widgets and applications
<code>XmResolvePartOffsets(1)</code>	Ensure upward-compatible widgets and applications

Widget Layout

<code>XmObjectAtPoint(1)</code>	Determine the child nearest to a point coordinate
<code>XmWidgetGetBaselines(1)</code>	Get the positions of the baselines in a widget
<code>XmWidgetGetDisplayRect(1)</code>	Get the display rectangle for a widget

Widget Selection

<code>XmTrackingEvent(1)</code>	Allow for modal selection of a component
<code>XmTrackingLocate(1)</code>	Allow for modal selection of a component

Window Manager

<code>XmIsMotifWMRunning(1)</code>	Check whether the Motif Window Manager (mwm) is running
------------------------------------	---

Appendix A: Function Summaries

A.2 Alphabetical Listing

MrmCloseHierarchy(3)	Close an Mrm hierarchy
MrmFetchBitmapLiteral(3)	Retrieve an exported bitmap from an Mrm hierarchy
MrmFetchColorLiteral(3)	Retrieve an exported color value from an Mrm hierarchy
MrmFetchIconLiteral(3)	Retrieve an exported icon from an Mrm hierarchy
MrmFetchLiteral(3)	Retrieve an exported value from an Mrm hierarchy
MrmFetchSetValues(3)	Set widget resources to values retrieved from an Mrm hierarchy
MrmFetchWidget(3)	Create the widget tree rooted at a named widget
MrmFetchWidgetOverride(3)	Create the widget tree rooted at a named widget, and override the resources set in the UID file
MrmInitialize(3)	Prepare the Mrm library for use
MrmOpenHierarchy(3)	Open an Mrm hierarchy
MrmOpenHierarchyFromBuffer(3)	Open an Mrm hierarchy
MrmOpenHierarchyPerDisplay(3)	Open an Mrm hierarchy
MrmRegisterClass(3)	Register a widget creation routine for a non-Motif component
MrmRegisterNames(3)	Register application-defined values and procedures
MrmRegisterNamesInHierarchy(3)	Register application-defined values and procedures for use in a specific UIL hierarchy
XmActivateProtocol(1)	Activate a protocol
XmActivateWMProtocol(1)	Activate the XA_WM_PROTOCOLS protocol
XmAddProtocolCallback(1)	Add client callbacks to a protocol
XmAddProtocols(1)	Add protocols to the protocol manager
XmAddTabGroup(1)	Add a widget to a tab group list
XmAddWMProtocolCallback(1)	Add client callbacks to an XA_WM_PROTOCOLS protocol
XmAddWMProtocols(1)	Add the XA_WM_PROTOCOLS protocol to the protocol manager
XmCascadeButtonGadgetHighlight(1)	Set the highlight state of a CascadeButtonGadget

Appendix A: Function Summaries

XmCascadeButtonHighlight(1)	Set the highlight state of a CascadeButton
XmChangeColor(1)	Update the colors of a widget
XmClipboardBeginCopy(1)	Set up storage for a clipboard copy operation
XmClipboardCancelCopy(1)	Cancel a copy operation to the clipboard
XmClipboardCopy(1)	Copy a data item to temporary storage for later copying to the clipboard
XmClipboardCopyByName(1)	Copy as data item passed by name
XmClipboardEndCopy(1)	End a copy operation to the clipboard
XmClipboardEndRetrieve(1)	End a copy operation from the clipboard
XmClipboardInquireCount(1)	Get the number of data item formats available on the clipboard
XmClipboardInquireFormat(1)	Get the specified clipboard data format name
XmClipboardInquireLength(1)	Get the length of the data item on the clipboard
XmClipboardInquirePendingItems(1)	Get a list of pending data ID/private ID pairs
XmClipboardLock(1)	Lock the clipboard
XmClipboardRegisterFormat(1)	Register a new format for clipboard data items
XmClipboardRetrieve(1)	Retrieve a data item from the clipboard
XmClipboardStartCopy(1)	Set up storage for a clipboard copy operation
XmClipboardStartRetrieve(1)	Start a clipboard retrieval operation
XmClipboardUndoCopy(1)	Remove the last item copied to the clipboard
XmClipboardUnlock(1)	Unlock the clipboard
XmClipboardWithdrawFormat(1)	Indicate that an application does not want to supply a data item any longer
XmComboBoxAddItem(1)	Add a compound string to the ComboBox List
XmComboBoxDeletePos(1)	Delete an item at the specified position from a ComboBox List
XmComboBoxSelectItem(1)	Select an item from a ComboBox List
XmComboBoxSetItem(1)	Select and make visible an item from a ComboBox List
XmComboBoxUpdate(1)	Update the ComboBox List after changes to component widgets
XmCommandAppendValue(1)	Append a compound string to a Command widget

Appendix A: Function Summaries

XmCommandError(1)	Display an error message in a Command widget
XmCommandGetChild(1)	Get the specified child of a Command widget
XmCommandSetValue(1)	Replace the Command string
XmContainerCopy(1)	Copy the Container primary selection to the clipboard
XmContainerCopyLink(1)	Copy links to the Container primary selection into the clipboard
XmContainerCut(1)	Cuts the Container primary selection onto the clipboard
XmContainerGetItemChild(1)	Find the children of a Container item
XmContainerPaste(1)	Pastes the clipboard selection into a Container
XmContainerPasteLink(1)	Copies links from the clipboard selection into a Container
XmContainerRelayout(1)	Force relayout of a Container widget
XmContainerReorder(1)	Reorder children of a Container
XmConvertStringToUnits(1)	Convert a string to an integer, optionally translating the units
XmConvertUnits(1)	Convert a value to a specified unit type
XmCreateObject(1)	Create an instance of a particular widget class or compound object
XmCvtByteStreamToXmString()	Convert a byte stream to a compound string
XmCvtCTToXmString(1)	Convert compound text to a compound string
XmCvtStringToUnitType(1)	Convert a string to a unit-type value
XmCvtTextPropertyToXmStringTable(1)	Convert an XTextProperty to a compound string table
XmCvtXmStringTableToTextProperty(1)	Convert a compound string table to an XTextProperty
XmCvtXmStringToByteStream(1)	Convert a compound string to byte stream format
XmCvtXmStringToCT(1)	Convert a compound string to compound text
XmDataFieldCopy	Copy the primary selection to the clipboard
XmDataFieldCut	Copy the primary selection to the clipboard and remove the selected text
XmDataFieldGetSelection	Get the value of the primary selection
XmDataFieldGetSelectionPosition	Get the position of the primary selection

Appendix A: Function Summaries

XmDataFieldGetString	Set the text string
XmDataFieldGetStringWcs	Set the current wide-character text string
XmDataFieldSetAddMode	Set the add mode state
XmDataFieldSetEditable	Set the edit permission state
XmDataFieldSetInsertionPosition	Set the position of the insertion cursor
XmDataFieldSetHighlight	Specify the DataField widget
XmDataFieldSetSelection	Set the value of the primary selection
XmDataFieldSetString	Set the text string
XmDataFieldShowPosition	Display the text at a specified position
XmDataFieldXYToPos	Get the character position for an x, y position
XmDeactivateProtocol(1)	Deactivate a protocol
XmDeactivateWMProtocol(1)	Deactivate the XA_WM_PROTOCOLS protocol
XmDestroyPixmap(1)	Remove a pixmap from the pixmap cache
XmDirectionMatch(1)	Compare two directions
XmDirectionMatchPartial(1)	Partially compare two directions
XmDirectionToStringDirection(1)	Convert a direction to a string direction
XmDragCancel(1)	Cancel a drag operation
XmDragStart(1)	Start a drag operation
XmDropDownGetArrow(1)	Return the “arrow” child of the XmDropDown
XmDropDownGetChild(1)	Get the specified child of a DropDown widget
XmDropDownGetLabel(1)	Return the “label” child of the XmDropDown
XmDropDownGetList(1)	Return the “list” child of the XmDropDown
XmDropDownGetText(1)	Return the “arrow” child of the XmDropDown
XmDropDownGetValue(1)	Retrieve the value from DropDown.
XmDropSideEndUpdate(1)	End an update of multiple drop site operations
XmDropSiteConfigureStackingOrder(1)	Change the stacking order of a drop site
XmDropSiteQueryStackingOrder(1)	Get the stacking order of a drop site
XmDropSiteRegister(1)	Register a widget as a drop site
XmDropSiteRetrieve(1)	Get the resource values of a drop site
XmDropSiteStartUpdate(1)	Start an update of multiple drop site operations
XmDropSiteUnregister(1)	Remove a drop site

Appendix A: Function Summaries

<code>XmDropSiteUpdate(1)</code>	Change the resource values for a drop site
<code>XmDropTransferAdd(1)</code>	Add drop transfer entries to a drop operation
<code>XmDropTransferStart(1)</code>	Start a drop operation
<code>XmFileSelectionBoxGetChild(1)</code>	Get a specified child of a FileSelection-Box widget
<code>XmFileSelectionDoSearch(1)</code>	Start a directory search
<code>XmFontListAdd(1)</code>	Create a new font list
<code>XmFontListAppendEntry(1)</code>	Append a font entry to a font list
<code>XmFontListCopy(1)</code>	Copy a font list
<code>XmFontListCreate(1)</code>	Create a new font list
<code>XmFontListCreate_r(1)</code>	Create a new font list in a thread-safe manner
<code>XmFontListEntryCreate(1)</code>	Create a new font list entry
<code>XmFontListEntryCreate_r(1)</code>	Create a new font list entry in a thread-safe manner
<code>XmFontListEntryFree(1)</code>	Free the memory used by a font list entry
<code>XmFontListEntryGetFont(1)</code>	Get the font information from a font list entry
<code>XmFontListEntryGetTag(1)</code>	Get the tag of a font list entry
<code>XmFontListEntryLoad(1)</code>	Load a font or create a font set, and then create a font list entry
<code>XmFontListFree(1)</code>	Free the memory used by a font list
<code>XmFontListFreeFontContext(1)</code>	Free a font context
<code>XmFontListGetNextFont(1)</code>	Retrieve information about the next font list element
<code>XmFontListInitFontContext(1)</code>	Create a font context
<code>XmFontListNextEntry(1)</code>	Retrieve the next font list entry in a font list
<code>XmFontListRemoveEntry(1)</code>	Remove a font list entry from a font list
<code>XmGetAtomName(1)</code>	Get the string representation of an Atom
<code>XmGetColorCalculation(1)</code>	Get the procedure that calculates default widget colors
<code>XmGetColors(1)</code>	Get the foreground, select, and shadow colors for a widget
<code>XmGetDestination(1)</code>	Get the current destination widget
<code>XmGetDragContext(1)</code>	Get information about a drag-and-drop operation
<code>XmGetFocusWidget(1)</code>	Get the widget that has the keyboard focus

Appendix A: Function Summaries

XmGetMenuCursor(1)	Get the current menu cursor
XmGetPixmap(1)	Create and return a pixmap
XmGetPixmapByDepth(1)	Create and return a pixmap of the specified depth
XmGetPostedFromWidget(1)	Get the widget that posted a menu
XmGetScaledPixmap(1)	Create and return a scaled pixmap
XmGetSecondaryResourceData(1)	Retrieve secondary widget resource data
XmGetTabGroup(1)	Get the tab group for a widget
XmGetTearOffControl(1)	Get the tear-off control for a Menu
XmGetVisibility(1)	Determine whether or not a widget is visible
XmGetXmDisplay(1)	Get the XmDisplay object for the Display
XmGetXmScreen(1)	Get the XmScreen object for a screen
XmHierarchyGetChildNodes(1)	Return the list of “node children” of the node widget
XmHierarchyOpenAllAncestors(1)	Open all ancestors of a given node
XmIconBoxIsCellEmpty(1)	An IconBox function that determines whether a cell in the IconBox is empty
XmImCloseXIM(1)	Close all input contexts associated with the input method
XmImFreeXIC(1)	Free an input context
XmImGetXIC(1)	Create an input context for a widget
XmImGetXIM(1)	Retrieve the input method for a widget
XmImMbLookupString(1)	Retrieve a compound string from an input method
XmImMbResetIC(1)	Reset an input context
XmImRegister(1)	Register an input context for a widget
XmImSetFocusValues(1)	Set the values and focus for an input context
XmImSetValues(1)	Set the values for an input context
XmImSetXIC(1)	Register a widget with an existing input context
XmImUnregister(1)	Unregister the input context for a widget
XmImUnsetFocus(1)	Unset focus for an input context
XmImVaSetFocusValues(1)	Set the values and focus for an input context
XmImVaSetValues(1)	Set the values for an input context
XmInsiallImage(1)	Install an image in the image cache
XmInternAtom(1)	Return an Atom for a given property name string

Appendix A: Function Summaries

XmIsMotifWMRunning(1)	Check whether the Motif Window Manager (mwm) is running
XmIsObject(1)	Determine whether a widget is a subclass of a given class
XmIsTraversable(1)	Determine whether or not a widget can receive the keyboard focus
XmListAddItem(1)	Add an item to a List
XmListAddItemUnselected(1)	Add an item to a List, without selecting the item
XmListAddItems(1)	Add items to a List
XmListAddItemsUnselected(1)	Add items to a List, without selecting the items
XmListDeleteAllItems(1)	Delete all of the items from a List
XmListDeleteItem(1)	Delete an item from a List
XmListDeleteItems(1)	Delete items from a List
XmListDeleteItemsPos(1)	Delete items starting at a specified position from a List
XmListDeletePos(1)	Delete an item at a specified position from a List
XmListDeletePositions(1)	Delete items at specified positions from a List
XmListDeselectAllItems(1)	Deselect all items in a List
XmListDeselectPos(1)	Deselect an item at a specified position in a List
XmListGetKbdItemPos(1)	Get the position of the item that has the location cursor in a List
XmListGetMatchPos(1)	Get all occurrences of an item in a List
XmListItemExists(1)	Determine if a specified item is in a List
XmListItemPos(1)	Determine the position of an item in a List
XmListPosSelected(1)	Check if the item at the specified position in the List is selected
XmListPosToBounds(1)	Return the bounding box of the item at the specified position in a List
XmListReplaceItems(1)	Replace specified items in a List
XmListReplaceItemsPos(1)	Replace items at specified positions in a List
XmListReplaceItemsPosUnselected(1)	Replace items at specified positions in a List, without selecting the replacement items
XmListReplaceItemsUnselected(1)	Replace items in the List, without selecting the replacement items

Appendix A: Function Summaries

<code>XmListReplacePositions(1)</code>	Replace items at the specified positions in a List
<code>XmListSelectItem(1)</code>	Select a specified item in a List
<code>XmListSelectPos(1)</code>	Select the item at the specified position in a List
<code>XmListSelectedPos(1)</code>	Get the positions of the selected items in a List
<code>XmListSetAddMode(1)</code>	Set add mode in a List
<code>XmListSetBottomItem(1)</code>	Set the last visible item in a List
<code>XmListSetBottomPos(1)</code>	Set the last visible position in a List
<code>XmListSetHorizPos(1)</code>	Set the horizontal position of a List
<code>XmListSetItem(1)</code>	Set the first visible item in a List
<code>XmListSetKbdItemPos(1)</code>	Set the position of the location cursor in a List
<code>XmListSetPos(1)</code>	Set the first visible position in a List
<code>XmListUpdateSelectedList(1)</code>	Update the set of selected items in a List
<code>XmListYToPos(1)</code>	Get the position of the item at the specified y-coordinate in a List
<code>XmMainWindowSep1(1)</code>	Get the widget ID of the first MainWindow Separator
<code>XmMainWindowSep2(1)</code>	Get the widget ID of the second MainWindow Separator
<code>XmMainWindowSep3(1)</code>	Get the widget ID of the third MainWindow Separator
<code>XmMainWindowSetAreas(1)</code>	Specify the children for a MainWindow
<code>XmMapSegmentEncoding(1)</code>	Get the compound text encoding format for a font list element tag
<code>XmMenuPosition(1)</code>	Position a popup menu
<code>XmMessageBoxGetChild(1)</code>	Retrieve a specified child of a MessageBox widget
<code>XmMultiListDeselectItems(1)</code>	Clear the selection state of items of a MultiList
<code>XmMultiListDeselectRow(1)</code>	Clear the selection state of a specified row of a MultiList widget
<code>XmMultiListGetSelectedRowArray(1)</code>	Returns NULL-terminated array of pointers to selected rows of MultiList
<code>XmMultiListGetSelectedRows(1)</code>	Find the rows that are currently selected
<code>XmMultiListMakeRowVisible(1)</code>	Make row of a MultiList widget visible
<code>XmMultiListSelectAllItems(1)</code>	Set the selection state on all rows of a MultiList
<code>XmMultiListSelectItems(1)</code>	Set the selection state of items of a MultiList

Appendix A: Function Summaries

<code>XmMultiListSelectRow(1)</code>	Set the selection state of a specified row of a MultiList widget
<code>XmMultiListToggleRow(1)</code>	Toggle the selection state of a specified row of a MultiList widget
<code>XmMultiListUnselectAllItems(1)</code>	Deselects all rows of the list
<code>XmMultiListUnselectItem(1)</code>	Deselects the specified item of the list
<code>XmNotebookGetPageInfo(1)</code>	Return information about a Notebook page
<code>XmObjectAtPoint(1)</code>	Determine the child nearest to a point coordinate
<code>XmOptionButtonGadget(1)</code>	Get the CascadeButtonGadget in an OptionMenu
<code>XmOptionLabelGadget(1)</code>	Get the LabelGadget in an OptionMenu
<code>XmPanedGetPanels(1)</code>	Retrieve the panes in the widget
<code>XmParseMappingCreate(1)</code>	Create a parse mapping
<code>XmParseMappingFree(1)</code>	Free the memory used by a parse mapping
<code>XmParseMappingGetValues(1)</code>	Fetch resources from a parse mapping object
<code>XmParseMappingSetValues(1)</code>	Set resources for a parse mapping object
<code>XmParseTableFree(1)</code>	Free the memory used by a parse table
<code>XmPrintPopupPDM(1)</code>	Notify the Print Display Manager
<code>XmPrintSetup(1)</code>	Create a PrintShell widget
<code>XmPrintSetup(1)</code>	Create a PrintShell widget
<code>XmPrintToFile(1)</code>	Save X print Server data to file
<code>XmProcessTraversal(1)</code>	Set the widget which has the keyboard focus
<code>XmRedisplayWidget(1)</code>	Force widget exposure for printing
<code>XmRegisterSegmentEncoding(1)</code>	Register a compound text encoding format for a font list element tag
<code>XmRemoveFromPostFromList(1)</code>	Make a menu inaccessible from a widget
<code>XmRemoveProtocolCallback(1)</code>	Remove client callback from a protocol
<code>XmRemoveProtocols(1)</code>	Remove protocols from the protocol manager
<code>XmRemoveTabGroup(1)</code>	Remove a widget from a tab group list
<code>XmRemoveWMProtocolCallback(1)</code>	Remove client callbacks from the XA_WM_PROTOCOLS protocol
<code>XmRemoveWMProtocols(1)</code>	Remove the XA_WM_PROTOCOLS protocol from the protocol manager
<code>XmRenderTableAddRenditions(1)</code>	Add renditions to a render table
<code>XmRenderTableCopy(1)</code>	Copy a render table

Appendix A: Function Summaries

<code>XmRenderTableCvtFromProp(1)</code>	Convert from a string representation into a render table
<code>XmRenderTableCvtToProp(1)</code>	Convert a render table into a string representation
<code>XmRenderTableFree(1)</code>	Free the memory used by a render table
<code>XmRenderTableGetRendition(1)</code>	Search a render table for a matching rendition
<code>XmRenderTableGetRenditions(1)</code>	Search a render table for a matching renditions
<code>XmRenderTableGetTags(1)</code>	Fetch the list of rendition tags from a render table
<code>XmRenderTableRemoveRenditions(1)</code>	Copy a render table, excluding specified renditions
<code>XmRenditionCreate(1)</code>	Create a rendition object
<code>XmRenditionFree(1)</code>	Free the memory used by a rendition object
<code>XmRenditionRetrieve(1)</code>	Fetch rendition object resources
<code>XmRenditionUpdate(1)</code>	Set rendition object resources
<code>XmRepTypeAddReverse(1)</code>	Install the reverse converter for a representation type
<code>XmRepTypeGetId(1)</code>	Get the ID number of a representation type
<code>XmRepTypeGetNameList(1)</code>	Get the list of value names for a representation type
<code>XmRepTypeGetRecord(1)</code>	Get the information about a representation type
<code>XmRepTypeGetRegistered(1)</code>	Get the registered representation types
<code>XmRepTypeInstallTearOffModelConverter(1)</code>	Install the resource converter for <code>XmNtearOffModel</code>
<code>XmRepTypeRegister(1)</code>	Register a representation type resource
<code>XmRepTypeValidValue(1)</code>	Determine the validity of a numerical value for a representation type
<code>XmResolveAllPartOffset(1)</code>	Ensure upward-compatible widgets and applications
<code>XmResolvePartOffsets(1)</code>	Ensure upward-compatible widgets and applications
<code>XmScaleGetValue(1)</code>	Get the slider value for a Scale widget
<code>XmScaleSetTicks(1)</code>	Set tick marks for a Scale widget
<code>XmScaleSetValue(1)</code>	Set the slider value for a Scale widget
<code>XmScrollBarGetValues(1)</code>	Get information about the current state of a ScrollBar

Appendix A: Function Summaries

<code>XmScrollBarSetValues(1)</code>	Set the current state of a ScrollBar widget
<code>XmScrollVisible(1)</code>	Make an obscured child of a Scrolled-Window visible
<code>XmScrolledWindowSetAreas(1)</code>	Set the children for a ScrolledWindow
<code>XmSelectionBoxGetChild(1)</code>	Get the specified child of a Selection-Box widget
<code>XmSetColorCalculation(1)</code>	Set the procedure that calculates default widget colors
<code>XmSetFontUnit(1)</code>	Set the font unit values
<code>XmSetFontUnits(1)</code>	Set the font unit values
<code>XmSetMenuCursor(1)</code>	Set the current menu cursor
<code>XmSetProtocolHooks(1)</code>	Set prehooks and posthooks for a protocol
<code>XmSetWMProtocolHooks(1)</code>	Set prehooks and posthooks for the XA_WM_PROTOCOLS protocol
<code>XmSimpleSpinBoxAddItem(1)</code>	Add an item to a SimpleSpinBox
<code>XmSimpleSpinBoxDeletePos(1)</code>	Delete an item at a specified position from a SimpleSpinBox
<code>XmSimpleSpinBoxSetItem(1)</code>	Set an item in a SimpleSpinBox
<code>XmSpinBoxValidatePosition(1)</code>	Validate the current value of a SpinBox
<code>XmStringBaseline(1)</code>	Get the baseline spacing for a compound string
<code>XmStringByteCompare(1)</code>	Compare two compound strings byte-by-byte
<code>XmStringByteStreamLength(1)</code>	Calculate the length of a byte stream
<code>XmStringCompare(1)</code>	Compare two compound strings
<code>XmStringComponentCreate(1)</code>	Create a compound string component
<code>XmStringConcat(1)</code>	Concatenate two compound strings
<code>XmStringConcatAndFree(1)</code>	Create a new compound string formed by concatenating two compound strings, freeing the two strings afterwards
<code>XmStringCopy(1)</code>	Copy a compound string
<code>XmStringCreate(1)</code>	Create a compound string
<code>XmStringCreateLocalized(1)</code>	Create a compound string in the current locale
<code>XmStringCreateLtoR(1)</code>	Create a compound string
<code>XmStringCreateSimple(1)</code>	Create a compound string in the current language environment
<code>XmStringDirectionCreate(1)</code>	Create a compound string containing a direction component
<code>XmStringDirectionToDirection(1)</code>	Convert a string direction to a direction

Appendix A: Function Summaries

XmStringDirectionToDirection(1)	Convert a string direction to a direction
XmStringDraw(1)	Draw a compound string
XmStringDrawImage(1)	Draw a compound string
XmStringDrawUnderline(1)	Draw a compound string with an underlined sub-string
XmStringEmpty(1)	Determine whether there are any text components in a compound string
XmStringExtent(1)	Get the smallest rectangle enclosing a compound string
XmStringFree(1)	Free the memory used by a compound string
XmStringFreeContext(1)	Free a compound string context
XmStringGenerate(1)	Generate a compound string
XmStringGetLtoR(1)	Get a text segment from a compound string
XmStringGetNextComponent(1)	Retrieves information about the next compound string component
XmStringGetNextSegment(1)	Retrieves information about the next compound string component
XmStringGetNextTriple(1)	Retrieve information about the next compound string segment
XmStringHasSubstring(1)	Determine whether a compound string contains a sub-string
XmStringHeight(1)	Get the line height of a compound string
XmStringInitContext(1)	Create a compound string context
XmStringIsVoid(1)	Determine whether there are valid segments in a compound string
XmStringLength(1)	Get the length of a compound string
XmStringLineCount(1)	Get the number of lines in a compound string
XmStringNConcat(1)	Concatenate a specified portion of a compound string to another compound string
XmStringNCopy(1)	Copy a specified portion of a compound string
XmStringParseText(1)	Convert a string to a compound string
XmStringPeekNextComponent(1)	Returns the type of the next compound string component
XmStringPutRendition(1)	Add rendition components to a compound string
XmStringSegmentCreate(1)	Create a compound string segment

Appendix A: Function Summaries

<code>XmStringSeparatorCreate(1)</code>	Create a compound string containing a separator component
<code>XmStringTableParseStringArray(1)</code>	Convert an array of strings into a compound string table
<code>XmStringTableProposeTabList(1)</code>	Create a Tab list for a compound string
<code>XmStringTableToXmString(1)</code>	Convert a compound string table to a compound string
<code>XmStringTableUnparse(1)</code>	Convert a compound string table to a string
<code>XmStringToXmStringTable(1)</code>	Convert a compound string to a compound string table
<code>XmStringUnparse(1)</code>	Convert a compound string to a string
<code>XmStringWidth(1)</code>	Return the width of a compound string
<code>XmTabCreate(1)</code>	Create a tab (<code>XmTab</code>) object
<code>XmTabFree(1)</code>	Free the memory used by an <code>XmTab</code> object
<code>XmTabGetValues(1)</code>	Fetch the value of an <code>XmTab</code> object
<code>XmTabListCopy(1)</code>	Copy a tab list object
<code>XmTabListFree(1)</code>	Free the memory used by a tab list
<code>XmTabListGetTab(1)</code>	Retrieve a tab from a tab list
<code>XmTabListInsertTabs(1)</code>	Insert tabs into a tab list
<code>XmTabListRemoveTabs(1)</code>	Copy a tab list, excluding specified tabs
<code>XmTabListReplacePositions(1)</code>	Copy a tab list, replacing tabs at specified positions
<code>XmTabListTabCount(1)</code>	Count the number of tabs in a tab list
<code>XmTabSetValue(1)</code>	Set the value of a tab object
<code>XmTabStackGetSelectedTab(1)</code>	Return the widget ID of the currently selected tab
<code>XmTabStackIndexToWidget(1)</code>	Return the widget ID of the tab in the <code>TabStack</code>
<code>XmTabStackSelectTab(1)</code>	Set the currently displayed child of the <code>TabStack</code>
<code>XmTabStackXYToWidget(1)</code>	Convert a pixel coordinate in the <code>TabStack</code> 's window to the widget of the tab
<code>XmTargetsAreCompatible(1)</code>	Determine whether or not the target types of a drag source and a drop site match
<code>XmTextClearSelection(1)</code>	Clear the primary selection
<code>XmTextCopy(1)</code>	Copy the primary selection to the clipboard
<code>XmTextCopyLink(1)</code>	Copy links from the primary selection to the clipboard

Appendix A: Function Summaries

<code>XmTextCut(1)</code>	Copy the primary selection to the clipboard, and remove the selected text
<code>XmTextDisableRedisplay(1)</code>	Prevent visual update
<code>XmTextEnableRedisplay(1)</code>	Allow visual update
<code>XmTextFieldClearSelection(1)</code>	Clear the primary selection
<code>XmTextFieldCopy(1)</code>	Copy the primary selection to the clipboard
<code>XmTextFieldCopyLink(1)</code>	Copy links from the primary selection to the clipboard
<code>XmTextFieldCut(1)</code>	Copy the primary selection to the clipboard, and remove the selected text
<code>XmTextFieldGetBaseline(1)</code>	Get the position of the baseline
<code>XmTextFieldGetCursorPosition(1)</code>	Get the position of the location cursor
<code>XmTextFieldGetEditable(1)</code>	Get the edit permission state
<code>XmTextFieldGetInsertionPosition(1)</code>	Get the position of the insertion cursor
<code>XmTextFieldGetLastPosition(1)</code>	Get the position of the last character
<code>XmTextFieldGetMaxLength(1)</code>	Get the maximum possible length of a text string
<code>XmTextFieldGetSelection(1)</code>	Get the value of the primary selection
<code>XmTextFieldGetSelectionPosition(1)</code>	Get the position of the primary selection
<code>XmTextFieldGetSelectionWcs(1)</code>	Get the wide-character value of the primary selection
<code>XmTextFieldGetString(1)</code>	Get the text string
<code>XmTextFieldGetStringWcs(1)</code>	Get the wide-character text string
<code>XmTextFieldGetSubstring(1)</code>	Get a copy of part of the text string
<code>XmTextFieldGetSubstringWcs(1)</code>	Get a copy of part of the wide-character text string
<code>XmTextFieldGetTopCharacter(1)</code>	Get the position of the first visible character
<code>XmTextFieldInsert(1)</code>	Insert a string into the text string
<code>XmTextFieldInsertWcs(1)</code>	Insert a wide-character string into the text string
<code>XmTextFieldPaste(1)</code>	Insert the clipboard selection
<code>XmTextFieldPasteLink(1)</code>	Insert links from the clipboard selection
<code>XmTextFieldPosToXY(1)</code>	Get the x, y coordinates of a character position
<code>XmTextFieldReplace(1)</code>	Replace part of the text string
<code>XmTextFieldReplaceWcs(1)</code>	Replace part of the wide-character text string
<code>XmTextFieldRermove(1)</code>	Delete the primary selection
<code>XmTextFieldSetAddMode(1)</code>	Set the add mode state
<code>XmTextFieldSetCursorPosition(1)</code>	Set the position of the location cursor

Appendix A: Function Summaries

<code>XmTextFieldSetEditable(1)</code>	Set the edit permission state
<code>XmTextFieldSetHighlight(1)</code>	Highlight part of the text string
<code>XmTextFieldSetInsertionPosition(1)</code>	Set the position of the insertion cursor
<code>XmTextFieldSetMaxLength(1)</code>	Set the maximum possible length of a text string
<code>XmTextFieldSetSelection(1)</code>	Set the value of the primary selection
<code>XmTextFieldSetSource(1)</code>	Set the text source
<code>XmTextFieldSetString(1)</code>	Set the text string
<code>XmTextFieldSetStringWcs(1)</code>	Set the wide-character text string
<code>XmTextFieldShowPosition(1)</code>	Scroll the text so that a specified position is visible
<code>XmTextFieldXYToPos(1)</code>	Get the character position for a given x, y coordinate
<code>XmTextFindString(1)</code>	Find the beginning position of a text string
<code>XmTextFindStringWcs(1)</code>	Find the beginning position of a wide-character text string
<code>XmTextGetBaseline(1)</code>	Get the position of the baseline
<code>XmTextGetCenterline(1)</code>	Get the height of vertical text
<code>XmTextGetCursorPosition(1)</code>	Get the position of the location cursor
<code>XmTextGetEditable(1)</code>	Get the edit permission state
<code>XmTextGetInsertionPosition(1)</code>	Get the position of the insertion cursor
<code>XmTextGetLastPosition(1)</code>	Get the position of the last character
<code>XmTextGetMaxLength(1)</code>	Get the maximum possible length of a text string
<code>XmTextGetSelection(1)</code>	Get the value of the primary selection
<code>XmTextGetSelectionPosition(1)</code>	Get the position of the primary selection
<code>XmTextGetSelectionWcs(1)</code>	Get the wide-character value of the primary selection
<code>XmTextGetSource(1)</code>	Get the text source
<code>XmTextGetString(1)</code>	Get the text string
<code>XmTextGetStringWcs(1)</code>	Get the wide-character text string
<code>XmTextGetSubstring(1)</code>	Get a copy of part of the text string
<code>XmTextGetSubstringWcs(1)</code>	Get a copy of part of the wide-character text string
<code>XmTextGetTopCharacter(1)</code>	Get the position of the first visible character
<code>XmTextInsert(1)</code>	Insert a string into the text string
<code>XmTextInsertWcs(1)</code>	Insert a wide-character string into the text string
<code>XmTextPaste(1)</code>	Insert the clipboard selection
<code>XmTextPasteLink(1)</code>	Insert links from the clipboard selection

Appendix A: Function Summaries

XmTextPosToXY(1)	Get the x, y coordinates of a character position
XmTextReplace(1)	Replace part of the text string
XmTextReplaceWcs(1)	Replace part of the wide-character text string
XmTextRermove(1)	Delete the primary selection
XmTextScroll(1)	Scroll the text
XmTextSetAddMode(1)	Set the add mode state
XmTextSetCursorPosition(1)	Set the position of the location cursor
XmTextSetEditable(1)	Set the edit permission state
XmTextSetHighlight(1)	Highlight part of the text string
XmTextSetInsertionPosition(1)	Set the position of the insertion cursor
XmTextSetMaxLength(1)	Set the maximum possible length of a text string
XmTextSetSelection(1)	Set the value of the primary selection
XmTextSetSource(1)	Set the text source
XmTextSetString(1)	Set the text string
XmTextSetStringWcs(1)	Set the wide-character text string
XmTextSetTopCharacter(1)	Set the position of the first visible character
XmTextShowPosition(1)	Scroll the text so that a specified position is visible
XmTextXYToPos(1)	Get the character position for a given x, y coordinate
XmToggleButtonGadgetGetState(1)	Get the state of a ToggleButtonGadget
XmToggleButtonGadgetSetState(1)	Set the state of a ToggleButtonGadget
XmToggleButtonGadgetSetValue(1)	Set the value of a ToggleButtonGadget
XmToggleButtonGetState(1)	Get the state of a ToggleButton
XmToggleButtonSetState(1)	Set the state of a ToggleButton
XmToggleButtonSetValue(1)	Set the value of a ToggleButton
XmTrackingEvent(1)	Allow for modal selection of a component
XmTrackingLocate(1)	Allow for modal selection of a component
XmTransferDone(1)	Complete a data transfer operation
XmTransferSendRequest(1)	Send a multiple transfer request
XmTransferSetParameters(1)	Set parameters for the next transfer operation
XmTransferStartRequest(1)	Initiate a multiple transfer batch operation
XmTransferValue(1)	Transfer data to a destination

Appendix A: Function Summaries

XmTranslateKey(1)	Convert a KeyCode to a KeySym using the default translator
XmUninstallImage(1)	Remove an image from the image cache
XmUpdateDisplay(1)	Update the display
XmVaCreateSimpleCheckBox(1)	Create a CheckBox compound object
XmVaCreateSimpleMenuBar(1)	Create a MenuBar compound object
XmVaCreateSimpleOptionMenu(1)	Create an OptionMenu compound object
XmVaCreateSimplePopupMenu(1)	Create a PopupMenu compound object as the child of a MenuShell
XmVaCreateSimplePulldownMenu(1)	Create a PulldownMenu compound object as the child of a MenuShell
XmVaCreateSimpleRadioBox(1)	Create a RadioBox compound object
XmWidgetGetBaselines(1)	Get the positions of the baselines in a widget
XmWidgetGetDisplayRect(1)	Get the display rectangle for a widget

Appendix A: Function Summaries