DECwindows Extensions to Motif

Order Number: AA–Q8YSA–TE

August 1994

This manual describes the programming extensions Digital provides to supplement the X Window System, Version 11, Release 5 and OSF/Motif Toolkit components included in the DECwindows Motif software.

Revision/Update Information:	This manual is a revision.
Operating System:	DEC OSF/1 Version 3.0 or higher Microsoft Windows NT Version 3.1 or higher ULTRIX Version 4.4 or higher
Software Version:	DEC OSF/1 Version 3.0 or higher eXcursion (SDK) for Windows NT Version 1.2 or higher ULTRIX Worksystem Software Version 4.4 or higher

Digital Equipment Corporation
Maynard, Massachusetts

August 1994

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

© Digital Equipment Corporation 1994. All Rights Reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: Alpha AXP, AXP, Bookreader, CDA, DDIS, DEC, DECdecision, DEC FUSE, DECnet, DECstation, DECsystem, DECterm, DECUS, DECwindows, DECwrite, Digital, DTIF, eXcursion, LN03, MASSBUS, MicroVAX, OpenVMS, PrintServer, Q–bus, ReGIS, ULTRIX, ULTRIX Mail Connection, ULTRIX Worksystem Software, UNIBUS, VAX, VAXcluster, VAX DOCUMENT, VAXserver, VAXstation, VMS, VT, and the DIGITAL logo.

Adobe, Display PostScript, and PostScript are registered trademarks of Adobe Systems Incorporated.

BITSTREAM is a registered trademark of Bitstream, Inc.

Helvetica and Times are trademarks of Linotype AG or its subsidiaries, or both.

ITC Avant Garde Gothic is a registered trademark of International Typeface Corporation.

Open Software Foundation, OSF, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

NT and Windows NT are registered trademarks and Windows is a trademark of Microsoft Corporation.

ZK5635

This document was prepared using VAX DOCUMENT Version 2.1.

Contents

About This Manual xi

1 DECwindows Xlib Extensions for OpenVMS

XSelectAsyncEvent 1–2
XSelectAsyncInput 1–4

2 DECwindows Toolkit Extensions

2.1 DECwindows Resource and Routine Extensions 2–1
2.1.1 DXmNautoUnrealize Resource 2–1
2.1.2 DXmNfitToScreenPolicy Resource 2–1
2.1.3 DXmNlayoutDirection Resource 2–2
2.1.3.1 DXmNlayoutDirection in DECwindows Extensions to Motif 2–2
2.1.3.2 DXmNlayoutDirection in DECwindows Motif 2–4
2.1.4 XmStringDirectionCreate Routine 2–7
2.2 DECwindows Toolkit Routines 2–8
DXmActivateWidget 2–10
DXmChildren 2–11
DXmColorMixGetNewColor 2–12
DXmColorMixSetNewColor 2–14
DXmCreateColorMix 2–16
DXmCreateColorMixDialog 2–18
DXmCreateCSText 2–38
DXmCreateCursor 2–53
DXmCreateHelpDialog 2–55
DXmCreatePrintBox 2–71
DXmCreatePrintDialog 2–73
DXmCreateScrolledCSText 2–91
DXmCreateSvn 2–93

DXmCSContainsStringCharSet 2–121
DXmCSTextClearSelection 2–122
DXmCSTextCopy 2–123
DXmCSTextCut 2–124
DXmCSTextDisableRedisplay 2–125
DXmCSTextEnableRedisplay 2–126
DXmCSTextGetEditable 2–127
DXmCSTextGetInsertionPosition 2–128
DXmCSTextGetLastPosition 2–129
DXmCSTextGetMaxLength 2–130
DXmCSTextGetSelection 2–131
DXmCSTextGetSelectionInfo 2–132
DXmCSTextGetString 2–133
DXmCSTextGetTopPosition 2–134
DXmCSTextHasSelection 2–135
DXmCSTextHorizontalScroll 2–136
DXmCSTextInsert 2–137
DXmCSTextNumLines 2–138
DXmCSTextPaste 2–139
DXmCSTextPosToXY 2–140
DXmCSTextRemove 2–141
DXmCSTextReplace 2–142
DXmCSTextSetAddMode 2–144
DXmCSTextSetEditable 2–145
DXmCSTextSetHighlight 2–146
DXmCSTextSetInsertionPosition 2–147
DXmCSTextSetMaxLength 2–148
DXmCSTextSetSelection 2–149
DXmCSTextSetString 2–150
DXmCSTextSetTopPosition 2–151
DXmCSTextShowPosition 2–152
DXmCSTextVerticalScroll 2–153
DXmCSTextXYToPos 2–154
DXmCvtCStoDDIF 2–155
DXmCvtCStoFC 2–156
DXmCvtCStoOS 2–157
DXmCvtDDIFtoCS 2–158

DXmCvtFCtoCS 2–159
DXmCvtOStoCS 2–160
DXmDescToNull 2–161
DXmDisplayCSMessage 2–162
DXmDisplayVMSMessage 2–165
DXmFindFontFallback 2–168
DXmFormSpaceButtonsEqually 2–169
DXmGetLocaleString 2–170
DXmHelpOnContext 2–172
DXmHelpSystemClose 2–173
DXmHelpSystemDisplay 2–175
DXmHelpSystemOpen 2–177
DXmInitialize 2–179
DXmLoadQueryFont 2–180
DXmNumChildren 2–181
DXmPrintWgtAugmentList 2–182
DXmPrintWgtPrintJob 2–185
DXmSvnAddEntries 2–186
DXmSvnAutoScrollCheck 2–188
DXmSvnAutoScrollDisplay 2–190
DXmSvnClearHighlight 2–191
DXmSvnClearHighlighting 2–192
DXmSvnClearSelection 2–193
DXmSvnClearSelections 2–194
DXmSvnDeleteEntries 2–195
DXmSvnDisableDisplay 2–196
DXmSvnEnableDisplay 2–197
DXmSvnFlushEntry 2–198
DXmSvnGetComponentNumber 2–199
DXmSvnGetComponentTag 2–200
DXmSvnGetComponentText 2–201
DXmSvnGetComponentWidth 2–202
DXmSvnGetDisplayed 2–203
DXmSvnGetEntryLevel 2–205
DXmSvnGetEntryNumber 2–206
DXmSvnGetEntryPosition 2–207
DXmSvnGetEntrySensitivity 2–209

DXmSvnGetEntryTag 2–210
DXmSvnGetHighlighted 2–211
DXmSvnGetNumDisplayed 2–213
DXmSvnGetNumHighlighted 2–214
DXmSvnGetNumSelections 2–215
DXmSvnGetPrimaryWorkWidget 2–216
DXmSvnGetSecondaryWorkWidget 2–217
DXmSvnGetSelections 2–218
DXmSvnGetTreePosition 2–220
DXmSvnHideHighlighting 2–221
DXmSvnHideSelections 2–222
DXmSvnHighlightAll 2–223
DXmSvnHighlightEntry 2–224
DXmSvnInsertComponent 2–225
DXmSvnInvalidateEntry 2–226
DXmSvnMapPosition 2–227
DXmSvnPositionDisplay 2–228
DXmSvnRemoveComponent 2–230
DXmSvnSelectAll 2–231
DXmSvnSelectComponent 2–232
DXmSvnSelectEntry 2–233
DXmSvnSetApplDragging 2–234
DXmSvnSetComponentHidden 2–235
DXmSvnSetComponentPixmap 2–237
DXmSvnSetComponentTag 2–239
DXmSvnSetComponentText 2–240
DXmSvnSetComponentWidget 2–242
DXmSvnSetComponentWidth 2–244
DXmSvnSetEntry 2–245
DXmSvnSetEntryIndexWindow 2–247
DXmSvnSetEntryNumComponents 2–248
DXmSvnSetEntryPosition 2–249
DXmSvnSetEntrySensitivity 2–251
DXmSvnSetEntryTag 2–252
DXmSvnSetTreePosition 2–253
DXmSvnShowHighlighting 2–254
DXmSvnShowSelections 2–255

DXmSvnValidateAll 2–256

3 DECwindows Extensions to UIL

3.1 Nested Compound Strings 3–1
3.2 Command Qualifiers to the UIL Compiler 3–3
3.2.1 UIL Compiler Command Qualifiers on OpenVMS Systems 3–3
3.2.2 UIL Compiler Command Flags on UNIX and Windows NT Systems 3–5
3.3 WML Binary Databases 3–7
3.3.1 WML Files on OpenVMS Systems 3–7
3.3.2 Supporting Additional Widgets on OpenVMS Systems 3–8
3.3.3 WML Files on UNIX and Windows NT Systems 3–9
3.3.4 Supporting Additional Widgets on UNIX and Windows NT Systems 3–10

A UIL Built-In Tables

A.1 DXmColorMix A–2
A.2 DXmColorMixDialog A–6
A.3 DXmCSText A–10
A.4 DXmHelpDialog A–13
A.5 DXmPrintBox A–17
A.6 DXmPrintDialog A–21
A.7 DXmScrolledCSText A–25
A.8 DXmSvn A–27
A.9 XmArrowButton A–30
A.10 XmArrowButtonGadget A–32
A.11 XmBulletinBoard A–33
A.12 XmBulletinBoardDialog A–35
A.13 XmCascadeButton A–37
A.14 XmCascadeButtonGadget A–39
A.15 XmCommand A–40
A.16 XmDrawingArea A–43
A.17 XmDrawnButton A–45
A.18 XmErrorDialog A–47
A.19 XmFileSelectionBox A–49
A.20 XmFileSelectionDialog A–53
A.21 XmForm A–56
A.22 XmFormDialog A–58
A.23 XmFrame A–61
A.24 XmInformationDialog A–63

A.25 XmLabel A–65
A.26 XmLabelGadget A–67
A.27 XmList A–69
A.28 XmMainWindow A–71
A.29 XmMenuBar A–73
A.30 XmMessageBox A–75
A.31 XmMessageDialog A–78
A.32 XmOptionMenu A–80
A.33 XmPanedWindow A–82
A.34 XmPopupMenu A–84
A.35 XmPromptDialog A–87
A.36 XmPulldownMenu A–89
A.37 XmPushButton A–92
A.38 XmPushButtonGadget A–95
A.39 XmQuestionDialog A–96
A.40 XmRadioBox A–99
A.41 XmRowColumn A–101
A.42 XmScale A–104
A.43 XmScrollBar A–106
A.44 XmScrolledList A–108
A.45 XmScrolledText A–110
A.46 XmScrolledWindow A–113
A.47 XmSelectionBox A–114
A.48 XmSelectionDialog A–117
A.49 XmSeparator A–120
A.50 XmSeparatorGadget A–121
A.51 XmText A–122
A.52 XmTextField A–124
A.53 XmToggleButton A–127
A.54 XmToggleButtonGadget A–129
A.55 XmWarningDialog A–131
A.56 XmWorkArea A–133
A.57 XmWorkingDialog A–136
A.58 user_defined A–138
A.59 All Widgets and Gadgets A–138
A.60 All Widgets A–141

B OpenVMS DECwTermPort Routine

DECwTermPort B–2

C SVN and Compound String Text Widget Translations

Index

Tables

1–1 Event Mask Elements 1–5
2–1 DXmNlayoutDirection Values 2–3
2–2 DXmNlayoutDirection Values for the DXmCreateScrolledCSText and DXmCreateCSText Functions 2–4
2–3 DXmNlayoutDirection Effect on Widget Components 2–5
3–1 OpenVMS Compiler Command Qualifiers 3–3
3–2 Motif UIL Compiler Command Flags on UNIX and Windows NT Systems 3–5
C–1 DXmSvn Widget Button Syntax Translations C–1
C–2 DXmSvn Widget Keyboard Syntax Translations C–2
C–3 Action Routines for DXmSvn Widget C–3
C–4 DXmCSText Widget Button Syntax Translations C–7
C–5 DXmCSText Widget Action Routines C–9
C–6 Additional DXmCSText Behavior C–20

About This Manual

Audience

This document, which supplements the *OSF/Motif Programmer's Reference* and the *X Window System* manuals, contains reference information for programmers who want to write applications that use the DECwindows Motif interfaces.

Organization

This document is organized as follows:

- Chapter 1 describes Xlib routines Digital has added to OpenVMS systems to supplement those provided by X Window System, Version 11, Release 5. These routines are not available for Digital UNIX systems or Windows NT systems.
- Chapter 2 describes Toolkit widget creation and manipulation routines, message routines, font routines, and convenience routines Digital has added to supplement those provided by the OSF/Motif Toolkit.
- Chapter 3 describes how to use extensions that Digital has added to the Motif User Interface Language (UIL), including how to use nested compound strings, new command line qualifiers in the UIL compiler, and the Widget Meta Language (WML) binary databases.
- Appendix A contains UIL built-in tables you can use to make sure that your UIL specification is consistent with the DECwindows Motif Toolkit.
- Appendix B describes the OpenVMS DECwTermPort routine that Digital provides to allow users to create a DECterm window on a local or remote node.
- Appendix C contains information on Structured Visual Navigation (SVN) and compound string text widget translations.

Related Documents

The following documents contain related information:

- *Overview of DECwindows Motif for OpenVMS Documentation* provides an overview of all the documents associated with the DECwindows Motif Version 1.2 release.
- *OSF/Motif Programmer's Reference* provides reference information on the Motif Toolkit.
- *OSF/Motif Programmer's Guide* describes how to program with the Motif Window Manager, Motif Toolkit, and the Motif User Interface Language (UIL).
- *DECwindows Motif Guide to Application Programming* describes how to program with the Digital extensions to the Motif Toolkit. It supplements the *OSF/Motif Programmer's Guide*.
- *OSF/Motif Style Guide* describes style guidelines for applications based on the Motif Toolkit.
- *DECwindows Companion to the OSF/Motif Style Guide* provides style guidelines for Digital extensions to Motif and describes topics not addressed in the *OSF/Motif Style Guide*.
- *X Window System* provides reference information on Xlib.
- *X Window System Toolkit* provides reference information on the Intrinsics.
- *VMS DECwindows Guide to Xlib (Release 4) Programming: MIT C Binding* describes how to program with Xlib using C bindings.
- *X and Motif Quick Reference Guide* provides quick reference information on Xlib, Intrinsics, and the Motif Toolkit.
- *DECwindows Motif for OpenVMS Applications Guide* describes how to use the applications bundled with DECwindows Motif.
- *Porting XUI Applications to Motif* describes how to port an existing DECwindows application to Motif.

The printed version of the DEC OSF/1 documentation set is color coded to help specific audiences quickly find the books that meet their needs. (You can order the printed documentation from Digital.) This color coding is reinforced with the use of an icon on the spines of books. The following list describes this convention:

Audience	Icon	Color Code
General Users	G	Teal
System Administrators	S	Red
Network Administrators	N	Yellow
Programmers	P	Blue
Reference Page Users	R	Black

Some books in the documentation set help meet the needs of several audiences. For example, the information in some system books is also used by programmers. Keep this in mind when searching for information on specific topics.

The *DEC OSF/1 Documentation Overview* provides information on all of the books in the DEC OSF/1 documentation set.

Reader's Comments

Digital welcomes your comments on this or any other DEC OSF/1 manual. A Reader's Comment form is located in the back of each printed DEC OSF/1 manual, and online in the following location:

```
/usr/doc/readers_comment.txt
```

You can send your comments in the following ways:

- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: 603-881-0120 Attn: UEG Publications, ZK03-3/Y32
- Mail:

  Digital Equipment Corporation
  UEG Publications Manager
  ZK03-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-9987

  The Readers Comment form located in the back of each printed manual is postage paid if you mail it in the United States.

If you have suggestions for improving particular sections or find any errors, please indicate the title, order number, and section numbers. Digital also welcomes general comments.

Conventions

The following conventions are used in this manual:

mouse	The term *mouse* is used to refer to any pointing device, such as a mouse, a puck, or a stylus.
MB1 (Select) MB2 (Drag) MB3 (Menu)	MB1 indicates the left mouse button, MB2 indicates the middle mouse button, and MB3 indicates the right mouse button. (The buttons can be redefined by the user.)
Ctrl+x	A sequence such as Ctrl+x (or Ctrl/x) indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, a field, a resource, or a reason.
numbers	Unless otherwise noted, all numbers in the text are assumed to be decimal. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Documentation Format of Routines

The routines described in this document use the following formatting conventions:

- **Routine Name**

  The exact (case-specific) C name of the routine.

- **Overview**

  A brief statement describing what the routine does.

- **Widget Class Hierarchy**

  For widget creation routines, a graphic representation of the order in which widgets inherit resources from their superclass widgets. (Exceptions are listed in the Resource Exceptions section.)

- **Format**

  The C routine declaration. Note the following syntax conventions:

Element	Syntax Rule
Routine entry point name	The name of the routine is shown in initial capital letters. The prefix X is used for Xlib routines and the prefix Xm or DXm is used for Toolkit routines. The routine entry point name is required.
Parentheses	Open and close parentheses (()) surround the argument list in a routine call. They are required.
Argument names	Argument names, including return arguments, are always shown in lowercase characters. All arguments are required, unless they are enclosed by brackets ([]). Arguments must be listed in the same order in your program as they appear in the format.
Spaces	One or more spaces are used between the entry point name and the first argument, and between each argument. They are not required.
Brackets ([])	Brackets surround optional arguments. The brackets are not required.
Commas	Commas are required between arguments. Commas that appear inside brackets are optional and appear only when the optional argument appears.

- **Resource Information**

  For widget creation routines, a list of the resources and their corresponding types.

- **Return Value**

  A description of the possible values that can be returned by the routine.

- **Arguments**

  Descriptions of the values that are valid for each argument and what those values indicate.

- **Resources**

  For widget creation routines, detailed descriptions of the widget-specific resources including the default value and access for each. Access is indicated by the following letter codes:

Letter Code	Type of Access
C	The resource can be set when the widget is first created.
S	The resource can be set at any time.
G	The value for the resource can be retrieved at any time.

- **Resource Exceptions**

  Exceptions regarding the use or characteristics of resources that widgets inherit from their superclass widgets. Exceptions include resources that are either not supported or are supported differently by a particular widget.

- **Callback Structure**

  For widget creation routines, the C format of the callback data structure.

- **Callback Field Descriptions**

  For widget creation routines, a description of the fields in the callback data structure.

- **Callback Reasons**

  For widget creation routines, a description of the callback reasons for the widget.

- **Description**

  A description of how to use the routine, including pointers to additional information.

- **Geometry management**

  For widget creation routines, a description of how the widget manages the sizing of its children.

- **Resizing**

  For widget creation routines, a description of how the widget resizes itself.

1

DECwindows Xlib Extensions for OpenVMS

This chapter describes the Xlib routines Digital provides on OpenVMS systems to supplement those included in X Window System, Version 11, Release 5. (Note that these routines are not available on Digital UNIX systems or Windows NT systems.)

XSelectAsyncEvent

Specifies an action routine and argument to be called when an event occurs.

Format

```
XSelectAsyncEvent(display, window_id, event_type,
                  ast_routine, ast_userarg)
      Display        *display;
      Window         window_id;
      unsigned long  event_type;
      int            (*ast_routine)();
      unsigned long  ast_userarg;
```

Arguments

display
The display information originally returned by XOpenDisplay.

window_id
The identifier of the window for which you want to select asynchronous events.

event_type
The type of event for which you want to select asynchronous events.

ast_routine
The particular asynchronous system trap (AST) action routine to use when notifying the client that the specified event has occurred.

ast_userarg
The user-specified argument to use when notifying the client that the specified event has occurred.

Description

Before calling XSelectInput to specify interest in a particular set of events for a window, clients can call XSelectAsyncEvent to specify an action routine and argument to be called when the specified event occurs.

Xlib uses the client's **ast_routine** and **ast_userarg** information to deliver an AST whenever it places the specified event on the event queue. The AST acts only as an event notification mechanism; the application uses the standard Xlib event routines to actually retrieve and process the event from the event queue.

When a client calls the **ast_routine**, the routine passes the **ast_userarg** argument to the client by value.

Clients can call XSelectAsyncEvent multiple times to specify different routine and argument pairs for different events for a window. The last call always takes precedence. If a client calls XSelectAsyncEvent with the **ast_routine** argument equal to zero, asynchronous notification is disabled, but the current selection for the specified event is unaffected.

Notification ASTs are queued in the same order as events are placed in the event queue by Xlib. Therefore, clients can assume that they receive notification ASTs in the same order that they find events in the queue.

XSelectAsyncEvent is similar to XSelectAsyncInput except that XSelectAsyncEvent specifies one event type and XSelectAsyncInput specifies an event mask.

XSelectAsyncInput

Specifies an action routine and arguments to be called when some subset of events occurs.

Format

```
XSelectAsyncInput(display, window_id, event_mask,
                  ast_routine, ast_userarg)
  Display        *display;
  Window         window_id;
  unsigned long  event_mask;
  int            (*ast_routine)();
  unsigned long  ast_userarg;
```

Arguments

display
The display information originally returned by XOpenDisplay.

window_id
The identifier of the window for which you want to select asynchronous input.

event_mask
A bitmask that specifies the event types for which you want to remove an event. The mask is an inclusive OR of one or more of the event mask elements described in Table 1–1.

ast_routine
The particular asynchronous system trap (AST) action routine to use when notifying the client that one of the specified events has occurred.

ast_userarg
The user-specified parameter to use when notifying the client that one of the specified events has occurred.

Table 1–1 Event Mask Elements

Bit	C Value	Description
1	KeyPressMask	Keyboard down events wanted (by the application)
2	KeyReleaseMask	Keyboard up events wanted (by the application)
3	ButtonPressMask	Pointer button down events wanted (by the application)
4	ButtonReleaseMask	Pointer button up events wanted (by the application)
5	EnterWindowMask	Pointerwindow entry events wanted (by the application)
6	LeaveWindowMask	Pointerwindow leave events wanted (by the application)
7	PointerMotionMask	Pointer motion events wanted (by the application)
8	PointerMotionHintMask	Pointer motion hints wanted (by the application)
9	Button1MotionMask	Pointer motion while button 1 down
10	Button2MotionMask	Pointer motion while button 2 down
11	Button3MotionMask	Pointer motion while button 3 down
12	Button4MotionMask	Pointer motion while button 4 down
13	Button5MotionMask	Pointer motion while button 5 down
14	ButtonMotionMask	Pointer motion while any button down
15	KeymapStateMask	Keyboard state wanted (by the application) at window entry and focus in
16	ExposureMask	Any exposure wanted (by the application)
17	VisibilityChangeMask	Any change in visibility wanted (by the application)
18	StructureNotifyMask	Any change in window structure wanted (by the application)

(continued on next page)

Table 1–1 (Cont.) Event Mask Elements

Bit	C Value	Description
19	ResizeRedirectMask	Redirect resize of this window
20	SubstructureNotifyMask	Substructure notification wanted (by the application)
21	SubstructureRedirectMask	Redirect substructure of window
22	FocusChangeMask	Any change in input focus wanted (by the application)
23	PropertyChangeMask	Any change in property wanted (by the application)
24	ColormapChangeMask	Any change in colormap wanted by the application
25	OwnerGrabButtonMask	Automatic grabs should activate with owner_events set to true

Description

Before calling XSelectInput to specify interest in a particular set of events for a window, clients can call XSelectAsyncInput to specify action routines and arguments to be called when some subset of those events occurs.

Xlib uses the client's **ast_routine** and **ast_userarg** information to deliver an AST whenever it places one of the specified events on its event queue. The **ast_routine** and **ast_userarg** arguments allow the client to specify the particular action routine and parameter pair to use when notifying the client that one of the specified events has occurred. The AST acts only as an event notification mechanism; the application uses the standard Xlib event routines to actually retrieve and process the event from the event queue.

Clients can call XSelectAsyncInput multiple times to specify different routine and parameter pairs for different sets of events for a window. The last call always takes precedence. If called with **ast_routine** equal to zero, asynchronous notification is disabled, but the current selection for the specified events is unaffected.

Notification ASTs are queued in the same order as events are placed in the event queue by Xlib. Therefore, clients can assume that they receive notification ASTs in the same order that they find events in the queue.

XSelectAsyncInput is similar to XSelectAsyncEvent except that XSelectAsyncEvent specifies one event type and XSelectAsyncInput specifies an event mask.

2
DECwindows Toolkit Extensions

This chapter describes the new and enhanced resources and routines Digital provides to supplement those included in the OSF/Motif Toolkit.

2.1 DECwindows Resource and Routine Extensions

The following sections describe three class-specific resources and an extension to the XmStringDirectionCreate routine.

2.1.1 DXmNautoUnrealize Resource

The DXmNautoUnrealize resource is a Boolean resource that, when False, indicates that the XmBulletinBoardDialog widget (and its subclasses) creates windows for itself and its children when it is first managed and never destroys (''unrealizes'') them. When True, the widget recreates the windows every time it is managed and destroys them when it is unmanaged.

When setting this resource, note the following effects on performance:

- The client central processing unit (CPU) load is highest when this resource is set to True.
- The server window load is highest when this resource is set to False.

The default is False.

2.1.2 DXmNfitToScreenPolicy Resource

The DXmNfitToScreenPolicy resource, which is specific only to the dialog shell widget, automatically sizes all dialog widgets for a screen. When the DXmNfitToScreenPolicy resource is set to AS_NEEDED in an application's defaults file, the dialog shell automatically resizes and positions all dialog shells that are too big for the user's screen.

You can set this resource only in an application's defaults file (access is CG); it cannot be set in a UIL module or through a call to XtSetArg. The format for setting this resource is as follows:

```
*DXmfitToScreenPolicy: AS_NEEDED
```

2.1.3 DXmNlayoutDirection Resource

The DXmNlayoutDirection resource is used by applications that require interpretation of direction, widget layout direction, and widget traversal direction. This resource allows the application to be configured for environments that expect direction from either left to right or from right to left. The choices (constants) for this resource are as follows:

Value	Description
DXmLAYOUT_LEFT_DOWN	The direction of the layout is right to left and top to bottom.
DXmLAYOUT_LEFT_UP	The direction of the layout is right to left and bottom to top.
DXmLAYOUT_RIGHT_DOWN	The direction of the layout is left to right and top to bottom.
DXmLAYOUT_RIGHT_UP	The direction of the layout is left to right and bottom to top.

The default is DXmLAYOUT_LEFT_DOWN.

The DXmNlayoutDirection resource can be inherited by any widget that inherits the XmPrimitive, XmManager, or XmGadget widget class.

2.1.3.1 DXmNlayoutDirection in DECwindows Extensions to Motif

The behavior of the DXmNlayoutDirection resource depends on the function or widget that you use it with. DXmNlayoutDirection performs as a read/write value that holds the layout direction of the widget components, such as labels, pop ups, push buttons, and scroll bars, in relation to each other for the following functions:

- DXmCreateColorMixDialog
- DXmCreateCSText
- DXmCreateHelpDialog
- DXmCreatePrintDialog

- DXmCreateScrolledCSText
- DXmCreateSvn

Table 2–1 describes the effect this resource has on the DXmLAYOUT_LEFT_DOWN and DXmLAYOUT_RIGHT_DOWN values for these functions.

Table 2–1 DXmNlayoutDirection Values

DXmLAYOUT_LEFT_DOWN	DXmLAYOUT_RIGHT_DOWN
The direction of the layout is right to left and top to bottom.	The direction of the layout is left to right and top to bottom. The default is DXmLAYOUT_RIGHT_DOWN, or 3. Access is CSG.

DXmNlayoutDirection performs as a read/write value that holds the layout direction of the vertical scroll bars in relation to the text for the DXmCreateScrolledCSText function. In addition, this resource controls the resizing direction for the widget width.

Note

The legal values for DXmNlayoutDirection, if specified in a resource file, are:

- left_down—layout is right to left and top to bottom.
- right_down—layout is left to right and top to bottom.

Table 2–2 describes the effect that this resource has on the DXmLAYOUT_LEFT_DOWN and DXmLAYOUT_RIGHT_DOWN values for this function.

Table 2–2 DXmNlayoutDirection Values for the DXmCreateScrolledCSText and DXmCreateCSText Functions

DXmLAYOUT_LEFT_DOWN	DXmLAYOUT_RIGHT_DOWN
The default value for XmNscrollLeftSide is TRUE. If XmNresizeWidth is TRUE, the left edge of the widget resizes.	The default value for XmNscrollLeftSide is FALSE. If XmNresizeWidth is TRUE, the right edge of the widget resizes. The default is DXmLAYOUT_RIGHT_ DOWN, or 3. Access is CSG.

2.1.3.2 DXmNlayoutDirection in DECwindows Motif

The following OSF/Motif widgets support DXmNlayoutDirection:

- XmLabelGadget class and XmLabelGadget subclasses:
  - XmCascadeButtonGadget
  - XmPushButtonGadget
  - XmToggleButtonGadget
- XmPrimitive subclasses:
  - XmLabel
  - XmList
  - XmScroll bar
  - XmCascadeButton
  - XmPushButton
  - XmToggleButton
- XmManager subclasses:
  - XmSelectionBox
  - XmForm
  - XmMessageBox
  - XmCommand
  - XmFileSelectionBox

Table 2–3 describes the effect that the DXmNlayoutDirection resource has on these widgets' components.

Table 2–3 DXmNlayoutDirection Effect on Widget Components

Widget Component	DXmLAYOUT_LEFT_DOWN	DXmLAYOUT_RIGHT_DOWN
CascadeButton / Gadget	Cascade graphic appears on the left side, and the menu is brought up on the left. XmALIGNMENT_BEGINNING aligns the right edges of lines and XmALIGNMENT_END aligns the left edges of lines.	Cascade graphic appears on the right side, and the menu is brought up on the right. XmALIGNMENT_BEGINNING aligns the left edges of lines and XmALIGNMENT_END aligns the right edges of lines.
Command	Prompt string is placed on the right side, with a default of "<".	Prompt string is placed on the left side, with a default of ">".
FileSelectionBox	Buttons are laid out from right to left, labels are aligned to the right side, scroll bars are aligned to the left side, and text is displayed from right to left.	Buttons are laid out from left to right, labels are aligned to the left side, scroll bars are aligned to the right side, and text is displayed from left to right.
Form	The word "near" means right when referenced in the *OSF/Motif Programmer's Reference* for the XmNrubberPositioning, XmNleftAttachment, XmNleftOffset, XmNleftPosition, and XmNleftWidget resource.	The word "near" is defined to mean left when referenced in the *OSF/Motif Programmer's Reference* for the XmNrubberPositioning, XmNleftAttachment, XmNleftOffset, XmNleftPosition, and XmNleftWidget resources.
	The word "far" is defined to mean left when referenced in the *OSF/Motif Programmer's Reference* for the XmNrightAttachment, XmNrightOffset, XmNrightPosition, and XmNrightWidget resources.	The word "far" is defined to mean right when referenced in the *OSF/Motif Programmer's Reference* for the XmNrightAttachment, XmNrightOffset, XmNrightPosition, and XmNrightWidget resources.
Label / Gadget	XmALIGNMENT_BEGINNING aligns the right edges of lines and XmALIGNMENT_END aligns the left edges of lines.	XmALIGNMENT_BEGINNING aligns the left edges of lines and XmALIGNMENT_END aligns the right edges of lines.

(continued on next page)

Table 2–3 (Cont.) DXmNlayoutDirection Effect on Widget Components

Widget Component	DXmLAYOUT_LEFT_DOWN	DXmLAYOUT_RIGHT_DOWN
Message Box	Buttons are laid out from right to left; pixmap is placed at the right side.	Buttons are laid out from left to right; pixmap is placed at the left side.
Option Menu	Label is aligned on right side and the bar graphic is placed on left side. If the pull-down menu cannot be placed over the button, the menu is displayed to the left side.	Label is aligned on the left side and bar graphic is placed on the right side. If the pull-down menu cannot be placed over the button, the menu is displayed to the right side.
PanedWindow	The sash is placed on the left side.	The sash is placed on the right side.
Popup Menu	The hot spot is located at the upper right corner.	The hot spot is located at the upper left corner.
Pull-Down Menu	If pulled down from a menu bar or a horizontal menu, the right edge of pull-down menu is aligned with the right edge of its parent cascade button.	If pulled down from a menu bar or a horizontal menu, the left edge of pull-down menu is aligned with the left edge of its parent cascade button.
Pushbutton /Gadget	The accelerator is displayed on the left side. XmALIGNMENT_ BEGINNING aligns the right edge of lines and XmALIGNMENT_END aligns the left edge of lines.	The accelerator is displayed on the right side. XmALIGNMENT_BEGINNING aligns the left edge of lines and XmALIGNMENT_END aligns the right edge of lines.
RowColumn	Children are laid out from right to left, including cascades in a menu bar.	Children are laid out from left to right, including cascades in a menu bar.
Scale	The text string is placed on the right side. If XmNorientation is XmVERTICAL, and XmNshowValue is TRUE, the value is displayed on the left side of the scale. If XmNorientation is XmHORIZONTAL, the default XmNprocessingDirection is XmMAX_ON_LEFT.	The text string is placed on the left side of the scale. If XmNorientation is XmVERTICAL and XmNshowValue is TRUE, the value is displayed on the right side of the scale. If XmNorientation is XmHORIZONTAL, the default XmNprocessingDirection is XmMAX_ON_RIGHT.

(continued on next page)

Table 2–3 (Cont.) DXmNlayoutDirection Effect on Widget Components

Widget Component	DXmLAYOUT_LEFT_DOWN	DXmLAYOUT_RIGHT_DOWN
scroll bar	If XmNorientation is XmHORIZONTAL, the default XmNprocessingDirection is XmMAX_ON_LEFT.	If XmNorientation is XmHORIZONTAL, the default XmNprocessingDirection is XmMAX_ON_RIGHT.
ScrolledWindow	The default XmNscrollbarPlacement is XmBOTTOM_LEFT.	The default XmNscrollbarPlacement is XmBOTTOM_RIGHT.
SelectionBox	The buttons are laid out from right to left, and labels are aligned on the right side.	The buttons are laid out from left to right, and labels are aligned on the left side.
ToggleButton / Gadget	The accelerator is displayed on the left side of the label, and the toggle graphic is placed on the right side of the label. XmALIGNMENT_BEGINNING aligns the right edges of lines, and XmALIGNMENT_END aligns the left edges of lines.	The accelerator is displayed on the right side of the label, and the toggle graphic is placed on the left side of the label. XmALIGNMENT_BEGINNING aligns the left edges of lines, and XmALIGNMENT_END aligns the right edges of lines.

2.1.4 XmStringDirectionCreate Routine

The XmStringDirectionCreate routine creates a compound string that specifies in which direction your system displays characters within a string or within a segment of a compound string. The data type you use to specify the direction is XmStringDirection.

In addition to specifying the left-to-right or right-to-left values for the XmStringDirection data type, a DECwindows extension allows you to specify that the application revert to the previous direction before displaying the next character in the string.

The functions of the XmStringDirection data types are summarized in the following table:

XmStringDirection Data Type	Description
XmSTRING_DIRECTION_L_TO_R	Sets the string direction left to right.
XmSTRING_DIRECTION_R_TO_L	Sets the string direction right to left.
XmSTRING_DIRECTION_REVERT	Reverts to the previous string direction (DECwindows extension).

2.2 DECwindows Toolkit Routines

The balance of this chapter describes, in alphabetical order, the Toolkit routines Digital has added to supplement those provided by the OSF/Motif Toolkit. These routines fall into the following general categories:

- Creation and manipulation routines that support the following DECwindows widgets:
  - ColorMix widget
  - Compound string text widget
  - Help widget
  - Print widget
  - Structured Visual Navigation (SVN) widget
- Convenience routines:
  - DXmActivateWidget
  - DXmChildren
  - DXmCreateCursor
  - DXmCSContainsStringCharSet
  - DXmDescToNull
  - DXmFormSpaceButtonsEqually
  - DXmGetLocaleString
  - DXmHelpOnContext
  - DXmInitialize
  - DXmNumChildren

- Conversion routines:
  - DXmCvtCStoDDIF
  - DXmCvtCStoFC
  - DXmCvtCStoOS
  - DXmCvtDDIFtoCS
  - DXmCvtFCtoCS
  - DXmCvtOStoCS
- DECwindows Motif Help System routines:
  - DXmHelpSystemClose
  - DXmHelpSystemDisplay
  - DXmHelpSystemOpen
- Font routines:
  - DXmFindFontFallback
  - DXmLoadQueryFont
- Message routines:
  - DXmDisplayCSMessage
  - DXmDisplayVMSMessage

The *DECwindows Motif Guide to Application Programming* contains complete information about the DECwindows-specific widgets, including guidelines (with examples) for using many of the routines described in this manual.

DXmActivateWidget

Allows the application to simulate push-button activation.

Format

```
void DXmActivateWidget(widget)
      Widget  widget;
```

Arguments

widget

A pointer to the widget data structure.

Description

The DXmActivateWidget routine allows an application to generate the same highlighting and callbacks that would occur if the user clicked on a push button. Using this routine in your application enables you to maintain a consistent user interface. For example, your application could allow a user to choose a function by either pressing a push button or, if the DXmActivateWidget routine has been called, by selecting a menu option that activates the corresponding push button.

The DXmActivateWidget routine only supports push buttons.

DXmChildren

Returns a list of the widget's children.

Format

```
WidgetList DXmChildren (widget)
            Widget  widget;
```

Return Value

A list of the widget's children.

Arguments

widget
The identifier (widget ID) of the widget.

Description

The DXmChildren routine returns a list of the widget's children. Note that children must request geometry management changes from their parent and that a parent widget can resize its children. You can use this routine to learn the length of the **widget_list** returned by DXmChildren.

DXmColorMixGetNewColor

Retrieves (returns) the ColorMix widget's current new color red, green, and blue values.

Format

```
void DXmColorMixGetNewColor(cmw, red, green, blue)
      DXmColorMixWidget  cmw;
      unsigned short     *red;
      unsigned short     *green;
      unsigned short     *blue;
```

Arguments

cmw
The widget identifier of the ColorMix widget.

red
A pointer to the returned new color red value.

green
A pointer to the returned new color green value.

blue
A pointer to the returned new color blue value.

Description

DXmColorMixGetNewColor allows the calling application or routine to quickly obtain the current color value from the ColorMix widget. Note that if your application uses the default color mixing subwidget, the application will perform this operation faster if you call this routine instead of the Intrinsic routine XtGetValues. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

Routine	Description
DXmColorMixSetNewColor	Sets the new color red, green, and blue values in the ColorMix widget
DXmCreateColorMix	Creates the ColorMix widget without a dialog box
DXmCreateColorMixDialog	Creates the ColorMix widget with a dialog box

DXmColorMixSetNewColor

Sets the new color red, green, and blue values in the ColorMix widget.

Format

```
void DXmColorMixSetNewColor(cmw, red, green, blue)
      DXmColorMixWidget  cmw;
      unsigned short     red;
      unsigned short     green;
      unsigned short     blue;
```

Arguments

cmw
The identifier (widget ID) of the ColorMix widget.

red
The new color red value. The value is expressed as an X color value (0 to 65535).

green
The new color green value. The value is expressed as an X color value (0 to 65535).

blue
The new color blue value. The value is expressed as an X color value (0 to 65535).

Description

The DXmColorMixSetNewColor routine allows the application or a user-supplied color mixer subwidget to pass the current color value to the ColorMix widget. Note that your application will perform this operation faster if you call this routine instead of the Intrinsic routine XtSetValues. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

Routine	Description
DXmColorMixGetNewColor	Retrieves (returns) the current color value from the ColorMix widget
DXmCreateColorMix	Creates the ColorMix widget without a dialog box
DXmCreateColorMixDialog	Creates the ColorMix widget with a dialog box

DXmCreateColorMix

Creates a ColorMix widget, without a pop-up dialog box.

Widget Class Hierarchy

Core Resource Set
\
Composite Resource Set
\
Constraint Resource Set
\
XmManager Resource Set
\
XmBulletinBoard Resource Set
\
DXmColorMix Resource Set

Format

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

Return Value

The identifier (widget ID) of the created ColorMix widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

arglist
The application argument list.

argcount
The number of arguments in the application argument list.

Description

The DXmCreateColorMix routine uses the same arguments and resources as the DXmCreateColorMixDialog routine to create a ColorMix widget, but does not include a pop-up dialog box. Note, however, that if color resources are limited, your application should use the pop-up ColorMix widget instead. See the DXmCreateColorMixDialog routine for complete information.

DXmCreateColorMixDialog

Creates a ColorMix widget with a dialog box.

Widget Class Hierarchy

Core Resource Set
\
Composite Resource Set
\
Constraint Resource Set
\
XmManager Resource Set
\
XmBulletinBoard Resource Set
\
DXmColorMixDialog Resource Set

Format

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

Resource Information

You can set the following widget-specific resources in the **arglist** when using any combination of default or application-provided color display and color mixer subwidgets:

```
XmString        DXmNclearLabel;
XmString        DXmNdisplayLabel;
Widget          DXmNdisplayWindow;
XmString        DXmNhelpLabel;
XmString        DXmNmainLabel;
XmString        DXmNmixerLabel;
Widget          DXmNmixerWindow;
unsigned short  DXmNnewBlueValue;
unsigned short  DXmNnewGreenValue;
unsigned short  DXmNnewRedValue;
unsigned short  DXmNorigBlueValue;
unsigned short  DXmNorigGreenValue;
unsigned short  DXmNorigRedValue;
XmString        DXmNresetLabelString;
XmString        DXmNscratchPadInfoLabel;
XmString        DXmNscratchPadLabel;
void proc       DXmNsetMixerColorProc;
void proc       DXmNsetNewColorProc;
XtCallbackList  XmNapplyCallback;
XmString        XmNapplyLabelString;
XtCallbackList  XmNcancelCallback;
XmString        XmNcancelLabelString;
XtCallbackList  XmNokCallback;
XmString        XmNokLabelString;
Widget          XmNworkWindow;
```

You can set the following widget-specific resources in the **arglist** only when using the default color display subwidget:

```
unsigned short  DXmNbackBlueValue;
unsigned short  DXmNbackGreenValue;
unsigned short  DXmNbackRedValue;
Dimension       DXmNdisplayColWinHeight;
Dimension       DXmNdisplayColWinWidth;
Dimension       DXmNdispWinMargin;
Boolean         DXmNmatchColors;
```

You can set the following widget-specific resources in the **arglist** only when using the default color mixer subwidget:

```
XmString        DXmNblackLabel;
XmString        DXmNblueLabel;
short           DXmNbrowserItemCount;
XmString        DXmNbrowserLabel;
unsigned char   DXmNcolorModel;
XmString        DXmNcoolerLabel;
XmString        DXmNdarkerLabel;
XmString        DXmNearthtoneLabel;
XmString        DXmNfullLabel;
XmString        DXmNgrayLabel;
XmString        DXmNgreenLabel;
XmString        DXmNgreyscaleLabel;
Boolean         DXmNgreyscaleOnGreyscale;
XmString        DXmNhlsLabel;
XmString        DXmNhueLabel;
short           DXmNinterpTileCount;
short           DXmNinterpTileHeight;
short           DXmNinterpTileWidth;
XmString        DXmNinterpTitleLabel;
XmString        DXmNlighterLabel;
XmString        DXmNlightLabel;
short           DXmNlightnessIncrement;
XmString        DXmNoptionLabel;
XmString        DXmNpastelLabel;
unsigned short  DXmNpickerColors;
XmString        DXmNpickerLabel;
short           DXmNpickerTileHeight;
short           DXmNpickerTileWidth;
XmString        DXmNpickerTitleLabel;
XmString        DXmNredLabel;
XmString        DXmNrgbLabel;
XmString        DXmNsatLabel;
XmString        DXmNsliderLabel;
XmString        DXmNsmearLabel;
XmString        DXmNspectrumLabel;
XmString        DXmNundoLabel;
XmString        DXmNuserPaletteLabel;
XmString        DXmNvalueLabel;
XmString        DXmNvividLabel;
XmString        DXmNwarmerLabel;
unsigned short  DXmNwarmthIncrement;
XmString        DXmNwhiteLabel;
```

Return Value

The identifier (widget ID) of the created ColorMix widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

arglist
The application argument list.

argcount
The number of arguments in the application argument list.

Resources (Default or Application-Provided Color Display/Mixer Subwidgets)

DXmNclearLabel
Specifies the text that appears as the label of the scratch pad clear push button. The default text is ‘‘Clear’’. Access is CSG.

DXmNdisplayLabel
Specifies the text that appears above the color display subwidget, centered between the left and right borders. There is no default. Access is CSG.

DXmNdisplayWindow
The color display subwidget. The default is the ColorMix widget’s color display subwidget. Access is CSG.

Note the following:

- Setting this resource to Null at widget creation time prevents the color display subwidget from being displayed.
- If an application substitutes its own color display subwidget for the default color display subwidget, the application is completely responsible for managing the subwidget (making it visible and controlling its geometry management). An application can return to the default color display subwidget by using the Intrinsic routine XtSetValues to set the **DXmNdisplayWindow** resource to Null. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

DXmNhelpLabel
Specifies the text that appears inside the Help push button. The default text is ''Help''. Access is CSG.

DXmNmainLabel
Specifies the text that appears at the top of the ColorMix widget, centered between the left and right borders. There is no default. Access is CSG.

DXmNmixerLabel
Specifies the text that appears above the color mixer subwidget, centered between the left and right borders. There is no default. Access is CSG.

DXmNmixerWindow
The color mixer subwidget. The default color mixer subwidget supports the hue, lightness, saturation (HLS), red, green, blue (RGB), Color Picker, Browser, and Greyscale Mixer color models. Setting this resource to Null at widget creation time prevents the color mixer subwidget from being displayed. Access is CSG.

Note the following:

- If an application substitutes its own color mixer subwidget for the default color mixer subwidget, the application is completely responsible for managing the subwidget (making it visible and controlling its geometry management). An application can later return to the default color mixer subwidget by using the Intrinsic routine XtSetValues to set the **DXmNmixerWindow** resource to Null. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)
- Applications that use the default color mixer subwidget are not required to update the new color value. However, applications that provide their own color mixer subwidget must update the new color value, by using either the Intrinsic routine XtSetValues or the DXmColorMixSetNewColor routine (which is faster).

DXmNnewBlueValue
The new blue color value for the ColorMix widget. The default is 0 unless **DXmNmatchColors** is True. If **DXmNmatchColors** is True, then **DXmNnewBlueValue** is always set by default to match **DXmNorigBlueValue** whenever the widget is created or made visible. Access is CSG.

DXmNnewGreenValue
The new green color value for the ColorMix widget. The default is 0 unless **DXmNmatchColors** is True. If **DXmNmatchColors** is True, then **DXmNnewGreenValue** is always set by default to match **DXmNorigGreenValue** whenever the widget is created or made visible. Access is CSG.

DXmNnewRedValue
The new red color value for the ColorMix widget. The default is 0 unless **DXmNmatchColors** is True. If **DXmNmatchColors** is True, then **DXmNnewRedValue** is always set by default to match **DXmNorigRedValue** whenever the widget is created or made visible. Access is CSG.

DXmNorigBlueValue
The original blue color value for the ColorMix widget. The default is 0; however, applications should set the original blue value. Access is CSG.

DXmNorigGreenValue
The original green color value for the ColorMix widget. The default is 0; however, applications should set the original green value. Access is CSG.

DXmNorigRedValue
The original red color value for the ColorMix widget. The default is 0; however, applications should set the original red value. Access is CSG.

DXmNresetLabelString
Specifies the text that appears inside the Reset push button. The default is ‘‘Reset’’. Access is CSG.

DXmNscratchPadInfoLabel
Specifies the text that appears as the label centered at the top of the scratch pad pop-up. The default text is ‘‘Save colors here...’’. Access is CSG.

DXmNscratchPadLabel
Specifies the text that appears as the label of the scratch pad push button. The default text is ‘‘Scratch Pad...’’. Access is CSG.

DXmNsetMixerColorProc
The resource used by the ColorMix widget to update the current color mixer controls when the application sets a new color. (It is intended for applications that use a non-default color mixer subwidget.) The default is a procedure (specific to the ColorMix widget) that updates the appropriate default color model (Picker, HLS, RGB, Browser, or Greyscale). Access is CSG.

DXmNsetNewColorProc
The resource used by the ColorMix widget to update the new color values displayed in the color display subwidget. If the application replaces the default color display subwidget and requires that the ColorMix widget update the new color, the application must set this resource. Otherwise, replacing the default color display subwidget sets this resource to Null. Access is CSG.

If this resource is set and the application manages this subwidget, the work window is placed below the color display and color mixer subwidgets (if present) and above the ColorMix widget push buttons.

XmNapplyCallback
Callback (or callbacks) made when the user clicks the Apply push-button widget in the ColorMix widget. The reason for this callback is **XmCR_APPLY**. The default is Null. Access is CSG.

XmNapplyLabelString
Specifies the text that appears inside the Apply push button. The default is ‘‘Apply’’. Access is CSG.

XmNcancelCallback
Callback (or callbacks) made when the user clicks the Cancel push-button widget in the ColorMix widget. The reason for this callback is **XmCR_CANCEL**. The default is Null. Access is CSG.

XmNcancelLabelString
Specifies the text that appears inside the Cancel push button. The default is ‘‘Cancel’’. Access is CSG.

XmNokCallback
Callback (or callbacks) made when the user clicks the OK push-button widget in the ColorMix widget. The reason for this callback is **XmCR_ACTIVATE**. The default is Null. Access is CSG.

XmNokLabelString
Specifies the text that appears inside the OK push button. The default is ‘‘OK’’. Access is CSG.

XmNworkWindow
An optional work area subwidget. The default is Null. Access is CSG.

Resources (Default Color Display Subwidget Only)

DXmNbackBlueValue

The value of the default color display subwidget's blue background color. The default value is 32767 (gray). Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

DXmNbackGreenValue

The value of the default color display subwidget's green background color. The default value is 32767 (gray). Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

DXmNbackRedValue

The value of the default color display subwidget's red background color. The default value is 32767 (gray). Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

DXmNdisplayColWinHeight

The height of the original and new color display windows. The default is 80 pixels. Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

DXmNdisplayColWinWidth

The width of the original and new color display windows. The default is 80 pixels. Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

DXmNdispWinMargin

The margin between the original and the new color display windows and the outside edges of the color display subwidget. The margin is the area affected by the background resources (set gray by default). The default width is 20 pixels. Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

DXmNmatchColors

A Boolean resource that specifies whether new color values are matched to original color values whenever the ColorMix widget is created and made visible. When True, new color values are matched to original color values. When False, new color values are not matched to original color values. The default is True. Access is CSG.

This resource is valid only if the default color display subwidget is used; otherwise, the resource is ignored.

Resources (Default Color Mixer Subwidget Only)

DXmNblackLabel

Specifies the text that appears below the left end of the middle scale subwidget. The default is ‘‘Black’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNblueLabel

Specifies the text that appears below the left end of the bottom scale subwidget. The default is ‘‘Blue’’. Access is CSG.

This resource is used in the RGB mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNbrowserItemCount

The number of visible items in the browser’s color list. You can set this resource only at widget creation time. Each visible item requires that the ColorMix widget allocate its own color cell. The default is 5. Access is CG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNbrowserLabel

Specifies the text that appears as the fourth item in the color model option menu. The default text is ‘‘Browser’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNcolorModel

The color model currently being used. The values for this resource are as follows:

Color Model	Value	When Default
Color Picker	DXmColorModelPicker	Color systems
HLS	DXmColorModelHLS	Not used as default
RGB	DXmColorModelRGB	Monochrome systems
Browser	DXmColorModelBrowser	Not used as default
Greyscale	DXmColorModelGreyscale	Gray scale systems

Applications should set this resource before the widget is managed (if at all), and allow users to switch color models by using the option menu. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNcoolerLabel

Specifies the text of the Cooler arrow button. The default text is ‘‘Cooler’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNdarkerLabel

Specifies the text of the Darker arrow button. The default text is ‘‘Darker’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNearthtoneLabel

The label used to represent the earthtones palette in the picker spectrum option menu. The default text is ‘‘Earthtones’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNfullLabel

Specifies the text that appears below the right end of the bottom scale subwidget. The default is ‘‘Full’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNgrayLabel
Specifies the text that appears below the left of the bottom scale subwidget. The default is ‘‘Gray’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNgreenLabel
Specifies the text that appears below the left end of the middle scale subwidget. The default is ‘‘Green’’. Access is CSG.

This resource is used in the RGB mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNgreyscaleLabel
Specifies the text that appears as the bottom item in the color model option menu. The default text is ‘‘Grayscale’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNgreyscaleOnGreyscale
A Boolean resource that, when True, specifies that the greyscale mixing color model is to be the default on greyscale systems. The default is True. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNhlsLabel
Specifies the text that appears as the second item in the color model option menu. The default is ‘‘HLS’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNhueLabel
Specifies the text that appears to the left of the top scale subwidget. The default is ‘‘Hue:’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNinterpTileCount
The number of interpolator tiles used. (The ColorMix widget must allocate a color cell for each interpolator tile.) You can set this resource only at widget creation time. The default is 10. Access is CG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNinterpTileHeight
The height of each interpolator tile, in pixels. The default is 30. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNinterpTileWidth
The width of each interpolator tile, in pixels. The default is 30. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNinterpTitleLabel
Specifies the text that appears centered above the interpolator widget. The default text is ‘‘Interpolator’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNlighterLabel
Specifies the text of the Lighter arrow button. The default text is ‘‘Lighter’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNlightLabel
Specifies the text that appears to the left of the bottom scale subwidget. The default is ‘‘Lightness:’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNlightnessIncrement
The percentage by which to increase or decrease the lightness of the color of each interpolator tile when the lighter or darker buttons are pressed. The default is 5. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNoptionLabel

Specifies the text that appears inside the color model option menu subwidget. The default is ‘‘Color Model:’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNpastelLabel

The label used to represent the pastels palette in the picker spectrum option menu. The default text is ‘‘Pastels’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNpickerColors

A palette of ten colors that will be available through the user palette menu option. If not specified, no user palette will appear in the menu; otherwise, the user palette will become the default picker palette. You can set this resource only at widget creation time. Access is CG.

Note that the first item in the array is the red value of the first spectrum tile, the second item is its green value, the third item is its blue value, the fourth item is the second tile’s red value, and so on. Thus, this array must contain thirty elements if it is set.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNpickerLabel

Specifies the text that appears as the top item in the color model option menu. The default text is ‘‘Picker’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNpickerTileHeight

The height of each individual spectrum tile, in pixels. The default is 30. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNpickerTileWidth
The width of each individual spectrum tile, in pixels. The default is 30. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNpickerTitleLabel
Specifies the text that appears centered above the picker widget. The default text is ‘‘Spectrum’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNredLabel
Specifies the text that appears below the left end of the top scale subwidget. The default is ‘‘Red’’. Access is CSG.

This resource is used in the RGB mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNrgbLabel
Specifies the text that appears as the third item in the color model option menu. The default is ‘‘RGB’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNsatLabel
Specifies the text that appears to the left of the middle scale subwidget. The default is ‘‘Saturation:’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNsliderLabel
Specifies the text that appears above the left end of the top scale subwidget. The default is ‘‘Percentage’’. Access is CSG.

This resource is used in the RGB mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNsmearLabel
Specifies the text of the Smear push button. The default text is ‘‘Smear’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNspectrumLabel

The label used to represent the ‘‘rainbow’’ palette in the picker spectrum option menu. The default text is ‘‘Spectrum’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNundoLabel

Specifies the text of the Undo push button. The default text is ‘‘Undo’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNuserPaletteLabel

The label used to represent the application-defined user palette in the spectrum option menu. The default text is ‘‘User palette’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNvalueLabel

Specifies the text that appears above the column of text subwidgets. The default is ‘‘Value’’. Access is CSG.

This resource is used in the RGB mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNvividLabel

The label used to represent the vivids palette in the picker spectrum option menu. The default text is ‘‘Vivids’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNwarmerLabel

Specifies the text of the Warmer arrow button. The default text is ‘‘Warmer’’. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNwarmthIncrement

The amount of red or blue added to the color of each interpolator tile when the warmer or cooler buttons are pressed. The default is 5000. Access is CSG.

This resource is valid only if the default color mixer subwidget is used; otherwise, the resource is ignored.

DXmNwhiteLabel

Specifies the text that appears below the right end of the middle scale subwidget. The default is ‘‘White’’. Access is CSG.

This resource is used in the HLS mixer if the default color mixer subwidget is used; otherwise, the resource is ignored.

Resource Exceptions

The following resources inherited from XmBulletinBoard are supported differently:

- The default for XmNdialogStyle is **XmDIALOG_MODELESS.**
- The default for XmNdialogTitle is ‘‘Color Mixing’’.
- The default for XmNresizePolicy is **XmRESIZE_NONE.**

Callback Structure

```
typedef struct
{
      int             reason;
      XEvent          *event;
      unsigned short  newred;
      unsigned short  newgrn;
      unsigned short  newblu;
      char            *newname;
      unsigned short  origred;
      unsigned short  origgrn;
      unsigned short  origblu;
} DXmColorMixCallbackStruct;
```

Callback Field Descriptions

reason

An integer set to the callback reason. See the Callback Reasons section for the values that are valid for this widget.

event
A pointer to the X event structure describing the event that generated this callback.

newred
The new red color value for the ColorMix widget.

newgrn
The new green color value for the ColorMix widget.

newblu
The new blue color value for the ColorMix widget.

newname
Passes a color name to an application. If the user selects a named color from the Browser and then triggers a callback to the application without modifying the new color, the **newname** field of the callback data structure is filled in with a pointer to an ASCII, null-terminated string that contains the color's X11 name. This string is read-only, and should not be freed or modified.

If a color is generated in one of the other color models, or generated in the browser and subsequently modified, the **newname** field in the callback structure is set to Null.

origred
The original red color value for the ColorMix widget.

origgrn
The original green color value for the ColorMix widget.

origblu
The original blue color value for the ColorMix widget.

Callback Reasons

XmCR_ACTIVATE
The user has activated the OK push button.

XmCR_APPLY
The user has activated the Apply push button.

XmCR_CANCEL
The user has activated the Cancel push button.

Description

The ColorMix widget allows your DECwindows Motif applications to query users for a color. The ColorMix widget provides users with immediate feedback, displaying each new color as it is selected. This widget uses a pop-up dialog box (or nonpop-up dialog box if you use the DXmCreateColorMix routine) that is preconfigured to contain the following subwidgets:

Subwidget	Description
Scratch pad	Stores intermediate colors for later use in color mixing.
Color display subwidget	Displays the original color and the new color.
Color model option menu subwidget	Implements choice of color model.
Color mixer subwidget	Provides graphic tools users can use to define new colors.
Push button subwidgets	Activates ColorMix widget functions.
Label subwidgets	Provides descriptive information.

The ColorMix widget allows a user to choose from up to five different color models:

- Color Picker

  The default color model for the ColorMix widget on color systems (it does not display on noncolor systems or on systems with too few resources).

  The Color Picker consists of a 10-color color spectrum and a color palette option menu, a color interpolator and its title, a smear button, and undo button, two paint bucket buttons, and arrow buttons to control the warmth and lightness of interpolator colors.

- HLS (hue, lightness, saturation)

  In the HLS color model, a color is specified by three characteristics: hue, lightness, and saturation. Hue is color. Lightness describes the intensity of the color (the amount of the color). Saturation describes the purity of the color or how much the color is diluted by white.

  To support the HLS color model, the color mixer subwidget contains three scales that represent the ranges of hue, lightness, and saturation. The hue scale presents color values as a range between 0 and 360. The lightness

and saturation scales present their values as a range of percentages between 0 and 100.

- RGB (red, green, blue)

  The default color model for the ColorMix widget on monochrome systems.

  In the RGB color model, a color is specified as a mixture of different intensities of red, green, and blue. In the X Window System, Version 11, you specify the intensity of red, green, or blue as a value between 0 and 65,535. Zero is the lowest intensity. Black is defined as a zero-intensity value for all three colors; white is 100 percent intensity for all three colors.

  To support the RGB color model, the color mixer subwidget contains three scales that represent the ranges of intensity of red, green, and blue. Each scale presents these color values as a percentage between 0 and 100. In addition, the color mixer subwidget contains text widgets in which users of your application can enter RGB values directly as text. The text widgets and the scales are linked: a change in one effects a corresponding change in the other.

- Browser (X11 named colors)

  A scrolled window that presents the user with a list of X11 named colors. Each button in the scrolled window shows the name of an X11 color. If enough resources are available, the background is set to that color. (If the new color is not further modified by other color models, the X11 name of this color is also returned to the application as part of the ColorMix widget's callback structure.) You can use the scroll bar to scroll through this color list. Clicking MB1 on a color in the list causes the color display subwidget to become filled with that color. The color browser is available on all systems.

- Greyscale Mixer

  A scale widget that allows generation of gray shades ranging from black to white. When the Greyscale mixer is selected, the current new color is converted to an appropriate shade of gray, which can then be adjusted with the scale widget. All shades of gray generated by this model are a mixture of equal portions of red, green, and blue, which means that they appear identically on both color and gray scale systems.

  The Greyscale mixer is available on all systems.

Note

When considering whether to use a pop-up or nonpop-up ColorMix widget, note the following:

- The pop-up ColorMix widget (which you create with the DXmCreateColorMixDialog routine) allocates the necessary color cells each time it is mapped and then releases them each time the widget is unmapped.
- The nonpop-up ColorMix widget (which you create with the DXmCreateColorMix routine) allocates the necessary color cells at creation time but does not release them until the widget is destroyed. As such, if color resources are limited, your application should use the pop-up ColorMix widget instead of the nonpop-up version.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the ColorMix widget. See also the following routines for related information:

Routine	Description
DXmCreateColorMix	Creates the ColorMix widget without a dialog box.
DXmColorMixGetNewColor	Retrieves (returns) the current color value from the ColorMix widget.
DXmColorMixSetNewColor	Sets the new color red, green, and blue values in the ColorMix widget.

Geometry Management

The ColorMix widget completely controls the layout of its children.

Resizing

The ColorMix widget computes the smallest possible size required to enclose all of its children, including any of the variably sized color mixing subwidgets. The size remains fixed while the user selects different color models from the option menu.

DXmCreateCSText

Creates a compound string text widget.

Widget Class Hierarchy

Core Resource Set
  \
  XmPrimitive Resource Set
    \
    DXmCSText Resource Set

Format

```
Widget DXmCreateCSText (parent, name, args, num_args)
        Widget   parent;
        char     *name;
        Args     *args;
        int      num_args;
```

Resource Information

You can set the following widget-specific resources in the **args**:

```
Boolean             DXmNbidirectionalCursor;
int                 DXmNeditingPath;
XtCallbackList      DXmNnofontCallback;
int                 DXmNtextPath;
XtCallbackList      XmNactivateCallback;
Boolean             XmNautoShowCursorPosition;
int                 XmNblinkRate;
int                 XmNcolumns;
DXmCSTextPosition   XmNcursorPosition;
Boolean             XmNcursorPositionVisible;
Boolean             XmNeditable;
int                 XmNeditMode;
XtCallbackList      XmNfocusCallback;
XmFontlist          XmNfontList;
XtCallbackList      XmNlosingFocusCallback;
Dimension           XmNmarginHeight;
Dimension           XmNmarginWidth;
int                 XmNmaxLength;
XtCallbackList      XmNmodifyVerifyCallback;
XtCallbackList      XmNmotionVerifyCallback;
Boolean             XmNpendingDelete
Boolean             XmNresizeHeight:
Boolean             XmNresizeWidth;
int                 XmNrows;
Boolean             XmNscrollHorizontal;
Boolean             XmNscrollLeftSide;
Boolean             XmNscrollTopSide;
Boolean             XmNscrollVertical;
XmRPointer          XmNselectionArray
int                 XmNselectionArrayCount
int                 XmNselectThreshold
DXmCSTextPosition   XmNtopPosition;
XmString            XmNvalue;
XtCallbackList      XmNvalueChangedCallback;
Boolean             XmNwordWrap;
```

Return Value

The identifier (widget ID) of the created compound string text widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

args
The application override argument list.

num_args
An integer that represents the number of arguments in the application override argument list.

Resources

DXmNbidirectionalCursor
A Boolean resource that, when True, specifies that the shape of the cursor at the insertion point will be dependent on the current editing direction. Access is CSG. The default is False.

DXmNeditingPath
A read-only value that holds the current editing text path (direction) in the compound string text widget. Initially, it is set to be equal to **DXmNtextPath**. This resource is used only if **XmNvalue** is Null. The values for this resource are as follows:

Value	Description
DXmDIRECTION_RIGHT_DOWN (0)	The direction of the text path is right and then down.
DXmDIRECTION_LEFT_DOWN (1)	The direction of the text path is left and then down.

The default is DXmDIRECTION_RIGHT_DOWN, or 0. Access is CG.

DXmNnofontCallback

Callback made when the compound string text widget cannot find the font required to display text assigned (tagged) with a specific character set. The reason for this callback is **DXmCR_NOFONT**. Other callback fields are as follows:

Field	Description
event	A pointer to the XEvent structure that generated the callback.
charset	A character set identifier for which the widget has no matching font in its font list. The callback modifies the widget font list to include an entry for the required character set.
charset_len	The length of the **charset** string.

The default is Null. Access is C.

DXmNtextPath

A read-only value that holds the main text path (direction) of the text in the compound text widget. This resource is used only if **XmNvalue** is Null. It is set from the initial compound string value of the widget. The values for this resource are as follows:

Value	Description
DXmDIRECTION_RIGHT_DOWN (0)	The direction of the text path is right and then down.
DXmDIRECTION_LEFT_DOWN (1)	The direction of the text path is left and then down.

The default is DXmDIRECTION_RIGHT_DOWN, or 0. Access is CG.

XmNactivateCallback

Callback (or callbacks) made when the user activates the CSText widget. The reason for this callback is **XmCR_ACTIVATE** and the associated callback structure is DXmCSTextCallbackStruct. The default is Null. Access is C.

XmNautoShowCursorPosition

A Boolean resource that, when True, ensures that the visible text contains the cursor. The default is True. Access is CSG.

XmNblinkRate
An integer indicating the blink rate of the text cursor in milliseconds. The default is 500 milliseconds. Access is CSG.

XmNcolumns
The width, in characters, of the window. The default is 20 characters (if no width is specified with the XmNwidth resource). Access is CSG.

XmNcursorPosition
An integer indicating the current location of the cursor. The default is 0. Access is CSG.

XmNcursorPositionVisible
A Boolean resource that, when True, specifies that the cursor position is marked by a blinking text cursor. The default is True. Access is CSG.

XmNeditable
A Boolean resource that, when True, indicates that the user can edit the text in the compound string text widget. When False, the user cannot edit the text. Access is CSG. The default is True.

XmNeditMode
Specifies the set of keyboard bindings used for CSText. The default set of keyboard bindings is XmSINGLE_LINE_EDIT. For multiline text, use XmMULTI_LINE_EDIT. Access is CSG.

XmNfocusCallback
Callback (or callbacks) made when the compound string text widget has accepted the input focus. The reason for this callback is **XmCR_FOCUS** and the associated callback structure is DXmCSTextCallbackStruct. The default is Null. Access is C.

XmNfontList
The font list to be used for the compound string text widget. The default is the current server font list (DXmDefaultFont). Access is CSG.

XmNlosingFocusCallback

Callback (or callbacks) made when the compound string text widget loses input focus. The callback reason is **XmCR_LOSING_FOCUS** and the associated callback structure is XmTextVerifyCallbackStruct. Other callback fields are as follows:

Field	Description
event	A pointer to the XEvent that generated this callback.
doit	Controls whether the action that invoked the callback is performed. Setting this field to False negates the action.
currInsert	The current position of the insertion cursor.

The default is Null. Access is C.

XmNmarginHeight

The number of pixels between the top or bottom edge of the window and the text. The default is 6 pixels. Access is CSG.

XmNmarginWidth

The number of pixels between the left or right edge of the window and the text. The default is 6 pixels. Access is CSG.

XmNmaxLength

The maximum length of the text, in characters, in the compound string text widget. The default is the largest integer (MAXINT). Access is CSG.

XmNmodifyVerifyCallback

Callback (or callbacks) made before text is deleted from or inserted into the CSText widget. The callback reason is **XmCR_MODIFYING_TEXT_VALUE** and the associated callback structure is DXmCSTextVerifyCallbackStruct. Other callback fields are as follows:

Field	Description
event	A pointer to the XEvent that generated this callback.
doit	Controls whether the action that invoked the callback is performed. Setting this field to False negates the action.
currInsert	The current position of the insertion cursor.
newInsert	The position of the cursor after the action has been completed.

Field	Description
startPos	The starting position of the text that is to be modified.
endPos	The ending position of the text that is to be modified. If no text is replaced or deleted, the value is the same as **startPos**.
text	A pointer to an XmString holding the text to be inserted.

The default is Null. Access is C.

XmNmotionVerifyCallback

Callback (or callbacks) made before the cursor is moved to another position. The callback reason is **XmCR_MOVING_INSERT_CURSOR** and the associated callback structure is XmTextVerifyCallbackStruct. Other callback fields are as follows:

Field	Description
event	A pointer to the XEvent that generated this callback.
doit	Controls whether the action that invoked the callback is performed. Setting this field to False negates the action.
currInsert	The current position of the insertion cursor.
newInsert	The position of the cursor after the action has been completed.

The default is Null. Access is C.

XmNpendingDelete

A Boolean resource that, when True, specifies that selected text containing the insertion point is to be deleted when new text is entered. The default is True. Access is CSG.

XmNresizeHeight

A Boolean resource that specifies whether the compound string text widget resizes its height to accommodate all the text contained in the widget. When True, the compound string text widget resizes its height and the text is always displayed starting from the first position in the source, even if instructed otherwise. This resource is ignored if **XmNscrollVertical** is True. The default is False. Access is CSG.

XmNresizeWidth
A Boolean resource that specifies whether the compound string text widget resizes its width to accommodate all the text contained in the widget. When True, the compound string text widget resizes its width. This resource is ignored if **XmNwordWrap** is True. The default is False. Access is CSG.

XmNrows
The height, in characters, of the window. This resource is ignored if the XmNeditMode resource is XmSINGLE_LINE_EDIT. The default is 1 character (if no height has been specified with the XmNheight resource). Access is CSG.

XmNscrollHorizontal
A Boolean resource that, when True, adds a scroll bar that allows the user to scroll through the text horizontally. The default is False. Access is CG.

This resource is valid only for the DXmCreateScrolledCSText routine.

XmNscrollLeftSide
A Boolean resource that, when True, places the vertical scroll bar on the left side of the compound string text window. This resource is ignored if the **XmNscrollVertical** resource is False. Access is CG.

This resource is valid only for the DXmCreateScrolledCSText routine.

XmNscrollTopSide
A Boolean resource that, when True, places the horizontal scroll bar at the top side of the compound string text window. The default is False. Access is CG.

This resource is valid only for the DXmCreateScrolledCSText routine.

XmNscrollVertical
A Boolean resource that, when True, adds a scroll bar that allows the user to scroll through the text vertically. The default is False. Access is CG.

This resource is valid only for the DXmCreateScrolledCSText routine.

XmNselectionArray

Specifies actions that result from multiple mouse clicks. The value of this resource is an array of XmTextScanType elements. The following values (listed in the order they occur in the default array) indicate the possible actions:

Value	Description
XmSELECT_POSITION	Resets the insertion cursor position
XmSELECT_WORD	Selects a word
XmSELECT_LINE	Selects a line of text
XmSELECT_ALL	Selects all of the text

The default is the default array. Access is CSG.

XmNselectionArrayCount

The number of elements in the **XmNselectionArray** resource. The default is 4. Access is CSG.

XmNselectThreshold

The number of pixels of motion required to select the next character during a mouse click and drag operation. The default is 5 pixels. Access is CSG.

XmNtopPosition

Displays the position of text at the top of the window. The default is 0. Access is CSG.

XmNvalue

The text contents of the compound string text widget. The default is Null. When Null, the text path and editing paths are set to DXmDIRECTION_ RIGHT_DOWN. Otherwise, the text and editing paths are set from the direction of the first segment of the value. The access is CSG.

XmNvalueChangedCallback

Callback (or callbacks) made when the value of the compound string text widget changes. The reason for this callback is **XmCR_VALUE_CHANGED** and the associated callback structure is DXmCSTextCallbackStruct. The default is Null. Access is C.

XmNwordWrap

A Boolean resource that specifies whether word wrap is set. When True, lines are broken at word breaks and text does not run off the right edge of the window. This resource is ignored if the XmNeditMode resource is set to XmSINGLE_LINE_EDIT. The default is False. Access is CSG.

Resource Exceptions

The DXmCSText widget supports the Core resources **XmNwidth** and **XmNheight** differently in that the defaults for both resources are set as large as necessary to display the rows and columns with the given margin width and margin height.

Callback Information

Following are descriptions of the two callback structures used by the CSText widget:

DXmCSTextCallback Structure

```
typedef struct
{
      int           reason;
      XEvent        *event;
      char          *charset;
      unsigned int  charset_len;
} DXmCSTextCallbackStruct;
```

DXmCSTextCallback Fields

reason
An integer set to the callback reason.

event
A pointer to the XEvent structure that generated the callback.

charset
A character set identifier for which the widget has no matching font in its font list. The callback modifies the widget font list to include an entry for the required character set.

charset_len
The length of the **charset** string.

DXmCSTextCallback Reasons

The following table lists the reasons and valid callback fields:

Reason	Description	Valid Fields
DXmCR_NOFONT	There was no entry in the widget font list for the required character set.	reason, event, charset, charset_len
XmCR_ACTIVATE	The user activated the CSText widget.	reason, event
XmCR_FOCUS	The compound string text widget has received the input focus.	reason, event
XmCR_HELP	The user selected help.	reason, event
XmCR_VALUE_CHANGED	The user changed the value of the text in the compound string text widget.	reason

XmTextVerifyCallback Structure

```
typedef struct
{
      int               reason;
      XEvent            *event;
      Boolean           doit;
      DXmCSTextPosition currInsert, newInsert;
      DXmCSTextPosition startPos, endPos;
      XmString          text;
} XmTextVerifyCallbackStruct, *XmTextVerifyPtr;
```

XmTextVerifyCallback Fields

reason
An integer set to the callback reason.

event
A pointer to the XEvent structure that generated the callback.

doit
Controls whether the action that invoked the callback is performed. Setting this field to False negates the action.

currInsert
The current position of the insertion cursor.

newInsert
The position of the cursor after the action has been completed. (Used only by the XmNmodifyVerifyCallback and XmNmotionVerifyCallback callback routines.)

startPos
The starting position of the text that is to be modified. (Used only by the XmNmodifyVerifyCallback callback routine.)

endPos
The ending position of the text that is to be modified. If no text is replaced or deleted, the value is the same as **startPos**. (Used only by the XmNmodifyVerifyCallback callback routine.)

text
A pointer to a compound string containing the text that is to be inserted. (Used only by the XmNmodifyVerifyCallback callback routine.)

XmTextVerifyCallback Reasons

The following table lists the reasons and valid callback fields:

Reason	Description	Valid Fields
XmCR_LOSING_FOCUS	The compound string text widget has lost the input focus.	reason, event, doit, currInsert
XmCR_MODIFYING_TEXT_VALUE	The user is deleting or inserting text.	reason, event, doit, currInsert, newInsert, startPos, endPos, text
XmCR_MOVING_INSERT_CURSOR	The user is moving the insertion cursor to another position.	reason, event, doit, currInsert, newInsert

DECwindows Toolkit Extensions
DXmCreateCSText

Description

The DXmCreateCSText routine creates a compound string text widget that provides your application with text editing capabilities, including the ability for users to enter and edit text in the same character set and writing direction used throughout the user interface. By default, the text window expands or shrinks as the user enters or deletes text characters. Note that the text window does not shrink below the initial size set at creation time.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the compound string text widget. See also the following routines for related information:

Routine	Description
DXmCSTextClearSelection	Clears the global selection highlighted in the compound string text widget.
DXmCSTextCopy	Copies the currently selected (highlighted) text to the clipboard.
DXmCSTextCut	Deletes the currently selected (highlighted) text after copying it to the clipboard.
DXmCSTextGetEditable	Indicates whether the text in the compound string text widget can be edited by the user.
DXmCSTextGetInsertionPosition	Retrieves (returns) the logical position of the insertion cursor.
DXmCSTextGetLastPosition	Retrieves (returns) the logical position corresponding to the last character in the text.
DXmCSTextGetMaxLength	Retrieves (returns) the current maximum allowable length of the text in the compound string text widget.
DXmCSTextGetSelection	Retrieves the text selected in the compound string text widget.
DXmCSTextGetSelectionInfo	Retrieves (returns) the left and right positions of the currently selected text.
DXmCSTextGetString	Retrieves all the text from the compound string text widget.

Routine	Description
DXmCSTextGetTopPosition	Retrieves (returns) the logical position of the first character in the displayed text.
DXmCSTextHasSelection	Indicates whether the compound string text widget currently owns the primary selection.
DXmCSTextHorizontalScroll	Scrolls text horizontally.
DXmCSTextInsert	Inserts new text into the compound string text widget.
DXmCSTextNumLines	Retrieves (returns) the number of visible lines in the compound string text widget.
DXmCSTextPaste	Pastes the data from the clipboard into the text at the current cursor position.
DXmCSTextPosToXY	Retrieves (returns) the x and y position of a specified character in the text.
DXmCSTextRemove	Removes the currently selected (highlighted) text.
DXmCSTextReplace	Replaces a specified segment of text in the compound string text widget.
DXmCSTextSetAddMode	Controls whether the CSText widget is in Add Mode (in relation to insertion cursor movement).
DXmCSTextSetEditable	Specifies whether the text in the widget can be edited by the user.
DXmCSTextSetHighlight	Changes the highlighting of the compound string text.
DXmCSTextSetInsertionPosition	Sets the insertion cursor to the given logical position in the source.
DXmCSTextSetMaxLength	Sets the maximum allowable length of the text in the compound string text widget.
DXmCSTextSetSelection	Designates the specified text as the current primary selection by highlighting it in the compound string text widget.

Routine	Description
DXmCSTextSetString	Changes the text in the compound string text widget to a new value.
DXmCSTextSetTopPosition	Sets the logical position of the first character in the displayed text.
DXmCSTextShowPosition	Displays the text located at a specified position.
DXmCSTextVerticalScroll	Scrolls text vertically.
DXmCSTextXYToPos	Identifies the position in the text of the character nearest to a specified x and y position.

Geometry Management

The compound string text widget does not support children.

Resizing

The compound string text widget does not support children.

DXmCreateCursor

Creates a specific cursor.

Format

```
Cursor DXmCreateCursor(widget, cursorkind)
        Widget  widget;
        int     cursorkind;
```

Return Value

The identifier (cursor ID) of the created cursor.

Arguments

widget

The widget from which display information is obtained.

cursorkind

An integer that represents the type of cursor, as follows:

UNIX Value	OpenVMS Value	Cursor Description
decw$c_wait_cursor (4)	DXm_WAIT_CURSOR	Watch
decw$c_questionmark_cursor (54)	DXm_HELP_CURSOR	Question mark
decw$c_inactive_cursor (6)	DXm_INACTIVE	CURSOR

Description

The DXmCreateCursor routine enables you to create a cursor that displays as a watch, question mark, or caution (‘‘do not enter’’) symbol (these are the most frequently used cursor symbols). The following X Window System fallbacks are provided in the event the DECwindows-specific cursors cannot be loaded:

DECwindows UNIX Cursor	DECwindows OpenVMS Cursor	X Window System Fallback
decw$c_wait_cursor	DXm_WAIT_CURSOR	XC_watch
decw$c_questionmark_cursor	DXm_HELP_CURSOR	XC_question_ arrow
decw$c_inactive_cursor	DXm_INACTIVE_CURSOR	XC_circle

You can specify additional cursor symbols by examining the XM:DECSPECIFIC.H file on OpenVMS systems and decwcursor.h on both UNIX and Windows NT systems. (X Window System fallbacks are provided for each of those cursors as well.)

DXmCreateHelpDialog

Creates the Help widget with a dialog box.

Widget Class Hierarchy

Core Resource Set
\
Composite Resource Set
\
Constraint Resource Set
\
XmManager Resource Set
\
DXmHelpDialog Resource Set

Format

```
Widget DXmCreateHelpDialog (parent, name, arglist, argcount)
        Widget   parent;
        char     *name;
        Arglist  arglist;
        int      argcount;
```

Resource Information

You can set the following widget-specific resources in the **arglist**:

```
XmString        DXmNaddtopicLabel;
XmString        DXmNapplicationName;
XmString        DXmNbadFrameMessage;
XmString        DXmNbadlibMessage;
Boolean         DXmNcacheHelpLibrary;
XmString        DXmNcloseLabel;
int             DXmNcols;
XmString        DXmNcopyLabel;
KeySym          DXmNcopyLabelMnem;
XmString        DXmNcopyLabelMnemCS;
Boolean         DXmNdefaultPosition;
XmString        DXmNdismissLabel;
XmString        DXmNeditLabel;
KeySym          DXmNeditLabelMnem;
XmString        DXmNeditLabelMnemCS;
XmString        DXmNerroropenMessage;
XmString        DXmNexitLabel;
KeySym          DXmNexitLabelMnem;
XmString        DXmNexitLabelMnemCS;
XmString        DXmNfileLabel;
KeySym          DXmNfileLabelMnem;
XmString        DXmNfileLabelMnemCS;
XmString        DXmNfirstTopic;
XmString        DXmNglossaryTopic;
XmString        DXmNgobackLabel;
KeySym          DXmNgobackLabelMnem;
XmString        DXmNgobackLabelMnemCS;
XmString        DXmNgobacktopicLabel;
XmString        DXmNgooverLabel;
KeySym          DXmNgooverLabelMnem;
XmString        DXmNgooverLabelMnemCS;
XmString        DXmNgotoLabel;
XmString        DXmNgototopicLabel;
KeySym          DXmNgototopicLabelMnem;
XmString        DXmNgototopicLabelMnemCS;
XmString        DXmNhelpAcknowledgeLabel;
XmString        DXmNhelphelpLabel;
KeySym          DXmNhelphelpLabelMnem;
XmString        DXmNhelphelpLabelMnemCS;
XmString        DXmNhelpLabel;
KeySym          DXmNhelpLabelMnem;
XmString        DXmNhelpLabelMnemCS;
XmString        DXmNhelpOnHelpTitle;
XmString        DXmNhelpontitleLabel;
XmString        DXmNhelptitleLabel;
XmString        DXmNhistoryboxLabel;
XmString        DXmNhistoryLabel;
KeySym          DXmNhistoryLabelMnem;
```

```
XmString       DXmNhistoryLabelMnemCS;
XmString       DXmNkeywordLabel;
KeySym         DXmNkeywordLabelMnem;
XmString       DXmNkeywordLabelMnemCS;
XmString       DXmNkeywordsLabel;
XmString       DXmNlibrarySpec;
unsigned int   DXmNlibraryType;
XmString       DXmNnokeywordMessage;
XmString       DXmNnotitleMessage;
XmString       DXmNnulllibMessage;
XmString       DXmNoverviewTopic;
int            DXmNrows;
XmString       DXmNsaveasLabel;
KeySym         DXmNsaveasLabelMnem;
XmString       DXmNsaveasLabelMnemCS;
XmString       DXmNsearchapplyLabel;
XmString       DXmNsearchkeywordboxLabel;
XmString       DXmNsearchLabel;
KeySym         DXmNsearchLabelMnem;
XmString       DXmNsearchLabelMnemCS;
XmString       DXmNsearchtitleboxLabel;
XmString       DXmNselectallLabel;
KeySym         DXmNselectallLabelMnem;
XmString       DXmNselectallLabelMnemCS;
XmString       DXmNtitleLabel;
KeySym         DXmNtitleLabelMnem;
XmString       DXmNtitleLabelMnemCS;
XmString       DXmNtitlesLabel;
XmString       DXmNtopictitlesLabel;
XmString       DXmNviewLabel;
KeySym         DXmNviewLabelMnem;
XmString       DXmNviewLabelMnemCS;
XmString       DXmNvisitglosLabel;
KeySym         DXmNvisitglosLabelMnem;
XmString       DXmNvisitglosLabelMnemCS;
XmString       DXmNvisitLabel;
XmString       DXmNvisittopicLabel;
KeySym         DXmNvisittopicLabelMnem;
XmString       DXmNvisittopicLabelMnemCS;
XtCallbackList XmNmapCallback;
XtCallbackList XmNunmapCallback;
```

Return Value

The identifier (widget ID) of the created Help widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

arglist
The application argument list.

argcount
An integer that represents the number of arguments in the application argument list.

Resources

DXmNaddtopicLabel
The label indicating additional topics for help. The default is ‘‘Additional topics’’. Access is CSG.

DXmNapplicationName
The application name to be used in the widget title bar. The default is Null. Access is CSG.

DXmNbadFrameMessage
The message displayed when a frame could not be found. The default is ‘‘Couldn’t find frame !CS’’. Access is CSG.

DXmNbadlibMessage
The message displayed when a requested library could not be found. The default is ‘‘Couldn’t open library !CS’’. Access is CSG.

DXmNcacheHelpLibrary
A Boolean resource that specifies whether the text of the help library is stored in the help widget’s cache memory. When True, the text is stored in cache memory; when False, the text is not stored in cache memory. The default is False. Access is CSG.

DXmNcloseLabel
The label for the Exit push button in the help widget window. The default is ‘‘Exit’’. Access is CSG.

DXmNcols
The width, in characters, of the help text window. The default is language dependent; the American English default is 55. Access is CSG.

DXmNcopyLabel
The label for the Copy menu item in the Edit menu. The default is ‘‘Copy’’. Access is CSG.

DXmNcopyLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Copy menu item in the Edit menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Copy menu label. The default is an underscore (_) inserted beneath the letter ‘‘C’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNcopyLabelMnemCS
Establishes the character set used when identifying the letter in the Copy menu label that corresponds to the key the user can press (as defined by the DXmNcopyLabelMnem resource) to activate the Copy menu item. The default character set is ISO8859-1. Access is CG.

DXmNdefaultPosition
A Boolean resource that, when True, causes the core resources **x** and **y** to be ignored in favor of the default widget position. The default widget position is centered in the parent window. When False, the specified **x** and **y** resources are used to position the widget. The default is True. Access is CSG.

DXmNdismissLabel
The label for the push button used to dismiss a help widget dialog box (for example, Search History, Search Title, Search Keyword boxes). The default is ‘‘Dismiss’’. Access is CSG.

DXmNeditLabel
The label for the Edit pull-down menu. The default is ‘‘Edit’’. Access is CSG.

DXmNeditLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Edit pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Edit pull-down menu label. The default is an underscore (_) inserted beneath the letter ‘‘E’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNeditLabelMnemCS
Establishes the character set used when identifying the letter in the Edit pull-down menu label that corresponds to the key the user can press (as defined by the DXmNeditLabelMnem resource) to activate the Edit pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNerroropenMessage
The error message displayed when a file cannot be opened. The default is ''Error opening file !CS''. Access is CSG.

DXmNexitLabel
The label for the exit menu item in the File pull-down menu. The default is ''Exit''. Access is CSG.

DXmNexitLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Exit menu item in the File pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Exit menu label. The default is an underscore (_) inserted beneath the letter ''E'', with the corresponding key defined on the keyboard. Access is CG.

DXmNexitLabelMnemCS
Establishes the character set used when identifying the letter in the Exit menu label that corresponds to the key the user can press (as defined by the DXmNexitLabelMnem resource) to activate the Exit menu item in the File pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNfileLabel
The label for the File pull-down menu. The default is ''File''. Access is CSG.

DXmNfileLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the File pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the File pull-down menu label. The default is an underscore (_) inserted beneath the letter ''F'', with the corresponding key defined on the keyboard. Access is CG.

DXmNfileLabelMnemCS
Establishes the character set used when identifying the letter in the File pull-down menu label that corresponds to the key the user can press (as defined by the DXmNfileLabelMnem resource) to activate the File pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNfirstTopic
The first help topic to be displayed. If a null string is passed, a list of level one topics is displayed. The default is Null. Access is CSG.

DXmNglossaryTopic
The application glossary topic. If a null string is provided, the Visit Glossary menu item does not appear in the View pull-down menu. The default is Null. Access is CSG.

DXmNgobackLabel
The label for the Go Back menu item in the View pull-down menu. The default is ''Go Back''. Access is CSG.

DXmNgobackLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Go Back menu item in the View pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Go Back menu label. The default is an underscore (_) inserted beneath the letter ''B'', with the corresponding key defined on the keyboard. Access is CG.

DXmNgobackLabelMnemCS
Establishes the character set used when identifying the letter in the Go Back menu label that corresponds to the key the user can press (as defined by the DXmNgobackLabelMnem resource) to activate the Go Back menu item in the View pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNgobacktopicLabel
The label for the Go Back push button in the help widget window. The default is ''Go Back''. Access is CSG.

DXmNgooverLabel
The label for the Go To Overview item in the View pull-down menu. The default is ''Go To Overview''. Access is CSG.

DXmNgooverLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Go To Overview item in the View pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Go To Overview menu label. The default is an underscore (_) inserted beneath the letter ''O'', with the corresponding key defined on the keyboard. Access is CG.

DXmNgooverLabelMnemCS
Establishes the character set used when identifying the letter in the Go To Overview menu label that corresponds to the key the user can press (as defined by the DXmNgooverLabelMnem resource) to activate the Go To Overview item in the View pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNgotoLabel
The label for the Go To push button in the help widget's dialog boxes. The default is ''Go To''. Access is CSG.

DXmNgototopicLabel
The label for the Go To Topic menu item in the View pull-down menu. The default is ''Go To Topic''. Access is CSG.

DXmNgototopicLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Go To Topic menu item in the View pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Go To Topic menu label. The default is an underscore (_) inserted beneath the letter ''T'', with the corresponding key defined on the keyboard. Access is CG.

DXmNgototopicLabelMnemCS
Establishes the character set used when identifying the letter in the Go To Topic menu label that corresponds to the key the user can press (as defined by the DXmNgototopicLabelMnem resource) to activate the Go To Topic menu item in the View pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNhelpAcknowledgeLabel
The label for the Acknowledge push button in the error message box. The default is ''OK''. Access is CSG.

DXmNhelphelpLabel
The label for the Overview menu item in the Using Help pull-down menu. The default is ''On Window...''. Access is CSG.

DXmNhelphelpLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Overview menu item in the Using Help pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Overview

menu label. The default is an underscore (_) inserted beneath the letter ''W'', with the corresponding key defined on the keyboard. Access is CG.

DXmNhelphelpLabelMnemCS
Establishes the character set used when identifying the letter in the Overview menu label that corresponds to the key the user can press (as defined by the DXmNhelphelpLabelMnem resource) to activate the Overview menu item in the Using Help pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNhelpLabel
The label for the Help pull-down menu. The default is ''Using Help''. Access is CSG.

DXmNhelpLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Help pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Help pull-down menu label. The default is an underscore (_) inserted beneath the letter ''U'', with the corresponding key defined on the keyboard. Access is CG.

DXmNhelpLabelMnemCS
Establishes the character set used when identifying the letter in the Help pull-down menu label that corresponds to the key the user can press (as defined by the DXmNhelpLabelMnem resource) to activate the Help pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNhelpOnHelpTitle
The label for the title bar in the Help-on-Help help subwidget. The default is ''Using Help''. Access is CSG.

DXmNhelpontitleLabel
The label for the help widget title bar used in conjunction with the application name. The default is ''Help On''. Access is CSG.

DXmNhelptitleLabel
The label for the help widget title bar when no application name is specified. The default is ''Help''. Access is CSG.

DXmNhistoryboxLabel
The label for the history dialog box. The default is ''Search Topic History''. Access is CSG.

DXmNhistoryLabel
The label for the History... menu item in the Search pull-down menu. The default is ''History...''. Access is CSG.

DXmNhistoryLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the History... menu item in the Search pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the History... menu label. The default is an underscore (_) inserted beneath the letter ''H'', with the corresponding key defined on the keyboard. Access is CG.

DXmNhistoryLabelMnemCS
Establishes the character set used when identifying the letter in the History... menu label that corresponds to the key the user can press (as defined by the DXmNhistoryLabelMnem resource) to activate the History... menu item in the Search pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNkeywordLabel
The label for the Keyword... menu item in the Search pull-down menu. The default is ''Keyword...''. Access is CSG.

DXmNkeywordLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Keyword... menu item in the Search pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Keyword... menu label. The default is an underscore (_) inserted beneath the letter ''K'', with the corresponding key defined on the keyboard. Access is CG.

DXmNkeywordLabelMnemCS
Establishes the character set used when identifying the letter in the Keyword... menu label that corresponds to the key the user can press (as defined by the DXmNkeywordLabelMnem resource) to activate the Keyword... menu item in the Search pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNkeywordsLabel
The label used in a Search Topic Keyword box to identify the text entry field. The default is ''Keyword''. Access is CSG.

DXmNlibrarySpec
A host system file specification that identifies the help topic library. The default is Null. Access is CG.

DXmNlibraryType
The type of help topic library specified by the DXmNlibrarySpec resource. The predefined value for this resource is DXmTextLibrary, which is help text in either an OpenVMS help library or a help directory on UNIX or Windows NT systems. Access is CG.

DXmNnokeywordMessage
The message displayed when a requested keyword cannot be found. The default is ‘‘Couldn’t find keyword !CS’’. Access is CSG.

DXmNnotitleMessage
The message displayed when a requested title cannot be found. The default is ‘‘No title to match string !CS’’. Access is CSG.

DXmNnulllibMessage
The text for the message displayed when no library has been specified. The default is ‘‘No library specified’’. Access is CSG.

DXmNoverviewTopic
The application overview topic. The default is Null. Access is CSG.

DXmNrows
Height, in characters, of the help text window. The default is language dependent; the American English default is 20. Access is CSG.

DXmNsaveasLabel
The label for the Save As... item in a File pull-down menu. The default is ‘‘Save As...’’. Access is CSG.

DXmNsaveasLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Save As... item in a File pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Save As... menu label. The default is an underscore (_) inserted beneath the letter ‘‘A’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNsaveasLabelMnemCS
Establishes the character set used when identifying the letter in the Save As... menu label that corresponds to the key the user can press (as defined by the DXmNsaveasLabelMnem resource) to activate the Save As... item in a File pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNsearchapplyLabel
The label for the push button used to initiate a search action in a Search dialog box. The default is ‘‘Apply’’. Access is CSG.

DXmNsearchkeywordboxLabel
The label for a Search Topic Keywords dialog box. The default is ‘‘Search Topic Keywords’’. Access is CSG.

DXmNsearchLabel
The label for the Search pull-down menu. The default is ‘‘Search’’. Access is CSG.

DXmNsearchLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Search pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Search pull-down menu label. The default is an underscore (_) inserted beneath the letter ‘‘S’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNsearchLabelMnemCS
Establishes the character set used when identifying the letter in the Search pull-down menu label that corresponds to the key the user can press (as defined by the DXmNsearchLabelMnem resource) to activate the Search pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNsearchtitleboxLabel
The title of a Search Topic Titles box. The default is ‘‘Search Topic Titles’’. Access is CSG.

DXmNselectallLabel
The label for the Select All item on the Edit pull-down menu. The default is ‘‘Select All’’. Access is CSG.

DXmNselectallLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Select All item on the Edit pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Select All menu label. The default is an underscore (_) inserted beneath the letter ‘‘S’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNselectallLabelMnemCS
Establishes the character set used when identifying the letter in the Select All menu label that corresponds to the key the user can press (as defined by the DXmNselectLabelMnem resource) to activate the Select All item on the Edit pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNtitleLabel
The label for the Title... item in the Search pull-down menu. The default is ‘‘Title...’’. Access is CSG.

DXmNtitleLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Title... item in the Search pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Title... menu label. The default is an underscore (_) inserted beneath the letter ‘‘T’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNtitleLabelMnemCS
Establishes the character set used when identifying the letter in the Title... menu label that corresponds to the key the user can press (as defined by the DXmNtitleLabelMnem resource) to activate the Title... item in the Search pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNtitlesLabel
The label identifying the text entry field on the Search Topic Titles box. The default is ‘‘Title’’. Access is CSG.

DXmNtopictitlesLabel
The label used to identify topics found as a result of a title search in a Search Topic Titles box. The default is ‘‘Topic Titles’’. Access is CSG.

DXmNviewLabel
The label for the View menu. The default is ‘‘View’’. Access is CSG.

DXmNviewLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the View menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the View menu label. The default is an underscore (_) inserted beneath the letter ‘‘V’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNviewLabelMnemCS
Establishes the character set used when identifying the letter in the View menu label that corresponds to the key the user can press (as defined by the DXmNviewLabelMnem resource) to activate the View menu. The default character set is ISO8859-1. Access is CG.

DXmNvisitglosLabel
The label for the Visit Glossary item in the View pull-down menu. The default is ‘‘Visit Glossary’’. Access is CSG.

DXmNvisitglosLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Visit Glossary item in the View pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Visit Glossary menu label. The default is an underscore (_) inserted beneath the letter ‘‘G’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNvisitglosLabelMnemCS
Establishes the character set used when identifying the letter in the Visit Glossary menu label that corresponds to the key the user can press (as defined by the DXmNvisitglosLabelMnem resource) to activate the Visit Glossary item in the View pull-down menu. The default character set is ISO8859-1. Access is CG.

DXmNvisitLabel
The label for the Visit push button in the help widget’s dialog boxes. The default is ‘‘Visit’’. Access is CSG.

DXmNvisittopicLabel
The label for the Visit Topic menu item in the View pull-down menu. The default is ‘‘Visit Topic’’. Access is CSG.

DXmNvisittopicLabelMnem
Specifies which key the user can press (instead of clicking MB1) to activate the Visit Topic menu item in the View pull-down menu. This resource defines the key on the keyboard and visually identifies that key for the user by inserting an underscore (_) beneath the designated letter in the Visit Topic menu label. The default is an underscore (_) inserted beneath the letter ‘‘V’’, with the corresponding key defined on the keyboard. Access is CG.

DXmNvisittopicLabelMnemCS

Establishes the character set used when identifying the letter in the Visit Topic menu label that corresponds to the key the user can press (as defined by the DXmNvisittopicLabelMnem resource) to activate the Visit Topic menu item in the View pull-down menu. The default character set is ISO8859-1. Access is CSG.

XmNmapCallback

Callback (or callbacks) made when the help widget is mapped. The callback reason is **XmCR_MAP**. The default is Null. Access is C.

XmNunmapCallback

Callback (or callbacks) made when the help widget is unmapped. The callback reason is **XmCR_UNMAP**. The default is Null. Access is C.

Resource Exceptions

The resources **XmNwidth** and **XmNheight** are supported differently by DXmCreateHelpDialog in that they cannot be set by the caller. These values are calculated by the widget, based on the size of the text window (**XmNcols** and **XmNrows**).

Callback Structure

```
typedef struct
{
      int     reason;
      XEvent  *event;
} XmAnyCallbackStruct;
```

Callback Field Descriptions

reason

An integer set to the callback reason. See the Callback Reasons section for the values that are valid for this widget.

event

A pointer to the X event structure describing the event that generated this callback.

Callback Reasons

XmCR_MAP
The help window was just mapped.

XmCR_UNMAP
The help window was just unmapped.

Description

The DXmCreateHelpDialog routine creates a help widget that includes its own dialog box. The help widget is a modeless widget that enables the application to display appropriate user assistance information in response to a user request. When the user requests help, the help widget displays an initial help topic, then gives the user the ability to view additional help topics.

The **DXmNfirstTopic** resource allows the application to provide context-sensitive help by selecting a specific topic based on implicit or explicit cues from the user.

After the widget has been created, you can change the help topic by specifying a new **DXmNfirstTopic** (using the Intrinsic routine XtSetValues), and then managing the widget (using the Intrinsic routine XtManageChild) to display the help window. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

When the user exits from a help session, the widget is automatically unmanaged.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the help widget.

Geometry Management

The help widget does not support children.

Resizing

The help widget sizes itself at creation, based on **XmNrows** and **XmNcols**.

DXmCreatePrintBox

Creates the print widget without a dialog box.

Widget Class Hierarchy

Core Resource Set
\
Composite Resource Set
\
Constraint Resource Set
\
XmManager Resource Set
\
XmBulletinBoard Resource Set
\
DXmPrintBox Resource Set

Format

```
Widget DXmCreatePrintBox(parent, name, arglist, argcount)
        Widget   parent;
        char     *name;
        Arglist  arglist;
        int      argcount;
```

Return Value

The identifier (widget ID) of the created print widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

arglist
The application argument list.

argcount
An integer that represents the number of arguments in the application argument list.

Description

The DXmCreatePrintBox routine uses the same arguments and resources as the DXmCreatePrintDialog routine to create a print widget, but does not include a dialog box. Refer to the DXmCreatePrintDialog routine for a complete description.

DXmCreatePrintDialog

Creates the print widget with a dialog box.

Widget Class Hierarchy

Core Resource Set
\
Composite Resource Set
\
Constraint Resource Set
\
XmManager Resource Set
\
XmBulletinBoard Resource Set
\
DXmPrintDialog Resource Set

Format

```
Widget DXmCreatePrintDialog(parent, name, arglist, argcount)
        Widget   parent;
        char     *name;
        Arglist  arglist;
        int      argcount;
```

Resource Information

You can set the following widget-specific resources in the **arglist**:

```
Boolean             DXmNautoPagination;
XmString            DXmNdefaultPrinter;
Boolean             DXmNdeleteFile;
Boolean             DXmNdoubleSpacing;
int                 DXmNfileBurstSheet;
int                 DXmNfileEndSheet;
int                 DXmNfileNameCount;
XmStringTable       DXmNfileNameList;
int                 DXmNfileStartSheet;
Boolean             DXmNheader;
Boolean             DXmNholdJob;
int                 DXmNinputTray;
XmString            DXmNjobName;
XmString            DXmNlayupDefinition;
int                 DXmNmessageLog;
Boolean             DXmNnotify;
int                 DXmNnumberCopies;
int                 DXmNnumberUp;
XmString            DXmNoperatorMessage;
XmString            DXmNoptionsDialogTitle;
int                 DXmNorientation;
int                 DXmNoutputTray;
XmString            DXmNpageRangeFrom;
XmString            DXmNpageRangeTo;
int                 DXmNpageSize;
Boolean             DXmNpassAll;
XmString            DXmNprintAfter;
int                 DXmNprinterCount;
XmString            DXmNprinterChoice;
XmString            DXmNprinterFormChoice;
XmStringTable       DXmNprinterFormList;
int                 DXmNprinterFormCount;
XmStringTable       DXmNprinterList;
XmString            DXmNprintFormatChoice;
int                 DXmNprintFormatCount;
XmStringTable       DXmNprintFormatList;
int                 DXmNpriority;
XmString            DXmNsetup;
int                 DXmNsheetCount;
int                 DXmNsheetSize;
int                 DXmNsides;
XmString            DXmNstartSheetComment;
unsigned long       DXmNsuppressOptionsMask;
Boolean             DXmNunmanageOnCancel;
Boolean             DXmNunmanageOnOk;
XtCallbackList      XmNcancelCallback;
XtCallbackList      XmNokCallback;
```

Return Value

The identifier (widget ID) of the created print widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

arglist
The application argument list.

argcount
An integer that represents the number of arguments in the application argument list.

Resources

DXmNautoPagination
A Boolean value that, when True, indicates that autopagination (formatting the output into pages based on length or line count) is turned on. When False, autopagination is turned off. By default, the value of this resource changes when the user switches print formats. The default settings for the print formats are as follows:

Print Format	Default
Default[1]	False
Text (ANSI Level 0)	True
Line Printer (ANSI Level 1)	True
Terminal	True
ANSI2 (ANSI Level 2)	False
ANSI (ANSI Level 3)	False
PostScript	False
ReGIS	False

[1]No data type specification is output.

Print Format	Default
TEK (Tektronix)	False
DDIF	False

Access is CSG.

DXmNdefaultPrinter
An XmString that allows you to specify a default printer. If you do not specify a default printer, SYS$PRINT is the default choice on OpenVMS systems. The PRINTER environment variable determines the default on UNIX systems. For Windows NT systems, LPT1: is the default. Access is CSG.

DXmNdeleteFile
A Boolean value that controls whether the file is deleted after it is printed. When True, the file is deleted. When False, the file is not deleted. The default is False. The user must have delete access to the file to delete it. Access is CSG.

DXmNdoubleSpacing
A Boolean value that controls the state of double-spacing. When True, double-spacing is turned on. When False, double-spacing is turned off. The default is False. Access is CSG.

DXmNfileBurstSheet
An integer value that allows you to control whether there will be a File Burst Sheet for your print job and how frequently it appears. The possible values are as follows.

Value	Description
DXmFILE_SHEET_DEFAULT	Indicates that you want the default for the queue on which the job is being printed.
DXmFILE_SHEET_NONE	Indicates that you do not want a burst page.
DXmFILE_SHEET_ONE	Indicates that you want only a single burst page that will precede the first file printed.
DXmFILE_SHEET_ALL	Indicates that you want the burst page to be printed before every file in the print job.

The default is DXmFILE_SHEET_DEFAULT, or 0. Access is CSG.

DXmNfileEndSheet

An integer value that allows you to control whether there will be a File End Sheet for your print job and how frequently it is printed. The possible values are as follows:

Value	Description
DXmFILE_SHEET_DEFAULT	Indicates that you want the default for the queue on which the job is being printed.
DXmFILE_SHEET_NONE	Indicates that you do not want a File End Sheet.
DXmFILE_SHEET_ONE	Indicates that you want only a single File End Sheet to follow the last file printed.
DXmFILE_SHEET_ALL	Indicates that you want the File End Sheet to be printed after every file in the print job.

The default is DXmFILE_SHEET_DEFAULT, or 0. Access is CSG.

DXmNfileNameCount

The number of files specified in the DXmNfileNameList resource, expressed as an integer value. The default is 0. Access is CSG.

DXmNfileNameList

An XmStringTable that enables the print widget to use the files specified by this resource to make assumptions about the print format of the files to be printed. If you set the DXmNfileNameList resource, you must also set the value of the DXmNfileNameCount resource. The default is Null. Access is CSG.

DXmNfileStartSheet

An integer value that allows you to control whether there will be a File Start Sheet for your print job and how frequently it is printed. The possible values are as follows:

Value	Description
DXmFILE_SHEET_DEFAULT	Indicates that you want the default for the queue on which the job is being printed.
DXmFILE_SHEET_NONE	Indicates that you do not want a File Start Sheet.

Value	Description
DXmFILE_SHEET_ONE	Indicates that you want only a single File Start Sheet that will precede the first file.
DXmFILE_SHEET_ALL	Indicates that you want the File Start Sheet to be printed before every file in the print job.

The default is DXmFILE_SHEET_DEFAULT, or 0. Access is CSG.

DXmNheader
A Boolean value that controls whether to print the header. When True, the header is printed. When False, the header is not printed. The default is False. Access is CSG.

DXmNholdJob
A Boolean value that controls whether the print job will finish without further manual intervention. If this resource is True, the print job is held in the print queue servicing the printer until it is manually released by queue management operations. The default is False. Access is CSG.

DXmNinputTray
An integer value that allows you to specify the input tray holding the medium to be used for the print job. The possible values for DXmNinputTray are as follows:

Value	Description
DXmINPUT_TRAY_DEFAULT	The default input tray is that which is specified for the queue.
DXmINPUT_TRAY_TOP	The top input tray is the source of the medium on which to print.
DXmINPUT_TRAY_MIDDLE	The middle input tray is the source of the medium on which to print.
DXmINPUT_TRAY_BOTTOM	The bottom input tray is the source of the medium on which to print.

The default is DXmINPUT_TRAY_DEFAULT, or 0. Access is CSG.

DXmNjobName
A pointer to an XmString that identifies the print job name. If no job name is specified, the job name is derived from the name of the first file being printed. The default is Null. Access is CSG.

DXmNlayupDefinition

On OpenVMS systems, an XmString that allows you to enter a module name contained in the LPS$LAYUP:x.LUP library. This module, in conjunction with the DXmNnumberUp resource, controls how the image is deposited on the medium. The default is Null. Access is CSG.

DXmNmessageLog

An integer value that allows you to specify what is to be done with error messages that are created in the course of printing. The possible values for DXmNmessageLog are as follows:

Value	Description
DXmMESSAGE_LOG_DEFAULT	Accept whatever is specified for the print queue.
DXmMESSAGE_LOG_IGNORE	Nothing is done with the error messages.
DXmMESSAGE_LOG_PRINT	The error messages are written to the File End Sheet.
DXmMESSAGE_LOG_KEEP	The error messages are retained in a file.
DXmMESSAGE_LOG_KEEP_AND_PRINT	The error messages are written to the File End Sheet (or additional job error sheet) and are retained in a log file.

The default is DXmMESSAGE_LOG_DEFAULT, or 0. Access is CSG.

DXmNnotify

A Boolean value that determines whether the user is notified when the file is printed. When True, the user is notified. When False, the user is not notified. The default is True. Access is CSG.

DXmNnumberCopies

An integer value that determines the number of copies of the file to print. The allowable range is from 1 to 255 copies. The default is 0 (which means to use the system default). Access is CSG.

DXmNnumberUp

An integer value in the range of 0–100 that specifies how many logical images are to be printed on one side of a page. The default is 0 (one logical image, without any adjustments to how the image is deposited on the medium). Access is CSG.

DXmNoperatorMessage

An XmString that allows you to enter a text field to be displayed on the operator's console when the job first starts printing. DXmNoperatorMessage can be used to pass messages to the operator concerning the handling of the printed output. The default is Null. Access is CSG.

DXmNoptionsDialogTitle

Specifies the title for the secondary dialog box that lists additional print options. The default is ''Print: Options''. Access is CSG.

DXmNorientation

An integer value that determines the paper orientation (on printers that support this feature). The possible values for orientation are as follows:

Value	Description
DXmORIENTATION_DEFAULT	Allows you to accept the orientation that is defined for the print queue you have selected.
DXmORIENTATION_PORTRAIT	Allows you to force a portrait orientation to the resulting printout.
DXmORIENTATION_LANDSCAPE	Allows you to force a landscape orientation to the resulting printout.

The default is DXmORIENTATION_DEFAULT, or 0. Access is CSG.

DXmNoutputTray

An integer value that allows you to specify the output tray in which the medium is to be placed after the sheet is printed. The possible values for DXmNoutputTray are as follows:

Value	Description
DXmOUTPUT_TRAY_DEFAULT	The default output tray is that which is specified for the queue.
DXmOUTPUT_TRAY_TOP	Allows you to select the top output tray to receive the printed medium.
DXmOUTPUT_TRAY_SIDE	Allows you to select the side output tray to receive the printed medium.

Value	Description
DXmOUTPUT_TRAY_FACE_UP	Allows you to select the faceup output tray to receive the printed medium.
DXmOUTPUT_TRAY_UPPER	Allows you to select the upper output tray to receive the printed medium.
DXmOUTPUT_TRAY_LOWER	Allows you to select the lower output tray to receive the printed medium.

The default is DXmOUTPUT_TRAY_DEFAULT, or 0. Access is CSG.

DXmNpageRangeFrom

A pointer to an XmString that identifies the starting page of the file to print. The DXmNpageRangeFrom and DXmNpageRangeTo resources allow you to enter a range of pages to be printed if you chose not to print a file in its entirety. When using this resource, note the following:

- If you want to start printing at the first page of the file, you may leave the DXmNpageRangeFrom resource blank (no value specified).
- If you specify a value for the DXmNpageRangeTo resource, the DXmNpageRangeFrom resource must be less than or equal to that value.

The default is Null. Access is CSG.

DXmNpageRangeTo

A pointer to an XmString that identifies the last page of the file to print. The DXmNpageRangeFrom and DXmNpageRangeTo resources allow you to enter a range of pages to be printed if you chose not to print a file in its entirety.

The DXmNpageRangeTo resource may be left blank to print to the last page of the file. The default is Null. Access is CSG.

DXmNpageSize

An integer value that determines the logical (or virtual) size of the print image. The possible values for page size are as follows:

Value	Size in Inches	Size in MM
DXmSIZE_DEFAULT	Queue Default	Queue Default
DXmSIZE_LETTER	8.5 x 11	216 x 279
DXmSIZE_LEDGER	11 x 17	279 x 432

Value	Size in Inches	Size in MM
DXmSIZE_LEGAL	8.5 x 14	216 x 356
DXmSIZE_EXECUTIVE	7.5 x 10	191 x 254
DXmSIZE_A5	5.8 x 8.3	148 x 210
DXmSIZE_A4	8.3 x 11.7	210 x 297
DXmSIZE_A3	11.7 x 16.5	297 x 420
DXmSIZE_B5	7.2 x 10.1	176 x 250
DXmSIZE_B4	10.1 x 14.3	250 x 353
DXmSIZE_C4_ENVELOPE	9 x 12.8	229 x 324
DXmSIZE_C5_ENVELOPE	6.4 x 9	162 x 229
DXmSIZE_C56_ENVELOPE	4.3 x 8.7	110 x 220
DXmSIZE_10X13_ENVELOPE	10 x 13	254 x 330.2
DXmSIZE_9X12_ENVELOPE	9 x 12	228.6 x 304.8
DXmSIZE_BUSINESS_ENVELOPE	4.125 x 9.5	104.8 x 241.3

The default is DXmSIZE_DEFAULT. Access is CSG.

DXmNpassAll

A Boolean value that, when True, allows all control characters to be passed to the printer. By default, the value of this resource changes when the user switches print formats. The default settings for the print formats are as follows:

Print Format	Default
Default[1]	False
Text (ANSI Level 0)	False
Line Printer (ANSI Level 1)	False
Terminal	False
ANSI2 (ANSI Level 2)	True
ANSI (ANSI Level 3)	True
PostScript	False
ReGIS	False
TEK (Tektronix)	True

[1]No data type specification is output.

Print Format	Default
DDIF	True

Access is CSG.

DXmNprintAfter

An XmString that allows you to enter a time and date when this print job first becomes eligible to be scheduled for printing. The XmString for the time and date must be the standard OpenVMS format DD-MMM-YYYY HH:MM, for example: 23-JUL-1991 17:30. For UNIX systems, see the at(1) reference page for information about date formats. The format for Windows NT is MM/DD/YY HH:MM[a|p], where a is for AM and p is for PM.

If this resource is not specified, the job becomes eligible for immediate print scheduling. The job also becomes eligible for immediate print scheduling if the user specifies this resource but specifies either nothing or ''now'' as the XmString. The default is ''now''. Access is CSG.

DXmNprinterChoice

An XmString that specifies the printer to use for the current print format. Applications usually use this resource to store information about a user's printer preference.

If you set this resource, the XmString must exactly match one of the print widget's supported printers, including the character set. The recommended approach is to not specify this resource when the print widget is first managed. Then, once the user has selected a print format, call the XtGetValues routine to obtain the value and then XtSetArg to set DXmNprinterChoice.

The print widget sets this resource as follows:

- If the user selects a new print format, the printer queue box might be repopulated (assuming the user has set up the print queue logicals).
- If the user has chosen this print format before and has already chosen a printer for this format, the print widget identifies this printer as the DXmNprinterChoice resource. The printer is automatically selected, regardless of the previous queue choice.
- If the user has not previously selected this print format, the print widget determines whether there is a printer that matches the one defined by DXmNdefaultPrinter. If that printer is available, the print widget selects it; otherwise the print widget selects the first printer in the list.

The default is Null. Access is CSG.

DXmNprinterCount

The number of printers specified in the DXmNprinterList resource, expressed as an integer. If your application requires the value of this resource, call the XtGetValues routine to obtain the value.

You cannot set a value for this resource. (Access is G.) The default is 0.

DXmNprinterFormChoice

An XmString that specifies the printer form to use as the default. Use this resource when you want your application to record a user's printer preference.

Note that if you set this resource, the XmString must exactly match one of the operating system's supported printer forms. (Use the DXmCvtOSToCS routine to convert the string to an XmString.) Digital recommends that you do not specify this resource when the print widget is first managed. Instead, after the user has selected a printer form, call the Intrinsic routines XtGetValues to obtain the value and XtSetArg to set DXmNprinterFormChoice. The default is Null. Access is CSG.

DXmNprinterFormList

An XmStringTable list of the available printer forms. If your application requires the value of this resource, call the XtGetValues routine to obtain the value.

You cannot set a value for this resource. (Access is G.) The default is Null.

DXmNprinterFormCount

The number of printer forms specified in the DXmNprinterFormList resource, expressed as an integer value. If your application requires the value of this resource, call the XtGetValues routine to obtain the value.

You cannot set a value for this resource. (Access is G.) The default is 0.

DXmNprinterList

An XmStringTable that determines the names to appear in the Printer list box. If your application requires the value of this resource, call the XtGetValues routine to obtain the value.

You cannot set a value for this resource. (Access is G.) The default is Null.

DXmNprintFormatChoice

An XmString that specifies the print format to use as the default. Use this resource when you want your application to record a user's print format preference.

Note that if you set this resource, the XmString must exactly match one of the the print widget's supported print formats, including the character set. Digital recommends that you do not specify this resource when the print widget is first managed. Instead, after the user has selected a print format, call the Intrinsic routines XtGetValues to obtain the value and XtSetArg to set DXmNprintFormatChoice. The default is Null. Access is CSG.

DXmNprintFormatCount

The number of print formats specified in the DXmNprintFormatList resource, expressed as an integer value. The default is 0. Access is CSG.

DXmNprintFormatList

An XmStringTable that determines the print widget's print format list, which is the list of print formats that are understood by the print widget. If you set this resource, each XmString in the table must exactly match one of the print widget's supported print formats.

The print widget supports the following data formats:

- Default (no data type specification is output)
- Text (ANSI Level 0)
- Line Printer (ANSI Level 1)
- Terminal
- ANSI2 (ANSI Level 2)
- ANSI (ANSI Level 3)
- PostScript
- ReGIS
- Tektronix
- DDIF

The default is Null. Access is CSG.

DXmNpriority

An integer value that determines the priority of the job (0–255) in the print queue. If no priority is specified (a value of 0), the default priority for that queue is used. The default is 0. Access is CSG.

DXmNsetup

An XmString that allows you to enter the comma-separated names of one or more setup modules to be transmitted to the printer ahead of the file to be printed. The default is Null. Access is CSG.

DXmNsheetCount

An integer value in the range of 0–10000 that specifies how many times each sheet is to be printed. This option is most frequently used with PostScript printers, which require only a single transmission of information to replicate a complex image (instead of multiple transmissions of the same information). The default is 0. Access is CSG.

DXmNsheetSize

An integer value that determines the size of the medium on which the printing is to take place. The possible values for sheet size are as follows:

Value	Size in Inches	Size in MM
DXmSIZE_DEFAULT	Queue Default	Queue Default
DXmSIZE_LETTER	8.5 x 11	216 x 279
DXmSIZE_LEDGER	11 x 17	279 x 432
DXmSIZE_LEGAL	8.5 x 14	216 x 356
DXmSIZE_EXECUTIVE	7.5 x 10	191 x 254
DXmSIZE_A5	5.8 x 8.3	148 x 210
DXmSIZE_A4	8.3 x 11.7	210 x 297
DXmSIZE_A3	11.7 x 16.5	297 x 420
DXmSIZE_B5	7.2 x 10.1	176 x 250
DXmSIZE_B4	10.1 x 14.3	250 x 353
DXmSIZE_C4_ENVELOPE	9 x 12.8	229 x 324
DXmSIZE_C5_ENVELOPE	6.4 x 9	162 x 229
DXmSIZE_C56_ENVELOPE	4.3 x 8.7	110 x 220
DXmSIZE_10X13_ENVELOPE	10 x 13	254 x 330.2
DXmSIZE_9X12_ENVELOPE	9 x 12	228.6 x 304.8
DXmSIZE_BUSINESS_ENVELOPE	4.125 x 9.5	104.8 x 241.3

The default is DXmSIZE_DEFAULT, or 0. Access is CSG.

DXmNsides

An integer value that determines on how many sides of a medium the printing is to take place and the manner in which the medium is turned when switching from one side to the other.

The possible values for DXmNsides are as follows:

Value	Description
DXmSIDES_DEFAULT	The default for the queue on which you are printing.
DXmSIDES_SIMPLEX_ONE	Prints on one side only. This is also known as one-sided simplex.
DXmSIDES_SIMPLEX_TWO	Prints on two sides of the sheet, but uses the same page layout conventions required for single-sided printing. The layout reflects margin and page number locations and related specifications.
DXmSIDES_DUPLEX_ONE	Prints only on the first side of each sheet, but retains the page layout intended for duplex printing. The layout reflects margin and page number locations and related specifications.
DXmSIDES_DUPLEX_TWO	Prints on two sides; the second side is accessed by flipping the page along its left edge, like turning the pages of a book. This is also known as two-sided duplex.
DXmSIDES_TUMBLE_ONE	Prints only on the first side of each sheet, but retains the page layout intended for tumble printing. The layout reflects margin and page number locations and related specifications.
DXmSIDES_TUMBLE_TWO	Prints on two sides, and the second side is reached by flipping the page along its top edge, like turning the pages of a legal document. This is also known as two-sided tumble.

The default is DXmSIDES_DEFAULT, or 0. Access is CSG.

DXmNstartSheetComment

The XmString to appear on your File Start Sheet. The default is Null. Access is CSG.

DXmNsuppressOptionsMask

Suppresses certain print widget features, such as the number of print choices available to the user. The possible values for this resource are as follows:

- DXmSUPPRESS_AUTOMATIC_PAGINATION
- DXmSUPPRESS_DELETE_FILE

- DXmSUPPRESS_DOUBLE_SPACING
- DXmSUPPRESS_FILE_BURST_SHEET
- DXmSUPPRESS_FILE_END_SHEET
- DXmSUPPRESS_FILE_START_SHEET
- DXmSUPPRESS_HEADER
- DXmSUPPRESS_HOLD_JOB
- DXmSUPPRESS_INPUT_TRAY
- DXmSUPPRESS_JOB_NAME
- DXmSUPPRESS_LAYUP_DEFINITION
- DXmSUPPRESS_MESSAGE_LOG
- DXmSUPPRESS_NONE
- DXmSUPPRESS_NOTIFY
- DXmSUPPRESS_NUMBER_COPIES
- DXmSUPPRESS_NUMBER_UP
- DXmSUPPRESS_OPERATOR_MESSAGE
- DXmSUPPRESS_ORIENTATION
- DXmSUPPRESS_OUTPUT_TRAY
- DXmSUPPRESS_PAGE_RANGE
- DXmSUPPRESS_PAGE_SIZE
- DXmSUPPRESS_PASS_ALL
- DXmSUPPRESS_PRINT_AFTER
- DXmSUPPRESS_PRINT_FORMAT
- DXmSUPPRESS_PRINTER
- DXmSUPPRESS_PRINTER_FORM
- DXmSUPPRESS_PRIORITY
- DXmSUPPRESS_SETUP
- DXmSUPPRESS_SHEET_COUNT
- DXmSUPPRESS_SHEET_SIZE
- DXmSUPPRESS_SIDES

- DXmSUPPRESS_START_SHEET_COMMENT

The default is DXmSUPPRESS_NONE, or 0. Access is CSG.

DXmNunmanageOnCancel
A Boolean value that specifies whether the print widget automatically unmanages itself when the Cancel button in the primary box is pressed. The default is False. Access is CSG.

DXmNunmanageOnOk
A Boolean value that specifies whether the print widget automatically unmanages itself when the OK button in the primary box is pressed. The default is False. Access is CSG.

XmNcancelCallback
Callback made when the user clicks on the Cancel push button in the print widget primary dialog box. If the DXmNunmanageOnCancel resource is set, the print widget automatically unmanages itself when the Cancel button in the primary box is pressed. The callback reason is **XmCR_CANCEL**. The default is Null. Access is CSG.

Your application can use the **XmNcancelCallback** callback to perform related functions as well.

XmNokCallback
Callback made when the user clicks on the OK push button in the print widget primary dialog box. If the DXmNunmanageOnOk resource is set, the print widget automatically unmanages itself when the OK button in the primary box is pressed. The callback reason is **XmCR_OK**. The default is Null. Access is CSG.

Note that your application can use the **XmNokCallback** callback to perform other functions, such as calling DXmPrintWgtPrintJob to submit the print job.

Callback Structure

```
typedef struct
{
      int     reason;
      XEvent  *event;
} XmAnyCallbackStruct;
```

Callback Field Descriptions

reason
An integer set to the callback reason. See the Callback Reasons section for the values that are valid for this widget.

event
A pointer to the X event structure describing the event that generated this callback.

Callback Reasons

XmCR_OK
The user activated the OK push button in the print widget primary dialog box.

XmCR_CANCEL
The user activated the Cancel push button in the print widget primary dialog box.

Description

The DXmCreatePrintDialog routine creates the print widget, which is a modeless widget that provides applications with a fast, convenient method of printing one or more files in multiple formats.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the print widget. See also the following routines for related information:

Routine	Description
DXmPrintWgtAugmentList	Defines additional print formats
DXmPrintWgtPrintJob	Submits a print job

DXmCreateScrolledCSText

Creates a compound string text widget with scroll bars.

Widget Class Hierarchy

Core Resource Set
\
XmPrimitive Resource Set
\
DXmScrolledCSText Resource Set

Format

```
Widget DXmCreateScrolledCSText(parent, name, args, num_args)
        Widget    parent;
        char      *name;
        Args      *args;
        Cardinal  num_args;
```

Return Value

The identifier (widget ID) of the created compound string text widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
The name of the created widget.

args
The application override argument list.

num_args
The number of arguments in the application override argument list.

Description

The DXmCreateScrolledCSText routine uses the same arguments and resources as the DXmCreateCSText routine to create a compound string text widget but uses the following resources to create horizontal and vertical scroll bars for the widget as well:

Resource	Description
XmNscrollHorizontal	Controls whether a scroll bar that allows the user to scroll through text horizontally is added to the compound string text window.
XmNscrollTopSide	Controls whether the horizontal scroll bar is positioned at the top of the compound string text window.
XmNscrollVertical	Controls whether a scroll bar that allows the user to scroll through text vertically is added to the compound string text window.
XmNscrollLeftSide	Controls whether the vertical scroll bar is placed on the left side of the compound string text window.

See the DXmCreateCSText routine for a complete description of all resources.

DXmCreateSvn

Creates an SVN widget

Widget Class Hierarchy

Core Resource Set
\
Composite Resource Set
\
Constraint Resource Set
\
XmManager Resource Set
\
DXmSvn Resource Set

Format

```
Widget DXmCreateSvn(parent, name, arglist, argcount)
        Widget  parent;
        char    *name;
        Arg     *arglist;
        int     argcount;
```

Resources

You can set the following widget-specific resources in the **arglist**:

```
XtCallbackList  DXmSvnNattachToSourceCallback
Boolean         DXmSvnNcolumnLines;
Dimension       DXmSvnNdefaultSpacing;
XtCallbackList  DXmSvnNdetachFromSourceCallback
XtCallbackList  DXmSvnNdisplayChangedCallback
short           DXmSvnNdisplayMode;
XtCallbackList  DXmSvnNdraggingCallback
XtCallbackList  DXmSvnNdraggingEndCallback
XtCallbackList  DXmSvnNentrySelectedCallback
XtCallbackList  DXmSvnNentryTransferCallback
XtCallbackList  DXmSvnNentryUnselectedCallback
Boolean         DXmSvnNexpectHighlighting;
XtCallbackList  DXmSvnNextendConfirmCallback
Boolean         DXmSvnNfixedWidthEntries;
XmFontList      DXmSvnNfontList;
XmFontList      DXmSvnNfontListLevel0;
XmFontList      DXmSvnNfontListLevel1;
XmFontList      DXmSvnNfontListLevel2;
XmFontList      DXmSvnNfontListLevel3;
XmFontList      DXmSvnNfontListLevel4;
Boolean         DXmSvnNforceSeqGetEntry;
XtCallbackList  DXmSvnNgetEntryCallback
Dimension       DXmSvnNghostHeight;
Pixmap          DXmSvnNghostPixmap;
Dimension       DXmSvnNghostWidth;
Position        DXmSvnNghostX;
Position        DXmSvnNghostY;
XtCallbackList  DXmSvnNhelpRequestedCallback;
Dimension       DXmSvnNindentMargin;
Boolean         DXmSvnNliveScrolling;
Boolean         DXmSvnNmultipleSelections;
XmString        DXmSvnNnavWindowTitle;
longword        DXmSvnNnumberOfEntries;
Widget          DXmSvnNoutlineHScrollWidget;
Widget          DXmSvnNpaneWidget;
XtCallbackList  DXmSvnNpopupMenuCallback
longword        DXmSvnNprimaryPercentage
Widget          DXmSvnNprimaryWindowWidget
Position        DXmSvnNsecondaryBaseX;
Boolean         DXmSvnNsecondaryComponentsUnmapped;
Widget          DXmSvnNsecondaryWindowWidget;
XtCallbackList  DXmSvnNselectAndConfirmCallback
short           DXmSvnNselectionMode;
XtCallbackList  DXmSvnNselectionsDraggedCallback
Boolean         DXmSvnNshowPathToRoot;
short           DXmSvnNstartColumnComponent;
short           DXmSvnNstartLocationCursor;
XtCallbackList  DXmSvnNtransitionsDoneCallback
```

```
Dimension       DXmSvnNtreeArcWidth;
Boolean         DXmSvnNtreeCenteredComponents;
Boolean         DXmSvnNtreeEntryOutlines;
Boolean         DXmSvnNtreeEntryShadows;
Boolean         DXmSvnNtreeIndexAll;
Dimension       DXmSvnNtreeLevelSpacing;
Boolean         DXmSvnNtreePerpendicularLines;
Dimension       DXmSvnNtreeSiblingSpacing;
short           DXmSvnNtreeStyle;
Boolean         DXmSvnNtruncateText;
Boolean         DXmSvnNuseScrollButtons;
```

Return Value

The identifier (widget ID) of the created SVN widget.

Arguments

parent
The identifier (widget ID) of the parent widget.

name
A character string that defines the name of the widget.

arglist
The application argument list.

argcount
An integer that represents the number of arguments in the application argument list. If there are no arguments in the argument list, argcount must equal 0. However, when argcount equals 0, the argument list does not have to be Null.

Resources

DXmSvnNattachToSourceCallback
One or more callbacks made when a widget attaches to the source data module, which then call the DXmSvnAddEntries routine to specify the initial entries (and number of entries) in the data hierarchy. The callback reason is **DXmSvnCRAttachToSource**. The default is Null. Access is CG.

DXmSvnNcolumnLines
A Boolean resource that, when True, specifies that columns are to be separated with a line. The default is False. Access is CSG.

DXmSvnNdefaultSpacing
The number of pixels between components. The default is 12 pixels. Access is CSG.

DXmSvnNdetachFromSourceCallback
Callback made when the widget is detached from the source data module (the widget is being destroyed). The application records that this widget will never call back. The callback reason is **DXmSvnCRDetachFromSource**. The default is Null. Access is CSG.

DXmSvnNdisplayChangedCallback
Callback made when the entries currently being displayed to the user have changed. This change may be in response to an expand or collapse operation, scrolling, or the application issuing calls to the DXmSvnPositionDisplay routine.

The callback reason is **DXmSvnCRDisplayChanged**. An additional callback field is **loc_cursor_entry_number**, which is the number of the entry where the location cursor is currently displayed. The default is Null. Access is CSG.

DXmSvnNdisplayMode
Specifies which display should appear on the screen, indicated by one of the following values:

Value	Description
DXmSvnKdisplayOutline	Display in outline mode.
DXmSvnKdisplayColumns	Display in column mode.
DXmSvnKdisplayTree	Display in tree mode.
DXmSvnKdisplayAllModes	Used only by the DXmSvnSetComponentHidden routine to hide a component in all modes.
DXmSvnKdisplayNone	Used only by the DXmSvnSetComponentHidden routine to display (not hide) a component in any mode.

The default is DXmSvnKdisplayOutline. Access is CSG.

DXmSvnNdraggingCallback

Callback made when a user presses MB2 to drag a set of entries. This callback is required for application-controlled dragging (which you set by using the DXmSvnSetApplDragging routine).

The callback reason is **DXmSvnCRDragging**. Other callback fields are as follows:

Field	Description
time	The time that the button press event occurred
x	The *x* position of the pointer
y	The *y* position of the pointer
dragged_entry_number	The number of the entry currently being dragged while MB2 is pressed
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the XEvent that generated this callback

The default is Null. Access is CSG.

DXmSvnNdraggingEndCallback

Callback made when the user releases MB2 after dragging a set of entries. This callback is required for application-controlled dragging (which you set by using the DXmSvnSetApplDragging routine).

The callback reason is **DXmSvnCRDraggingEnd**. Other callback fields are as follows:

Field	Description
time	The time that the button release event occurred
x	The *x* position of the cursor when the mouse button was released
y	The *y* position of the cursor when the mouse button was released
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed

Field	Description
event	A pointer to the XEvent that generated this callback

The default is Null. Access is CSG.

DXmSvnNentrySelectedCallback

Callback made when the user selects an entry. The fields provided in the callback structure enable the application to obtain more information about the selected entry from the source module.

The callback reason is **DXmSvnCREntrySelected**. Other callback fields are as follows:

Field	Description
entry_number	The entry selected.
component_number	The component within the entry the mouse was positioned over when the entry was selected.
time	Time when the entry was selected.
entry_tag	The value (previously set in the application with the DXmSvnSetEntry or DXmSvnSetEntryTag routine) associated with the **entry_number** field.
first_selection	One of the following values:

Value	Description
DXmSvnKnotFirst	This is not the first selection that has been made.
DXmSvnKfirstOfOne	There is only one selection being made.
DXmSvnKfirstOfMany	This is the first of many selections being made.

This value (when treated as a Boolean function) is usually True. It is False only when the user has made a range selection *and* this is not the first DXmSvnNentrySelectedCallback.

entry_level	The entry's level number.

Field	Description
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed.
event	A pointer to the XEvent that generated this callback.

The default is Null. Access is CSG.

DXmSvnNentryTransferCallback

Callback made when the user clicks on MB2 without moving the mouse.

The callback reason is **DXmSvnCREntryTransfer**. Other callback fields are as follows:

Field	Description
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed.
transfer_mode	The value in this field reflects whether or not the user pressed a modifier key with an MB2 click and how the application should respond (either taking no action at all, or moving or copying the entry). The possible values for this field are as follows:

Value	Description
DXmSvnKtransferUnknown	No modifier was specified with the MB2 click.
DXmSvnKtransferMove	The ALT modifier was specified with the MB2 click.
DXmSvnKtransferCopy	The Ctrl modifier was specified with the MB2 click.

event	A pointer to the X event structure describing the event that generated this callback.

The default is Null. Access is CSG.

DXmSvnNentryUnselectedCallback

Callback made when the user cancels (''unselects'') the selection of an entry.

The fields provided in the callback structure enable an application to obtain more information about the entry from its source module.

The callback reason is **DXmSvnCREntryUnselected**. Other callback fields are as follows:

Field	Description
entry_number	The canceled entry
component_number	The component within the entry the mouse was positioned over when the entry was canceled
time	The time when the selection was canceled
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

The default is Null. Access is CSG.

DXmSvnNexpectHighlighting

A Boolean resource that, when True, creates a rectangle large enough to highlight the entries. The default is False. Access is CSG.

DXmSvnNextendConfirmCallback

Callback made when the user double clicks MB1 with multiple entries selected. To determine which entries are selected, the application must call the DXmSvnGetNumSelections and the DXmSvnGetSelections routines.

The callback reason is **DXmSvnCRExtendConfirm**. Other callback fields are as follows:

Field	Description
entry_number	The entry selected
time	The time when the double click occurred
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

The default is Null. Access is CSG.

DXmSvnNfixedWidthEntries
A Boolean resource that, when True, specifies that all entries, including the selectable white space to the right of each entry, will be the same width. Fixed width entries are the width of the display window. The default is True. Access is CSG.

DXmSvnNfontList
The default font when no level font is specified. The default is DXmDefaultFont. Access is CSG.

DXmSvnNfontListLevel0
The default font list used by level 0 entries. The default is Null. Access is CSG.

DXmSvnNfontListLevel1
The default font list used by level 1 entries. The default is Null. Access is CSG.

DXmSvnNfontListLevel2
The default font list used by level 2 entries. The default is Null. Access is CSG.

DXmSvnNfontListLevel3
The default font list used by level 3 entries. The default is Null. Access is CSG.

DXmSvnNfontListLevel4
The default font list used by level 4 entries. The default is Null. Access is CSG.

DXmSvnNforceSeqGetEntry
A Boolean resource that, when True, specifies that the SVN widget generate the DXmSvnNgetEntryCallback callbacks in sequential order only. The default is False. Access is CSG.

DXmSvnNgetEntryCallback
Callback made when the SVN widget requires information about an entry. The application should respond by calling the DXmSvnSetEntry routine. Control should not be returned from the callback until this call and subsequent calls are made to the DXmSvnSetComponentPixmap, DXmSvnSetComponentText, or DXmSvnSetComponentWidget routines.

The callback reason is **DXmSvnCRGetEntry**. Other callback fields are as follows:

Field	Description
entry_number	The entry number required
entry_tag	The value (previously set in the application with the DXmSvnSetEntry or DXmSvnAddEntries routine) associated with the **entry_number** field
entry_level	The entry's level number
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed

The default is Null. Access is CSG.

DXmSvnNghostHeight
The height of the pixmap for a specific type of highlighted image, or ''ghost'', that follows the entries as they are dragged. The default is 0. Access is CSG.

DXmSvnNghostPixmap
The pixmap for a specific type of highlighted image, or ''ghost'', that follows the entries as they are dragged. The default is Null. Access is CSG.

DXmSvnNghostWidth
The width of the pixmap for a specific type of highlighted image, or ''ghost'', that follows the entries as they are dragged. The default is 0. Access is CSG.

DXmSvnNghostX
Relative *x* position on the screen where the dragging ''ghost'' (a specific type of highlighted image that follows the entries as they are dragged) is attached to the cursor. The default is 0. Access is CSG.

DXmSvnNghostY
Relative *y* position on the screen where the dragging ''ghost'' (a specific type of highlighted image that follows the entries as they are dragged) is attached to the cursor. The default is 0. Access is CSG.

DXmSvnNhelpRequestedCallback

Callback made when the user requests help. The callback reason is **DXmSvnCRHelpRequested**. Other callback fields are as follows:

Field	Description
component_number	The component number within the entry on which help was requested
time	The time when the user requested help
entry_number	The entry number for which help was requested
entry_tag	The value (previously set in the application with the DXmSvnSetEntry routine) associated with the **entry_number** field
entry_level	The level number of the entry
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

Note that the values for the **entry_number** and **loc_cursor_entry_number** fields vary depending on how the user requests help, as follows:

- The user presses the Help key on the keyboard.

  In this instance, the SVN widget returns a valid entry number in the **loc_cursor_entry_number** field and then invokes SVN help for the corresponding entry.

- The user invokes context-sensitive help and then clicks MB1.

  In this instance, the application uses context-sensitive help (which you include in the application by specifying the DXmHelpOnContext routine) to invoke help for an SVN subwidget. A negative value in the **entry_number** field indicates which of the following subwidgets the user has clicked on to request help:

Subwidget	Value
DXmSvnKHelpScroll	-1
DXmSvnKHelpNavButton	-2
DXmSvnKNavWindow	-3

The default is Null. Access is CSG.

DXmSvnNindentMargin
The number of pixels each entry level should be indented. The default is 16 pixels. Access is CSG.

DXmSvnNliveScrolling
A Boolean resource that, when True, allows scrolling in outline and column display mode, but not in tree mode. If an application uses the existing index window instead, this resource must be set to False. The default is True. Access is CSG.

DXmSvnNprimaryPercentage
The percentage of the display window that is in the primary side of the column view. The default is 50 percent. Access is CSG.

DXmSvnNprimaryWindowWidget
The identifier (widget ID) of the primary window widget in the SVN display. The default is Null. Access is G (read-only).

DXmSvnNmultipleSelections
A Boolean resource that, when True, allows the user to specify multiple selections. The default is True. Access is CSG.

DXmSvnNnavWindowTitle
The title for a pop-up navigation window. The default is Null. Access is CSG.

DXmSvnNnumberOfEntries
The number of entries. The initial value is 0. This value increases or decreases as entries are changed by the DXmSvnAddEntries and DXmSvnDeleteEntries routines. Access is G (read-only).

DXmSvnNoutlineHScrollWidget
The identifier (widget ID) of the primary horizontal scroll bar. The default is Null. Access is G (read-only).

DXmSvnNpaneWidget
The identifier (widget ID) of the pane widget. The default is Null. Access is G (read-only).

DXmSvnNpopupMenuCallback

Callback made after an MB3 click that notifies the application to generate a pop-up menu for this entry.

The callback reason is **DXmSvnCRPopupMenu**. Other callback fields are as follows:

Field	Description
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

The default is Null. Access is CSG.

DXmSvnNsecondaryBaseX

The base *x* position of the secondary window in the column view. The default is 0. Access is CSG.

DXmSvnNsecondaryComponentsUnmapped

A Boolean resource that, when True, specifies that secondary components still exist even though they are not visible in the column view. The default is False. Access is CSG.

DXmSvnNsecondaryWindowWidget

The identifier (widget ID) of the secondary widget in the SVN display. The default is Null. Access is G (read-only).

DXmSvnNselectAndConfirmCallback

Callback made when the user double clicks MB1 on a single entry, indicating that the user wants to expand or collapse (contract) the entry. The application can determine which entry has been selected by either examining the **entry_number** and **component_number** fields of the callback data structure or by calling the DXmSvnGetSelections routine.

This callback is also called when all three of the following conditions exist:

1. The DXmSvnNMultipleSelections resource is set to True.
2. There is no DXmSvnNExtendConfirm callback supplied.
3. The user has double clicked to terminate the selection of a range of entries.

In such an instance, the individual calls are from the highest entry number selected through the lowest entry number selected.

The callback reason is **DXmSvnCRSelectAndConfirm**. Other callback fields are as follows:

Field	Description
component_number	The component number on which the mouse was positioned
time	When the second click of the double click occurred
entry_number	The entry number of the selected entry
entry_tag	The value (previously set in the application with the DXmSvnSetEntry or DXmSvnAddEntries routine) associated with the **entry_number** field
entry_level	The level number of the entry
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

The default is Null. Access is CSG.

DXmSvnNselectionMode

Specifies what is to be selected. The selection mode, which is used by the column view, is indicated by one of the following values:

Value	Description
DXmSvnKselectEntry	Selects the entire entry.
DXmSvnKselectComp	Selects only the component.
DXmSvnKselectCompAndPrimary	Selects the current component and the entire primary side of the column view.
DXmSvnKselectEntryOrComp	Selects either the entire entry (if the selection is made on the primary side) or only the component (if the selection is made on the secondary side of the display window).

The default is DXmSvnKselectEntry. Access is CSG.

DXmSvnNselectionsDraggedCallback

The user has used MB2 to drag selected entries to another location. Note that the use of this resource is valid only when application-controlled dragging has not been set (with the DXmSvnNdraggingCallback and DXmSvnNdraggingEndCallback callbacks).

The callback reason is **DXmSvnCRSelectionsDragged**. Other callback fields are as follows:

Field	Description
component_number	The component number on which the mouse button was released
x	The *x* position at which the mouse button was released
y	The *y* position at which the mouse button was released
entry_number	The entry number on which the mouse button was released

Field	Description
entry_tag	The value (previously set in the application with the DXmSvnSetEntry routine) associated with the **entry_number** field
entry_level	The entry's level number
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

The default is Null (dragging will *not* occur). Access is CSG.

DXmSvnNshowPathToRoot

A Boolean resource that, when True, shows the path to the root, which is the first entry level in the display (level 0). This path is represented by a horizontal line over which the hierarchy of entry levels (each indented) is displayed. The default is True. Access is CSG.

DXmSvnNstartColumnComponent

The number of the component that begins the secondary column. The default is 0, which indicates that no components should be displayed in the secondary column. Access is CSG.

DXmSvnNstartLocationCursor

Allows the application to specify the entry where the location cursor will be displayed when the SVN widget is first activated. You can set this resource only at widget creation time; it cannot be changed thereafter. The default position of the location cursor is entry number 1. Access is C.

DXmSvnNtransitionsDoneCallback

Callback made when the user has released MB1 following an application-controlled drag operation.

The callback reason is **DXmSvnCRTransitionsDone**. Other callback fields are as follows:

Field	Description
loc_cursor_entry_number	The number of the entry where the location cursor is currently displayed
event	A pointer to the X event structure describing the event that generated this callback

The default is Null. Access is CSG.

DXmSvnNtreeArcWidth
The width for the arc of the outline that delineates the boundary between each tree display mode entry. The default is 15 pixels. Access is CSG.

DXmSvnNtreeCenteredComponents
A Boolean resource that, when True, centers components vertically within an entry. The SVN widget automatically alters the size of the entries to fit the centered components. The default is False. Access is CSG.

DXmSvnNtreeEntryOutlines
A Boolean resource that, when True, displays an outline around tree mode entries. The default is True. Access is CSG.

DXmSvnNtreeEntryShadows
A Boolean resource that, when True, adds shadowing to the outline surrounding tree mode entries. The default is True. Access is CSG.

DXmSvnNtreeIndexAll
A Boolean resource that, when True, specifies that all tree mode entries appear in the index window when the user is scrolling. The default is True. Access is CSG.

DXmSvnNtreeLevelSpacing
Specifies how much space, in pixels, to insert between tree levels. The default is 5 pixels. Access is CSG.

DXmSvnNtreePerpendicularLines
A Boolean resource that, when True, specifies that lines connecting the entries in tree mode are perpendicular instead of diagonal. The default is True.

DXmSvnNtreeSiblingSpacing
Specifies how much space, in pixels, to insert between sibling tree mode entries. The default is 5 pixels. Access is CSG.

DXmSvnNtreeStyle

Specifies the style (format) of a tree display, as indicated by one of the following values:

Value	Description
DXmSvnKtopTree	A vertical tree format (oriented from the top).
DXmSvnKhorizontalTree	A horizontal tree format (oriented from the left).
DXmSvnKoutlineTree	Oriented in outline form.
DXmSvnKuserDefinedTree	Oriented in an application-defined format. The SVN widget uses the x- and y-coordinate values you specify for an entry to determine the entry's position in the tree.

The default is DXmSvnKoutlineTree. Access is CSG.

DXmSvnNtruncateText

A Boolean resource that, when True, truncates text that is too long for the display. The default is False. Access is CSG.

DXmSvnNuseScrollButtons

A Boolean resource that, when True, creates outer arrows on a scroll bar, which the user can click on to move the display. The default is True. Access is CSG.

Callback Structure

```
typedef struct
{
  int             reason;
  int             entry_number;
  int             component_number;
  int             first_selection;
  int             x;
  int             y;
  XtPointer       entry_tag;
  Time            time;
  int             entry_level;
  int             loc_cursor_entry_number;
  int             transfer_mode;
  int             dragged_entry_number;
  XEvent          *event;
} DXmSvnCallbackStruct;
```

Note that all callbacks do not use all the fields listed except **reason**, which is common to all.

Callback Field Descriptions

reason
An integer set to the callback reason.

entry_number
The identifier of the entry.

component_number
The identifier of the component.

first_selection
One of the following values:

Value	Description
DXmSvnKnotFirst	This is not the first selection that has been made.
DXmSvnKfirstOfOne	There is only one selection being made.
DXmSvnKfirstOfMany	This is the first of many selections being made.

This value (when treated as a Boolean function) is usually True. It is False only when the user has made a range selection *and* this is not the first DXmSvnNentrySelectedCallback.

x
The *x* position at which the event occurred.

y
The *y* position at which the event occurred.

entry_tag
The value associated with the **entry_number** field. You can set this value in your application using a routine that is appropriate for the particular callback, as follows:

Callback	Routine
DXmSvnNentrySelectedCallback	DXmSvnSetEntry DXmSvnSetEntryTag
DXmSvnNgetEntryCallback	DXmSvnSetEntry DXmSvnAddEntries
DXmSvnNhelpRequestedCallback	DXmSvnSetEntry
DXmSvnNselectAndConfirmCallback	DXmSvnSetEntry DXmSvnAddEntries
DXmSvnNselectionsDraggedCallback	DXmSvnSetEntry

time
The time when the event occurred.

entry_level
The current entry number.

loc_cursor_entry_number
The number of the entry that currently has the location cursor.

transfer_mode
A constant specifying the transfer operation that should occur after an MB2 click with no mouse movement.

dragged_entry_number
Contains the number of the entry currently being dragged with MB2 pressed.

event
A pointer to the X event structure describing the event that generated this callback.

Callback Reasons

The following table lists the reasons and the valid callback fields for each:

Reason	Valid Fields
DXmSvnCRAttachToSource	reason
DXmSvnCRDetachFromSource	reason
DXmSvnCRDisplayChanged	reason
DXmSvnCRDragging	reason, x, y, time, loc_cursor_entry_number, dragged_entry_number, event
DXmSvnCRDraggingEnd	reason, x, y, time, loc_cursor_entry_number, event
DXmSvnCREntrySelected	reason, entry_number, component_number, first_selection, entry_tag, time, loc_cursor_entry_number, event
DXmSvnCREntryTransfer	reason, loc_cursor_entry_number, transfer_mode, event
DXmSvnCREntryUnselected	reason, entry_number, component_number, time, loc_cursor_entry_number
DXmSvnCRExtendConfirm	reason, entry_number, time, loc_cursor_entry_number, event
DXmSvnCRGetEntry	reason, entry_number, entry_tag, entry_level, loc_cursor_entry_number
DXmSvnCRHelpRequested	reason, entry_number, component_number, entry_tag, time, entry_level, loc_cursor_entry_number, event
DXmSvnCRPopupMenu	reason, loc_cursor_entry_number, event
DXmSvnCRSelectAndConfirm	reason, entry_number, component_number, entry_tag, time, entry_level, loc_cursor_entry_number, event
DXmSvnCRSelectionsDragged	reason, entry_number, component_number, x, y, entry_tag, entry_level, loc_cursor_entry_number, event
DXmSvnCRTransitionsDone	reason, event

Keyboard Bindings

The following table describes how specific keys function within the SVN environment. The table also lists the Motif name and LK401 keyboard equivalent for each key. See the *OSF/Motif Style Guide* and the *DECwindows Companion to the OSF/Motif Style Guide* for additional information about DECwindows Motif keyboard bindings.

Motif Name	LK401 Equivalent[1]	SVN Function
KActivate	Return Ctrl+Return	Selects and confirms the current entry. Equivalent to a double MB1 click on the mouse.
KBeginData	Ctrl+Alt+ ←	Scrolls to the first entry in the SVN display; selects and moves the location cursor to that entry.
KBeginLine	Alt+←	Scrolls to the extreme left of the display.
KDown	↓	Selects and moves the location cursor to the next entry in display, scrolling the display down if necessary.
KEndData	Ctrl+Alt+ →	Scrolls to the last entry in the SVN display; selects and moves the location cursor to that entry.
KEndLine	Alt+→	Scrolls to the extreme right of the display.
KExtendDown	Shift+↓	Adds the next entry to the extended selection. Selects and moves the location cursor to the next entry.
KExtendUp	Shift+ ↑	Adds the previous entry to the extended selection. Selects and moves the location cursor to the previous entry.
KHelp	Help	Reports the help callback to the application if specified.
KLeft	←	Scrolls the display one character to the left.
KMenu	F4	Reports the DXmNpopupMenuCallback to the application if specified for the entry with the location cursor.

[1]Note that on an LK201 keyboard, *Prev* is *Prev Screen*, *Next* is *Next Screen*, and *Alt* is *Compose Character*.

Motif Name	LK401 Equivalent[1]	SVN Function
KNextColumn	Ctrl+ →	Selects the next column component to the right.
KNextPara	Ctrl+ ↓	Scrolls to the bottom of the display.
KPageDown	Next	Scrolls the display to the next screen of entries. Selects and moves the location cursor to the entry located at the same relative position on the next screen.
KPageLeft	Ctrl+Prev	Scrolls the display one screen to the left.
KPageRight	Ctrl+Next	Scrolls the display one screen to the right.
KPageUp	Prev	Scrolls the display to the previous screen of entries. Selects and moves the location cursor to the entry located at the same relative position on the previous screen.
KPrevColumn	Ctrl+ ←	Selects the previous column component to the left.
KPrevPara	Ctrl+ ↑	Scrolls to the top of the display.
KRight	→	Scrolls the display one character to the right.
KSelect	Select	Selects the current entry and cancels all other selected entries.
KSelectAll	Ctrl+Slash (/)	Selects all entries known to SVN.
KUp	↑	Selects and moves the location cursor to the previous entry in the display, scrolling the display if necessary.

[1]Note that on an LK201 keyboard, *Prev* is *Prev Screen*, *Next* is *Next Screen*, and *Alt* is *Compose Character*.

Description

The DXmCreateSvn routine allows you to create a hierarchical structure of information that is presented to a user in an organized manner and in a way that lets the user navigate in, and select from, that structure. Your application need only tell the SVN widget about the organization of the data and respond to SVN widget callbacks; the SVN widget is responsible for actually displaying the data.

You can use SVN to display hierarchical information in three different formats, or modes:

- Outline format.
- Tree format, which can be oriented from the top, the left, in outline form, and in a user-defined style.
- Column format, which differs from outline format in that a window pane separates a set of components from the rest of the display. You can scroll horizontally on each side, independently of the other side. However, you have only one vertical scroll bar.

Each SVN line, or entry, in your hierarchy can display as many as thirty pieces of information (called components) depending on the amount of information users need. The components can be of three data types: text, pixmaps, and widgets.

The SVN widget includes additional support routines (also described in this document) that allow your application to insert and remove components, set the text associated with a component, set a component to be hidden, set and determine a component's width, and determine a component's number.

When you create an SVN widget, note that applications must be able to communicate with the widget through Toolkit routine calls that request information about and manipulate selections. As such, when you use the DXmCreateSvn routine to create the SVN widget, you **must** include the following SVN callbacks:

- DXmSvnNattachToSourceCallback
- DXmSvnNgetEntryCallback
- DXmSvnNselectAndConfirmCallback

See the *DECwindows Motif Guide to Application Programming* for a complete description of the SVN widget. See also the following SVN routines for related information.

Routine	Description
DXmSvnAddEntries	Opens new entries in the structure.
DXmSvnAutoScrollCheck	Determines whether the DXmSvnAutoScrollDisplay routine should be called.

Routine	Description
DXmSvnAutoScrollDisplay	Automatically scrolls the screen one unit based on the cursor position.
DXmSvnClearHighlight	Removes (clears) highlighting from a specified entry.
DXmSvnClearHighlighting	Removes (clears) all highlighting from the entries.
DXmSvnClearSelection	Clears (cancels) the selection of a specified entry.
DXmSvnClearSelections	Clears (cancels) all selections.
DXmSvnDeleteEntries	Deletes entries from the structure.
DXmSvnDisableDisplay	Temporarily disables the SVN widget from generating a display.
DXmSvnEnableDisplay	Reenables the SVN widget.
DXmSvnFlushEntry	Displays the specified entry on the screen if it is the next logically displayed entry.
DXmSvnGetComponentNumber	Retrieves (returns) the component number containing a supplied tag value.
DXmSvnGetComponentTag	Retrieves (returns) a component tag for a specified component number.
DXmSvnGetComponentText	Retrieves (returns) the address of a compound string.
DXmSvnGetComponentWidth	Retrieves (returns) the width of a specified component.
DXmSvnGetDisplayed	Retrieves (returns) displayed entry numbers and related information necessary to draw a corresponding display.
DXmSvnGetEntryLevel	Retrieves (returns) the level of an entry in the display.
DXmSvnGetEntryNumber	Retrieves (returns) the entry number for a specified entry tag value.
DXmSvnGetEntryPosition	Retrieves (returns) the position of an entry in the display.

DECwindows Toolkit Extensions
DXmCreateSvn

Routine	Description
DXmSvnGetEntrySensitivity	Retrieves (returns) the sensitivity of an entry in the display.
DXmSvnGetEntryTag	Retrieves (returns) an entry's tag for a specified entry number.
DXmSvnGetHighlighted	Retrieves (returns) a list of entries currently highlighted by the user.
DXmSvnGetNumDisplayed	Retrieves (returns) the number of entries currently being displayed in the window.
DXmSvnGetNumHighlighted	Retrieves (returns) the number of highlighted entries.
DXmSvnGetNumSelections	Retrieves (returns) the number of selected entries.
DXmSvnGetPrimaryWorkWidget	Retrieves (returns) the widget ID of the primary SVN work widget (window).
DXmSvnGetSecondaryWorkWidget	Retrieves (returns) the widget ID of the secondary SVN work widget (window).
DXmSvnGetSelections	Retrieves (returns) a list of selected entry numbers.
DXmSvnGetTreePosition	Retrieves (returns) the position of the display window for future use.
DXmSvnHideHighlighting	Specifies that the SVN widget show highlighted entries in the default format, without using highlighting graphics.
DXmSvnHideSelections	Specifies that the SVN widget show selected entries in the default format, without using selection graphics.
DXmSvnHighlightAll	Highlights all entries.
DXmSvnHighlightEntry	Highlights a specified entry.
DXmSvnInsertComponent	Inserts a new component into each SVN entry.
DXmSvnInvalidateEntry	Specifies that the SVN widget redisplay (update) an entry.
DXmSvnMapPosition	Associates the *x* and *y* position with an entry number.
DXmSvnPositionDisplay	Displays a specified entry.

Routine	Description
DXmSvnRemoveComponent	Removes a component from each SVN entry.
DXmSvnSelectAll	Selects all entries.
DXmSvnSelectComponent	Selects a specified entry using a particular component.
DXmSvnSelectEntry	Selects a specified entry.
DXmSvnSetApplDragging	Turns application dragging mode on or off in the SVN widget.
DXmSvnSetComponentHidden	Controls which modes the SVN widget uses to display a particular component.
DXmSvnSetComponentPixmap	Adds a pixmap component to an entry.
DXmSvnSetComponentTag	Sets the tag of a component.
DXmSvnSetComponentText	Adds a read-only component string to an entry.
DXmSvnSetComponentWidget	Adds an SVN subwidget component to an entry.
DXmSvnSetComponentWidth	Sets the width of a component.
DXmSvnSetEntry	Provides the SVN widget with new entry level information.
DXmSvnSetEntryIndexWindow	Controls whether the SVN widget displays the entry in the index window.
DXmSvnSetEntryNumComponents	Provides the SVN widget with the number of components for an entry.
DXmSvnSetEntryPosition	Sets the position of an entry in user-defined tree mode.
DXmSvnSetEntrySensitivity	Controls whether the user can select a particular entry.
DXmSvnSetEntryTag	Sets an entry tag with a specified entry.
DXmSvnSetTreePosition	Sets the position of the tree in tree display mode.
DXmSvnShowHighlighting	Specifies that the SVN widget use highlight graphics when showing highlighted entries.

DECwindows Toolkit Extensions
DXmCreateSvn

Routine	Description
DXmSvnShowSelections	Specifies that the SVN widget use selection graphics on the entries when showing selections.
DXmSvnValidateAll	Ensures that all entries are valid.

DXmCSContainsStringCharSet

Examines XmString segments for ISO_LATIN_1 characters.

Format

```
Boolean DXmCSContainsStringCharSet(str)
        XmString  str;
```

Return Value

A Boolean value that, when True, indicates that all XmString segments contain characters from the ISO_LATIN_1 character set.

Arguments

str
A compound string.

Description

The DXmCSContainsStringCharSet routine examines all XmString segments in the application for characters belonging to the ISO_LATIN_1 character set. If non-ISO_LATIN_1 characters are detected, this routine will return False.

DXmCSTextClearSelection

Clears (cancels) the global selection highlighted in the compound string text widget.

Format

```
void DXmCSTextClearSelection(widget, time)
      Widget  widget;
      Time    time;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

time
The time of the event that generated the call to DXmCSTextClearSelection.

Description

The DXmCSTextClearSelection routine clears (cancels) the global selection highlighted in the compound string text widget. See the DXmCreateCSText, DXmCSTextGetSelection, and DXmCSTextSetSelection routines for related information.

DXmCSTextCopy

Copies the currently selected (highlighted) text to the clipboard.

Format

```
Boolean DXmCSTextCopy (widget, time)
         Widget  widget;
         Time    time;
```

Return Value

A Boolean value that, when True, indicates that the text has been successfully copied to the clipboard.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

time
The time of the event that generated the call to DXmCSTextCopy.

Description

The DXmCSTextCopy routine copies the selected (highlighted) text in the compound string text widget to the clipboard, returning the success or failure of that operation. (See the *OSF/Motif Programmer's Reference* and the *OSF/Motif Programmer's Guide* for more information about the clipboard.) See the DXmCreateCSText, DXmCSTextCut, DXmCSTextPaste, and DXmCSTextRemove routines for related information.

DXmCSTextCut

Deletes the currently selected (highlighted) text after copying it to the clipboard.

Format

```
Boolean DXmCSTextCut(widget, time)
         Widget  widget;
         Time    time;
```

Return Value

A Boolean value that, when True, indicates that the text has been successfully copied to the clipboard and then deleted.

Arguments

widget

The identifier (widget ID) of the compound string text widget.

time

The time of the event that generated the call to DXmCSTextCut.

Description

The DXmCSTextCut routine deletes (cuts) the selected (highlighted) text in the compound string text widget after copying the text to the clipboard, returning the success or failure of that operation. (See the *OSF/Motif Programmer's Reference* and the *OSF/Motif Programmer's Guide* for more information about the clipboard.) See the DXmCreateCSText, DXmCSTextCopy, DXmCSTextPaste, and DXmCSTextRemove routines for related information.

DXmCSTextDisableRedisplay

Temporarily prevents the visual update of the DXmCSText widget.

Format

```
void DXmCSTextDisableRedisplay(widget)
        Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextDisableRedisplay routine prevents redisplay of the specified DXmCSText widget even though its visual attributes have been modified. The visual appearance of the widget remains unchanged until DXmCSTextEnableRedisplay is called. This allows an application to make multiple changes to the widget without causing intermediate visual updates.

DXmCSTextEnableRedisplay

Forces the visual update of the DXmCSText widget.

Format

```
void DXmCSTextEnableRedisplay(widget)
        Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextEnableRedisplay is used in conjunction with DXmCSTextDisableRedisplay, which suppresses visual update of DXmCSText. When DXmCSTextEnableRedisplay is called, it determines if any visual attributes have been set or modified for the specified widget since DXmCSTextDisableRedisplay was called. If so, it forces the widget to update its visual display for all of the intervening changes. Any subsequent changes that affect visual appearance cause the widget to update its visual display.

DXmCSTextGetEditable

Indicates whether the text in the compound string text widget can be edited by the user.

Format

```
Boolean DXmCSTextGetEditable(widget)
         Widget  widget;
```

Return Value

A Boolean value that, when True, indicates that the user can edit the text in the compound string text widget. When False, the user cannot edit the text.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetEditable routine indicates, by returning True or False, whether the text in the compound string text widget can be edited by the user. See the DXmCreateCSText and DXmCSTextSetEditable routines for related information.

DXmCSTextGetInsertionPosition

Retrieves (returns) the logical position of the insertion cursor.

Format

```
DXmCSTextPosition DXmCSTextGetInsertionPosition(widget)
                  Widget  widget;
```

Return Value

An integer that represents the logical position of the cursor, by number of characters from the beginning of the text buffer.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetInsertionPosition routine retrieves the logical position of the insertion cursor by returning an integer that indicates the distance (in number of characters) between the cursor and the beginning of the text buffer. See the DXmCSTextSetInsertionPosition for related information.

DXmCSTextGetLastPosition

Retrieves (returns) the logical position of the last character in the text.

Format

```
DXmCSTextPosition DXmCSTextGetLastPosition(widget)
                  Widget  widget;
```

Return Value

An integer that represents the logical position of the last character in the text buffer.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetLastPosition routine retrieves the logical position of the last character in the text by returning an integer. The position may correspond to a line feed.

DXmCSTextGetMaxLength

Retrieves (returns) current maximum allowable length of the text in the compound string text widget.

Format

```
int DXmCSTextGetMaxLength(widget)
     Widget  widget;
```

Return Value

An integer that represents the maximum allowable length, in characters, of the text in the compound string text widget.

Argument

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetMaxLength routine retrieves the current maximum allowable length of the text in the compound string text widget by returning an integer. See the DXmCreateCSText and DXmCSTextSetMaxLength routines for related information.

DXmCSTextGetSelection

Retrieves the text selected in the compound string text widget.

Format

```
XmString DXmCSTextGetSelection(widget)
          Widget  widget;
```

Return Value

A pointer to the selected text compound string.

Argument

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetSelection routine retrieves the text selected (highlighted) in the compound string text widget. It returns a Null pointer if no text is selected in the widget. The application is responsible for freeing the storage associated with the text by calling the Intrinsic routine XtStringFree.

See the DXmCreateCSText, DXmCSTextSetSelection, DXmCSTextClearSelection, DXmCSTextCut, DXmCSTextCopy, and DXmCSTextPaste routines for related information. See also the *X Window System Toolkit* manual for more information about using the Intrinsic routines.

DXmCSTextGetSelectionInfo

Retrieves (returns) the left and right positions of the currently selected text.

Format

```
Boolean DXmCSTextGetSelectionInfo (widget, left, right)
         Widget             widget;
         DXmCSTextPosition  *left;
         DXmCSTextPosition  *right;
```

Return Value

A Boolean value that, when False, indicates that there is no currently selected text; otherwise, the value is True.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

left
The left logical position corresponding to the selected text.

right
The right logical position corresponding to the selected text.

Description

The DXmCSTextGetSelectionInfo routine, when True, returns the left and right logical positions corresponding to the currently selected (highlighted) text. If no text is selected, this routine returns False.

DXmCSTextGetString

Retrieves all the text from the compound string text widget.

Format

```
XmString DXmCSTextGetString(widget)
          Widget  widget;
```

Return Value

A pointer to a compound string holding all the current text in the compound string text widget.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetString routine retrieves the current compound string from the compound string text widget. The application is responsible for freeing the storage associated with the compound string by calling the Intrinsic routine XtStringFree. See the DXmCreateCSText, DXmCSTextSetString, and DXmCSTextReplace routines for related information. See also the *X Window System Toolkit* manual for more information about using the Intrinsic routines.

DXmCSTextGetTopPosition

Retrieves (returns) the logical position of the first character in the displayed text.

Format

```
DXmCSTextPosition DXmCSTextGetTopPosition(widget)
                  Widget  widget
```

Return Value

An integer that represents the logical position of the first character in the text buffer.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextGetTopPosition routine returns the logical position of the first character in the displayed text. If the direction is left to right, the first character is at the top left of the display. If the direction is right to left the first character is at the top right of the display.

DXmCSTextHasSelection

Indicates whether the compound string text widget currently owns the primary selection.

Format

```
Boolean DXmCSTextHasSelection(widget)
         Widget  widget
```

Return Value

A Boolean value that, when True, indicates that the widget currently owns the primary selection.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextHasSelection routine returns True if the compound string text widget currently owns the primary selection and False if it does not.

DXmCSTextHorizontalScroll

Scrolls text horizontally.

Format

```
void DXmCSTextHorizontalScroll(widget, n)
      Widget  widget
      int     n
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

n
An integer that represents the number of pixels to scroll to the left or right. A positive value means scroll right; a negative value means scroll left.

Description

The DXmCSTextHorizontalScroll routine scrolls text horizontally (left or right) by the given number of pixels.

DXmCSTextInsert

Inserts new text into the compound string text widget.

Format

```
DXmCSTextStatus DXmCSTextInsert(widget, pos, value)
                Widget             widget
                DXmCSTextPosition  pos;
                XmString           value;
```

Return Value

A value that represents the status (success or failure) of the insertion operation, as follows:

Value	Description
DXmCSTextStatusEditDone	Text was inserted successfully.
DXmCSTextStatusEditError	Text was not inserted (failure).

Arguments

widget
The identifier (widget ID) of the compound string text widget.

pos
The logical position at which to insert the new text.

value
The compound string representation of the text.

Description

The DXmCSTextInsert routine inserts new text into the compound string text widget at the specified logical position, returning the success or failure of that operation.

DXmCSTextNumLines

Retrieves (returns) the number of visible lines in the compound string text widget.

Format

```
int DXmCSTextNumLines(widget)
     Widget  widget
```

Return Value

An integer that represents the number of visible lines.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextNumLines routine returns an integer that represents the number of lines in the compound string text widget that are visible to the user.

DXmCSTextPaste

Pastes the data from the clipboard into the text at the current cursor position.

Format

```
Boolean DXmCSTextPaste(widget)
         Widget  widget
```

Return Value

A Boolean value that, when True, indicates that the data was pasted successfully from the clipboard into the text.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextPaste routine pastes the data from the clipboard into the text at the current cursor position, returning the success or failure of that operation. (See the *OSF/Motif Programmer's Reference* and the *OSF/Motif Programmer's Guide* for more information about the clipboard.) See the DXmCreateCSText, DXmCSTextCopy, DXmCSTextCut, and DXmCSTextRemove routines for related information.

DXmCSTextPosToXY

Retrieves (returns) the *x* and *y* position of a specified character in the text.

Format

```
Boolean DXmCSTextPosToXY(widget, position, x, y)
         Widget             widget;
         DXmCSTextPosition  position;
         Position           *x;
         Position           *y;
```

Return Value

A Boolean function that, when True, returns the *x* and *y* position of the character in text. When False, no *x* and *y* position is returned.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

position
Specifies the position in the text of the character for which the x- and y-coordinates will be returned.

x
A pointer to the x-coordinate relative to the upper left corner of the widget. (Valid only when the routine returns True.)

y
A pointer to the y-coordinate relative to the upper left corner of the widget. (Valid only when the routine returns True.)

Description

The DXmCSTextPosToXY routine converts the logical position of a specified character in the text to the corresponding x- and y-coordinates. See the DXmCSTextXYToPos routine for related information.

DXmCSTextRemove

Removes the currently selected (highlighted) text.

Format

```
Boolean DXmCSTextRemove(widget)
         Widget  widget
```

Return Value

A Boolean value that, when True, indicates that the text was removed successfully.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

Description

The DXmCSTextRemove routine removes the currently selected (highlighted) text, returning the success or failure of that operation. See the DXmCreateCSText, DXmCSTextCopy, DXmCSTextCut, and DXmCSTextPaste routines for related information.

DXmCSTextReplace

Replaces a specified segment of text in a compound string text widget.

Format

```
DXmCSTextStatus DXmCSTextReplace(widget, from_pos, to_pos, value)
                Widget             widget;
                DXmCSTextPosition  from_pos;
                DXmCSTextPosition  to_pos;
                XmString           value;
```

Return Value

A value that represents the status (success or failure) of the replace operation, as follows:

Value	Description
DXmCSTextStatusEditDone	Text replaced successfully.
DXmCSTextStatusEditError	Text not replaced (failure).

Arguments

widget
The identifier (widget ID) of the compound string text widget.

from_pos
Represents the first character position of the text being replaced.

to_pos
Represents the last character position of the text being replaced.

value
Replacement text for part of the current text in the compound string text widget, or additional text to be inserted into the compound string text widget.

Description

The DXmCSTextReplace routine replaces part of the text in the compound string text widget. Within the widget, positions are numbered starting at 0 and increasing sequentially. For example, to replace the second and third characters in the text, **from_pos** should be 1 and **to_pos** should be 3. To insert text after the fourth character, **from_pos** and **to_pos** should both be 4. See the DXmCreateCSText, DXmCSTextSetString, and DXmCSTextGetString routines for related information.

DXmCSTextSetAddMode

Controls whether the CSText widget is in Add Mode.

Format

```
void DXmCSTextSetAddMode(widget,state)
      Widget   widget;
      Boolean  state;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

state
A Boolean function that, when True, turns on Add Mode and, when False, turns off Add Mode.

Description

The DXmCSTextSetAddMode routine controls whether the CSText widget is in Add Mode, which, when True, allows the user to move the insertion cursor without affecting the primary selection.

DXmCSTextSetEditable

Specifies whether the text in the widget can be edited by the user.

Format

```
void DXmCSTextSetEditable(widget, editable)
     Widget   widget;
     Boolean  editable;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

editable
A Boolean value that, when True, indicates that the user can edit the text in the compound string text widget. When False, the user cannot edit the text.

Description

The DXmCSTextSetEditable specifies whether text in the compound string text widget can be edited by the user, returning either True or False to indicate the permission state. See the DXmCreateCSText and DXmCSTextGetEditable routines for related information.

DXmCSTextSetHighlight

Changes the highlighting of the compound string text located between two specified logical positions.

Format

```
void DXmCSTextSetHighlight(widget, left, right, mode)
      Widget             widget;
      DXmCSTextPosition  left;
      DXmCSTextPosition  right;
      XmHighlightMode    mode;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

left
The left boundary of the text to be highlighted.

right
The right boundary of the text to be highlighted.

mode
One of the following types of highlighting:

Value	Description
XmHIGHLIGHT_NORMAL	Removes highlighting
XmHIGHLIGHT_SELECTED	Uses reverse video
XmHIGHLIGHT_SECONDARY_SELECTED	Uses underlining

Description

The DXmCSTextSetHighlight routine changes the highlighting of the compound string text located between two specified positions, by removing the highlighting or by implementing underlining or reverse video. This visual change in the display does not affect (set) the selections.

DXmCSTextSetInsertionPosition

Sets the insertion cursor to the specified logical position in the text.

Format

```
void DXmCSTextSetInsertionPosition(widget, position)
     Widget             widget;
     DXmCSTextPosition  position;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

position
The position of the insertion cursor.

Description

The DXmCSTextSetInsertionPosition routine sets the insertion cursor to the specified logical position (indicated by number of characters) in the text. See the DXmCSTextGetInsertionPosition for related information.

DXmCSTextSetMaxLength

Sets the maximum allowable length of the text in the compound string text widget.

Format

```
void DXmCSTextSetMaxLength(widget, max_length)
      Widget  widget;
      int     max_length;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

max_length
An integer that represents the maximum length, in characters, of the text in the compound string text widget. This argument sets the **XmNmaxLength** resource used by the DXmCreateCSText routine.

Description

The DXmCSTextSetMaxLength routine sets the maximum allowable length of the text in the compound string text widget. It prohibits the user from entering text longer than this limit. See the related routines DXmCreateCSText and DXmCSTextGetMaxLength.

DXmCSTextSetSelection

Designates the specified text as the current primary selection by highlighting it in the compound string text widget.

Format

```
void DXmCSTextSetSelection(widget, first, last, time)
      Widget              widget;
      DXmCSTextPosition   first;
      DXmCSTextPosition   last;
      Time                time;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

first
Represents the position of the first character in text being selected.

last
Represents the position of the last character in the text being selected.

time
Specifies the time of the event that generated the call to the DXmCSTextSetSelection routine.

Description

The DXmCSTextSetSelection routine designates the specified text as the current primary selection by highlighting it in the compound string text widget. Within the text window, **first** marks the position of the first character in the text and **last** marks the last character. The field characters start at 0 and increase sequentially.

See the DXmCreateCSText, DXmCSTextGetSelection, and DXmCSTextClearSelection routines for related information.

DXmCSTextSetString

Replaces all the text in the compound string text widget with new text.

Format

```
void DXmCSTextSetString(widget, value)
      Widget    widget;
      XmString  value;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

value
The text that replaces all text in the current compound string text widget.

Description

The DXmCSTextSetString routine replaces all the text in the compound string text widget with new text specified by the **value** argument. See the DXmCreateCSText and DXmCSTextGetString routines for related information.

DXmCSTextSetTopPosition

Sets the logical position of the first character in the displayed text.

Format

```
void DXmCSTextSetTopPosition(widget, top_position)
     Widget             widget;
     DXmCSTextPosition  top_position;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

top_position
The logical position of the first character in the text.

Description

The DXmCSTextSetTopPosition routine sets the logical position of the first character in the displayed text. If the direction is left to right, the first character is at the top left of the display. If the direction is right to left the first character is at the top right of the display.

DXmCSTextShowPosition

Displays text located at a specified position.

Format

```
void DXmCSTextShowPosition(widget, position)
      Widget             widget;
      DXmCSTextPosition  position;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

position
The logical position of a character in the text that is to be displayed.

Description

The DXmCSTextShowPosition displays the text located at a specified position, which is the logical position of a character in that text.

DXmCSTextVerticalScroll

Scrolls text vertically.

Format

```
void DXmCSTextVerticalScroll(widget, n)
      Widget  widget;
      int     n;
```

Arguments

widget
The identifier (widget ID) of the compound string text widget.

n
An integer that represents the number of lines to scroll up or down. A positive value means scroll up; a negative value means scroll down.

Description

The DXmCSTextVerticalScroll routine scrolls text vertically by the given number of lines. See the DXmCSTextHorizontalScroll routine for related information.

DXmCSTextXYToPos

Retrieves (returns) the position in the text of the character nearest to a specified *x* and *y* position.

Format

```
DXmCSTextPosition DXmCSTextXYToPos(widget, x, y)
                  Widget    widget;
                  Position  x;
                  Position  y;
```

Return Value

An integer that identifies the position of the character in the text. This integer represents the number of characters from the beginning of the text buffer, with 0 indicating the position of the first character.

Arguments

widget
The identifier (widget ID) of the compound string text widget.

x
The x-coordinate relative to the upper left corner of the widget.

y
The y-coordinate relative to the upper left corner of the widget.

Description

The DXmCSTextXYToPos routine returns an integer that identifies the position in the text of the character that is nearest to a specified *x* and *y* position. See the DXmCSTextPosToXY routine for related information.

DXmCvtCStoDDIF

Converts a compound string into a DDIF format string.

Format

```
Opaque DXmCvtCStoDDIF (cs, byte_count, status)
        XmString  cs;
        long      *byte_count;
        long      *status;
```

Return Value

A pointer to the DDIF string.

Arguments

cs
A compound string.

byte_count
The number of bytes in the output length of the DDIF string.

status
The status of the conversion procedure, indicated by one of the following values:

Value	Description
DXmCvtStatusOK	The conversion was successful.
DXmCvtStatusDataLoss	Data was lost during the conversion.
DXmCvtStatusFail	The conversion was not successful.

Description

The DXmCvtCStoDDIF routine converts a compound string into a DDIF format string. The user must free the string by using the Intrinsic routine XtFree. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

DXmCvtCStoFC

Converts a compound string to a file-compatible format string. Currently uses text format.

Format

```
Opaque DXmCvtCStoFC (cs, byte_count, status)
        XmString  cs;
        long      *byte_count;
        long      *status;
```

Return Value

A pointer to the file-compatible format string.

Arguments

cs
A compound string.

byte_count
The number of bytes in the output length of the string.

status
The status of the conversion procedure, indicated by one of the following values:

Value	Description
DXmCvtStatusOK	The conversion was successful.
DXmCvtStatusDataLoss	Data was lost during the conversion.
DXmCvtStatusFail	The conversion was not successful.

Description

The DXmCvtCStoFC routine converts a compound string to a file-compatible format string. It currently uses text format. The user must free the string by using the Intrinsic routine XtFree. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

DXmCvtCStoOS

Converts a compound string to an operating-system specific format. Currently uses text format.

Format

```
Opaque DXmCvtCStoOS (cs, byte_count, status)
        XmString  cs;
        long      *byte_count;
        long      *status;
```

Return Value

The address of the file-compatible format string.

Arguments

cs
A compound string.

byte_count
The number of bytes in the length of the output string.

status

The status of the conversion procedure, indicated by one of the following values:

Value	Description
DXmCvtStatusOK	The conversion was successful.
DXmCvtStatusDataLoss	Data was lost during the conversion.
DXmCvtStatusFail	The conversion was not successful.

Description

The DXmCvtCStoOS routine converts a compound string to an operating-system specific format. It currently uses text format. The user must free the string by using the Intrinsic routine XtFree. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

DXmCvtDDIFtoCS

Converts a string in DDIF format to a compound string.

Format

```
XmString DXmCvtDDIFtoCS (ddif,size,return_status)
          Opaque  ddif;
          long    *size;
          long    *return_status;
```

Return Value

A pointer to the compound string.

Arguments

ddif
A pointer to the DDIF input string.

size
The length of the compound string.

return_status
The status of the conversion procedure, indicated by one of the following values:

Value	Description
DXmCvtStatusOK	The conversion was successful.
DXmCvtStatusDataLoss	Data was lost during the conversion.
DXmCvtStatusFail	The conversion was not successful.

Description

The DXmCvtDDIFtoCS routine converts a string in DDIF format to a compound string. The user must free the string by using the Toolkit routine XmStringFree. (See the *OSF/Motif Programmer's Reference* for more information about using this routine.)

DXmCvtFCtoCS

Converts a string in the file-compatible format to a compound string.

Format

```
XmString DXmCvtFCtoCS (fc, byte_count, status)
          Opaque  fc;
          long    *byte_count;
          long    *status;
```

Return Value

A pointer to the compound string.

Arguments

fc
A pointer to the input string.

byte_count
The number of bytes in the length of the output string.

status
The status of the conversion procedure, indicated by one of the following values:

Value	Description
DXmCvtStatusOK	The conversion was successful.
DXmCvtStatusDataLoss	Data was lost during the conversion.
DXmCvtStatusFail	The conversion was not successful.

Description

The DXmCvtFCtoCS routine converts a string in the file-compatible format to a compound string. The user must free the string by using the Toolkit routine XmStringFree. (See the *OSF/Motif Programmer's Reference* for more information about using this routine.)

DXmCvtOStoCS

Converts a string in the operating-system-specific format to a compound string.

Format

```
XmString DXmCvtOStoCS (os_string, byte_count, status)
          Opaque  os_string;
          long    *byte_count;
          long    *status;
```

Return Value

A pointer to the compound string.

Arguments

os_string
A pointer to the input string.

byte_count
The number of bytes in the length of the output string.

status
The status of the conversion procedure, indicated by one of the following values:

Value	Description
DXmCvtStatusOK	The conversion was successful.
DXmCvtStatusDataLoss	Data was lost during the conversion.
DXmCvtStatusFail	The conversion was not successful.

Description

The DXmCvtOStoCS routine converts a string in the operating-system-specific format to a compound string. The user must free the string by using the Toolkit routine XmStringFree. (See the *OSF/Motif Programmer's Reference* for more information about using this routine.)

DXmDescToNull

Converts an OpenVMS string descriptor to a null-terminated string. This routine is specific to OpenVMS operating systems.

Format

```
char *DXmDescToNull(desc)
       struct dsc$descriptor_s    *desc;
```

Return Value

The address of a null-terminated string that has been allocated a block of storage by the Intrinsic routine XtMalloc. The user must free this string, using the Intrinsic routine XtFree.

Arguments

desc

The address of an OpenVMS string descriptor for the string to be converted.

Description

The DXmDescToNull routine converts an OpenVMS string descriptor to a null terminated string, which must be freed with the Intrinsic routine XtFree. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

DXmDisplayCSMessage

Displays a compound string message.

Format

```
Widget DXmDisplayCSMessage(parent_widget, name,
                           default_position, x, y,
                           style, message_vector,
                           widget, convert_proc,
                           ok_callback, help_callback)
        Widget          parent_widget;
        char            *name;
        Boolean         default_position;
        Position        x;
        Position        y;
        unsigned char   style;
        int             *message_vector;
        Widget          *widget;
        int             *convert_proc;
        XtCallbackList  ok_callback;
        XtCallbackList  help_callback;
```

Return Value

The identifier (widget ID) of the message box widget.

Arguments

parent_widget
The identifier (widget ID) for the parent widget of the created widget.

name
The name of the created widget.

default_position
When True, the *x* and *y* positions of the widget window are ignored in favor of the default, which centers the message box in the window of the parent widget.

x
The placement, in pixels, of the left side of the message box relative to the inner upper left corner of the parent window.

y
The placement, in pixels, of the upper side of the message box relative to the inner upper left corner of the parent window.

style
The style of the dialog box widget used for the message box. The predefined values for this resource are as follows:

Value	Description
XmDIALOG_FULL_APPLICATION_ MODAL	Used for dialogs that must be responded to before other interactions in the same application
XmDIALOG_MODELESS	Modeless type dialog box
XmDIALOG_PRIMARY_ APPLICATION_MODAL	Used when dialogs must be responded to before other interactions in ancestors of the widget
XmDIALOG_SYSTEM_MODAL	Modal type dialog box

The default is XmDIALOG_MODELESS.

message_vector
The message argument vector specifying the compound strings and associated information.

The first longword contains the number of longwords in the message blocks to follow. The first longword in each message block contains a pointer to the compound string. The next word consists of the $FAO parameter count. The remaining longwords in the message block are the $FAO parameters.

In addition to the standard $FAO system service flags, the compound string message routine will accept the new FAO directive ''!CS.'' When used, this directive will insert a compound string itself.

widget
The identifier (widget ID) of an already existing message box widget. The Intrinsic routine XtSetValues is called on this widget to change the text of the message to match the new message. (Note, however, that this call to XtSetValues does not set **default_position**, **x**, **y**, or **style** for the existing message box widget.) If this argument is 0, a new message box widget is created. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

convert_proc
A pointer to a routine that is executed after the message is formatted but before it is displayed.

A pointer to the formatted string is passed to the routine as a parameter. The parameter is a null-terminated character string.

ok_callback
A callback descriptor data structure. The callback is executed when the user clicks on the Acknowledged button. The reason is **XmCR_OK**.

help_callback
A callback descriptor data structure. The callback is executed when the user requests help. The reason returned is **XmCR_HELP**.

Description

The DXmDisplayCSMessage routine accepts an array of compound strings, formats them, and, if necessary, creates a message box. A return of 0 (Null) indicates that the message box could not be created.

DXmDisplayVMSMessage

Accepts and displays an OpenVMS message. (This routine is available only on OpenVMS systems.)

Format

```
Widget DXmDisplayVMSMessage(parent_widget, name,
                            default_position, x, y,
                            style, message_vector,
                            widget, convert_proc,
                            ok_callback, help_callback)
     Widget          parent_widget;
     char            *name;
     Boolean         default_position;
     Position        x;
     Position        y;
     unsigned char   style;
     int             *message_vector;
     Widget          *widget;
     int             *convert_proc;
     XtCallbackList  ok_callback;
     XtCallbackList  help_callback;
```

Return Value

The identifier (widget ID) of the message box widget.

Arguments

parent_widget
The identifier (widget ID) for the parent widget of the created widget.

name
The name of the created widget.

default_position
When True, the *x* and *y* positions of the message box are ignored in favor of the default, which centers the message box in the window of the parent widget.

x
The placement, in pixels, of the left side of the message box relative to the inner upper left corner of the parent window.

y

The placement, in pixels, of the upper side of the message box relative to the inner upper left corner of the parent window.

style

The style of the dialog box widget used for the message box. The predefined values for this resource are as follows:

Value	Description
XmDIALOG_FULL_APPLICATION_MODAL	Used for dialogs that must be responded to before other interactions in the same application
XmDIALOG_MODELESS	Modeless type dialog box
XmDIALOG_PRIMARY_APPLICATION_MODAL	Used when dialogs must be responded to before other interactions in ancestors of the widget
XmDIALOG_SYSTEM_MODAL	Modal type dialog box

The default is XmDIALOG_MODELESS.

message_vector

The message argument vector specifying the message identifier and associated information. This argument is identical to the OpenVMS $PUTMSG system service.

The first longword contains the number of longwords in the message blocks to follow. The first longword in each message block contains a pointer to the OpenVMS message identifier. Message identifiers are passed by value.

If the message is supplied by the application, the next word consists of the $FAO parameter count. The remaining longwords in the message block are the $FAO parameters.

widget

The identifier (widget ID) of an already existing message box widget. The Intrinsic routine XtSetValues is called on this widget to change the text of the message to match the new message. (Note, however, that this call to XtSetValues does not set **default_position**, **x**, **y**, or **style** for the existing message box widget.) If this argument is 0, a new message box widget is created. (See the *X Window System Toolkit* manual for more information about using the Intrinsic routines.)

convert_proc
A pointer to a routine that is executed after the message is formatted but before it is displayed.

A pointer to the formatted string is passed to the routine as an argument. The parameter is a null-terminated character string.

ok_callback
A callback descriptor data structure. The callback is executed when the user clicks on the Acknowledged button. The reason is **XmCR_OK**.

help_callback
A callback descriptor data structure. The callback is executed when the user requests help. The reason returned is **XmCR_HELP**.

Description

The DXmDisplayVMSMessage routine accepts standard OpenVMS message vectors (defined by the $PUTMSG system service), retrieves them, and creates a message box to display the message. A return of 0 (Null) indicates that the message box could not be created.

DXmFindFontFallback

Associates the given XLFD font name with a new XLFD font name.

Format

```
char *DXmFindFontFallback(fontname)
       char  *fontname;
```

Return Value

A pointer to an XLFD font name string.

Arguments

fontname
A pointer to the XLFD font name.

Description

The DXmFindFontFallback routine associates the given XLFD (X Logical Font Description) font name with a new XLFD font name. This routine does not load the generated font, nor guarantee that it will load successfully. The application calling the routine is responsible for freeing the returned font name through the Intrinsic routine XtFree when the font name is no longer required.

See the DXmFindFontFallback routine for related information. See also the *X Window System Toolkit* manual for information about the Intrinsic routine XtFree and the *X Window System* for information about how the X Window System defines valid font names.

DXmFormSpaceButtonsEqually

Sets push buttons in a form widget to be equally spaced and sized.

Format

```
void DXmFormSpaceButtonsEqually (parent, widget_list, num_widgets)
      Widget    parent;
      Widget    *widget_list;
      Cardinal  num_widgets;
```

Arguments

parent
The identifier (widget ID) of the form widget containing the buttons.

widget_list
Array of widget IDs of the push buttons to be changed.

Note that you must list the widget IDs in the same order as they are displayed in the dialog box and that you should not specify left or right attachments for the push buttons.

num_widgets
The number of widgets in the widget_list.

Description

The DXmFormSpaceButtonsEqually routine enables you to set the push buttons in a form widget to be equally spaced and sized. Your application must call this routine after the dialog box is managed.

DXmGetLocaleString

Provides locale-sensitive Motif compound string version of Toolkit ASCII default values.

Format

```
XmString DXmGetLocaleString(context, string, word_type)
          I18nContext   context;
          char          *string;
          I18nWordType  word_type;
```

Return Value

A pointer to the Motif compound string containing the value.

Arguments

context
Reserved for future use. (Null value passed.)

string
Null-terminated string specifying the ASCII string value whose locale-sensitive version is to be retrieved (for example, the string "OK").

word_type
Specifies the type (part of speech) of a word in the string, as indicated by one of the following values:

Value	Description
I18NADJECTIVE	The word is an adjective.
I18NADVERB	The word is an adverb.
I18NNOUN	The word is a noun.
I18NVERB	The word is a verb.

Description

The DXmGetLocaleString routine provides a locale-sensitive Motif compound string version of Toolkit ASCII default values, such as "OK" or "Cancel". You can specify the part of speech (adjective, adverb, verb, or noun) associated with each word in the string.

DXmHelpOnContext

Enters the application into context-sensitive help mode and either calls a help callback (if one exists) or searches for one in the widget's parent.

Format

```
void DXmHelpOnContext (widget, confine)
      Widget   widget;
      Boolean  confine;
```

Arguments

widget

The identifier of the widget (widget ID) to pass to the XmTrackingLocate routine.

confine

A Boolean value passed to XmTrackingLocate. When True, confines the help cursor (?) to the widget specified in the **widget** argument.

Description

The DXmHelpOnContext routine enters context-sensitive help mode, changing the cursor to the help cursor (?) and then waiting for an MB1 click on a widget. If the selected widget has a help callback, that help callback is called; otherwise, the widget's parent is checked for a help callback.

Note that a Null callback structure pointer is passed to the help callback. Therefore, help callbacks in your application should test the value of this pointer before they attempt to use it.

For related information, refer to the description of the XmTrackingLocate routine in the *OSF/Motif Programmer's Reference*.

DXmHelpSystemClose

Format

```
void DXmHelpSystemClose(help_context, routine, tag)
      Opaque  help_context;
      void    ((*routine )()),
      Opaque  tag );
```

Arguments

help_context

Used by the DECwindows Motif Help System to pass required help information from one routine to another.

routine

Used for error processing. If an error occurs within the DECwindows Motif Help System and it cannot be processed by Bookreader, the DECwindows Motif Help System calls the error processing routine you have included in your application and passes in an integer, or **status**, to indicate the status of the error processing operation, as follows:

Value	Description
2	The DECwindows Motif Help System could not translate a specified value into a valid file specification.

Note the following:

- The **status** argument is a *system-supplied* argument that supplements the **routine** and **tag** arguments you have already specified in the application.
- If an error cannot be processed by Bookreader and you have not specified a supplementary error handling routine, your system will generate an access violation error message.

tag

The argument you supply for the previously described **routine** argument. Associating a tag with the routine enables you to more easily determine where errors occur.

DECwindows Toolkit Extensions
DXmHelpSystemClose

Description

Within the DECwindows Motif Help System, the DXmHelpSystemClose routine closes all of the remaining topic and navigation windows in the Bookreader. You should call this routine when your application is closing all of its other windows.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the DECwindows Motif Help System. See also the DXmHelpSystemDisplay and DXmHelpSystemOpen routines for related information.

DXmHelpSystemDisplay

Displays a topic or directory of the help file in Bookreader. (Note that Bookreader is not available on Windows NT systems.)

Format

```
void DXmHelpSystemDisplay(help_context, help_file, keyword, name, routine, tag)
      Opaque  help_context;
      char    *help_file;
      char    *keyword;
      char    *name;
      void    ((*routine )()),
      Opaque  tag );
```

Arguments

help_context

Used by the DECwindows Motif Help System to pass the required help information from one routine to another.

help_file

Optional. The file name of the help file. If Null or "" is supplied, Bookreader will use the help file name passed in to the DXmHelpSystemOpen routine; otherwise a new file name can be passed.

keyword

One of two values: ‘‘topic’’ or ‘‘dir’’. Pointers to these strings can be passed instead.

name

The topic symbol from the help callback if ‘‘topic’’ is used for the keyword. If ‘‘dir’’ is used, then **name** refers to the name of the Bookreader directory that should be opened, such as "Contents" or "Index". Either specific strings or pointers to strings may be passed in.

routine

Used for error processing. If an error occurs within the DECwindows Motif Help System and it cannot be processed by Bookreader, the DECwindows Motif Help System calls the error processing routine you have included in your application and passes in an integer, or **status**, to indicate the status of the error processing operation, as follows:

Value	Description
2	The DECwindows Motif Help System could not translate a specified value into a valid file specification.

Note the following:

- The **status** argument is a *system-supplied* argument that supplements the **routine** and **tag** arguments you have already specified in the application.
- If an error cannot be processed by Bookreader and you have not specified a supplementary error handling routine, your system will generate an access violation error message.

tag

The argument you supply to the previously described **routine** argument. Associating a tag with the routine enables you to more easily determine where errors occur.

Description

The DXmHelpSystemDisplay routine (which you use instead of a call to the help widget) displays a topic or directory of the help file in Bookreader, using the DECwindows Motif Help System. This routine can also be used to display topics or directories of books other than the ones passed in to DXmHelpSystemOpen.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the DECwindows Motif Help System. See also the DXmHelpSystemDisplay and DXmHelpSystemOpen routines for related information.

DXmHelpSystemOpen

Prepares to display a topic from the specified help file name. (Note that Bookreader is not available on Windows NT systems.)

Format

```
void DXmHelpSystemOpen(help_context, main_window, help_file, routine, tag)
      Opaque  *help_context;
      Widget  main_window;
      char    *help_file;
      void    ((*routine )()),
      Opaque  tag);
```

Arguments

help_context

Used by the DECwindows Motif Help System to pass required help information from one routine to another. Note that **help_context** must be defined globally; the DXmHelpSystemDisplay and DXmHelpSystemClose routines subsequently use the address initially set for this argument.

main_window

The identifier (ID) for the main window of the application.

help_file

The file name of the help file. The default specifications are as follows:

Default on OpenVMS Systems	Default on UNIX Systems	Description
SYS$HELP:	/usr/lib/X11/help	Directory location
.DECW$BOOK	.decw_book	File extension

You can also provide your own location or extension along with the help file name.

Note that **help_file** is required for this routine but is optional for DXmHelpSystemDisplay.

routine

Used for error processing. If an error occurs within the DECwindows Motif Help System and it cannot be processed by Bookreader, the DECwindows Motif Help System calls the error processing routine you have included in your application and passes in an integer, or **status**, to indicate the status of the error processing operation, as follows:

Value	Description
2	The DECwindows Motif Help System could not translate a specified value into a valid file specification.

Note the following:

- The **status** argument is a *system-supplied* argument that supplements the **routine** and **tag** arguments you have already specified in the application.
- If an error cannot be processed by Bookreader and you have not specified a supplementary error handling routine, your system will generate an access violation error message.

tag

The argument you supply to the previously described **routine** argument. Associating a tag with the routine enables you to more easily determine where errors occur.

Description

Within the DECwindows Motif Help System, the DXmHelpSystemOpen routine prepares to display a topic from the help file name passed in. You should make the call to this routine just before the main loop of your application.

See the *DECwindows Motif Guide to Application Programming* for a complete description of the DECwindows Motif Help System. See also the DXmHelpSystemClose and DXmHelpSystemDisplay routines for related information.

DXmInitialize

Prepares an application to use MRM widget-fetching facilities for the Digital supplied (DXm) Motif widgets.

Format

```
void DXmInitialize()
```

Description

The DXmInitialize routine performs a function similar to the MrmInitialize function, but for the DXm widgets. Calling DXmInitialize is equivalent to calling MrmRegisterClass for each of the DXm widgets (DXmColorMix, DXmHelp, DXmPrint, DXmCSText, and DXmSVN).

DXmLoadQueryFont

Loads the font specified by the font name or generates a new font name.

Format

```
XFontStruct  *DXmLoadQueryFont (d, fontname)
               Display  *d;
               char     *fontname;
```

Return Value

A pointer to the XFontStruct of the font (if a font is successfully loaded). If a font cannot be loaded, Null is returned.

Arguments

d
The pointer to the display on which the font will be loaded.

fontname
A pointer to the XLFD font name used to load a font.

Description

The DXmLoadQueryFont routine loads the font specified by the font name or, if the font fails to load, generates a new font name, and loads that font instead.

See the *X Window System* for information about how the X Window System defines valid font names. See also the DXmFindFontFallback routine.

DXmNumChildren

Returns the number of children that belong to the widget.

Format

```
Cardinal DXmNumChildren(widget)
          Widget  widget;
```

Return Value

The number of children (a positive integer).

Arguments

widget
The identifier (widget ID) of the widget.

Description

The DXmNumChildren routine returns a positive integer that represents the number of children that belong to the widget.

DXmPrintWgtAugmentList

Defines additional print formats and lets you add new options to the print widget option menus.

Format

```
unsigned long int DXmPrintWgtAugmentList(pw,list,data)
                  Widget     pw;
                  int        list;
                  XtPointer  data;
```

Return Value

An integer that represents either the number of print formats known to the print widget or the identifier of a new option added to the print widget option menu, depending on the operation performed. A Null return indicates that the addition of a print format or option was unsuccessful.

Arguments

pw
The print widget.

list
The print formats or options, indicated by the following values:

Value	Description
DXmFILE_BURST_SHEET	Specifies the availability and frequency of printed File Burst Sheets
DXmFILE_END_SHEET	Specifies the availability and frequency of printed File End Sheets
DXmFILE_START_SHEET	Specifies the availability and frequency of printed File Start Sheets
DXmINPUT_TRAY	Specifies the input tray holding the sheet to be used for the print job
DXmMESSAGE_LOG	Specifies what is to be done with error messages generated during the print operation

Value	Description
DXmOUTPUT_TRAY	Specifies the output tray in which the sheet is to be placed
DXmPAGE_SIZE	Specifies the size of the print image
DXmPRINT_FORMAT	Specifies the print format
DXmSHEET_SIZE	Specifies the size of the sheet
DXmSIDES	Specifies how many sides of the sheet will be printed

data

One of the following data structures:

- DXmPrintFormatStruct, which has the following structure:

```
typedef struct _DXmPrintFormatStruct
{
    XmString    ui_string;  ❶
    XmString    os_string;  ❷
    XmString    var_string; ❸
} DXmPrintFormatStruct;
```

❶ The label displayed in the user interface

❷ Specifies a string that identifies the print format for the operating system

❸ The OpenVMS logical or UNIX or Windows NT environment variable that identifies the print format

- DXmPrintOptionMenuStruct, which has the following structure:

```
typedef struct _DXmPrintOptionMenuStruct
{
    XmString    ui_string;  ❶
    XmString    os_string;  ❷
} DXmPrintOptionMenuStruct;
```

❶ The label displayed in the user interface

❷ Specifies a string that identifies the print option for the operating system

Description

The DXmPrintWgtAugmentList routine defines additional print formats and lets you add new options to the print widget option menus. Note that the return value (an integer) is not required when adding print formats. However, the Intrinsics routines XtGetValues and XtSetValues use that integer when subsequent calls are made on the option menus.

See the *X Window System Toolkit* manual for more information about the Intrinsic routines and the *DECwindows Motif Guide to Application Programming* for a complete description of the print widget.

See also the following routines for related information:

- DXmCreatePrintBox
- DXmCreatePrintDialog
- DXmPrintWgtPrintJob

DXmPrintWgtPrintJob

Submits files to the print service.

Format

```
unsigned long int DXmPrintWgtPrintJob(print_widget, filenames,
                                      filename_count)
                  Widget         print_widget;
                  XmStringTable  filenames;
                  int            filename_count;
```

Return Value

An integer that indicates the success or failure of the print job, as follows:

Value	Description
0	Print job failed.
1	Print job was successful.

Arguments

print_widget
The identifier (widget ID) of the print widget.

filenames
The file names to be printed.

filename_count
An integer that represents the number of file names.

Description

The DXmPrintWgtPrintJob routine completes the submission of the identified files to the appropriate print service and returns the success of the print job to the application.

DXmSvnAddEntries

Opens new entries in the structure.

Format

```
void DXmSvnAddEntries(widget, after, count, level, tags, index)
      Widget   widget;
      int      after;
      int      count;
      int      level;
      int*     tags[];
      Boolean  index;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

after
An integer that represents the entry number that the newly opened positions are inserted after. (If you want to add an entry before number 1, then set this argument to 0.)

count
An integer that represents the number of entries to add to the array.

level
An integer that represents the level number of the entries being added. A level number of 0 indicates top-level entries.

tags
A pointer to an array of longwords that contain the tag values for each entry being added. If the tag values are not being used, a Null pointer may be passed. This value can also be changed with either the DXmSvnSetEntry or DXmSvnSetEntryTag routine call.

index
A Boolean value that, when True, indicates that all of the entries being added should appear in the scroll index window when the user drags the slider. This value can be changed with either the DXmSvnSetEntry or DXmSvnSetEntryIndexWindow routine. Note that the index window is used only when the DXmNliveScrolling resource is set to False.

Description

The DXmSvnAddEntries routine enables an application to process the expansion of an entry. The value supplied for **count** determines how many entries the application should open. The actual text of the entry is not passed.

The first time an entry is displayed to the user, the SVN widget issues the DXmSvnNgetEntryCallback to obtain information about the entry. If the user never positions the window in an area where a particular entry is visible, the callback will never be made. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnAutoScrollCheck

Controls whether the DXmSvnAutoScrollDisplay routine should be called.

Format

```
int DXmSvnAutoScrollCheck(widget, x, y)
     Widget  widget;
     int     x;
     int     y;
```

Return Value

An integer that indicates whether the DXmSvnAutoScrollDisplay routine will be called, as follows:

Value	Description
0 (False)	DXmSvnAutoScrollDisplay will not be called.
1 (True)	DXmSvnAutoScrollDisplay will be called.

Arguments

widget
The identifier (widget ID) of the SVN widget.

x
An integer that represents the *x* location of the cursor relative to the upper left corner of the SVN window.

y
An integer that represents the *y* location of the cursor relative to the upper left corner of the SVN window.

Description

The DXmSvnAutoScrollCheck routine determines whether the cursor is currently in an area where automatic scrolling (initiated when the user moves an entry outside the window) can occur. This routine returns True if the cursor is in such an area *and* if there are no entries displayed in that direction. The application should call the DXmSvnAutoScrollDisplay routine if this routine returns True.

Use this routine if your application controls the dragging operations (using the DXmSvnSetApplDragging routine). See the DXmSvnSetApplDragging and DXmSvnAutoScrollDisplay routines for related information.

DXmSvnAutoScrollDisplay

Automatically scrolls the screen one unit based on the cursor position.

Format

```
void DXmSvnAutoScrollDisplay(widget, x, y)
      Widget  widget;
      int     x;
      int     y;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

x
An integer that represents the current *x* location of the cursor.

y
An integer that represents the current *y* location of the cursor.

Description

The DXmSvnAutoScrollDisplay routine automatically scrolls the screen one unit based on the current cursor location. If the cursor is not in a location that permits automatic scrolling, this call is ignored.

Use this routine if your application controls the dragging operations (using the DXmSvnSetApplDragging routine). See the DXmSvnSetApplDragging and DXmSvnAutoScrollCheck routines for related information.

DXmSvnClearHighlight

Removes (clears) highlighting from a specified entry.

Format

```
void DXmSvnClearHighlight(widget, entry)
      Widget  widget;
      int     entry;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry that will no longer be highlighted.

Description

The DXmSvnClearHighlight routine removes (clears) highlighting from a specified entry in the structure. If the entry is not currently highlighted, this call is ignored. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnClearHighlighting

Removes (clears) all highlighting from the entries.

Format

```
void DXmSvnClearHighlighting(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnClearHighlighting routine removes (clears) all highlighting from all entries in the structure. If there are currently no highlighted entries, this call is ignored. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnClearSelection

Clears (cancels) the selection of a specified entry.

Format

```
void DXmSvnClearSelection(widget, entry)
      Widget  widget;
      int     entry;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry that will no longer be selected.

Description

The DXmSvnClearSelection routine clears, or cancels, the selection of a particular entry in the structure. If the entry is not currently selected, the call is ignored. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnClearSelections

Clears all selections.

Format

```
void DXmSvnClearSelections(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnClearSelections routine clears (cancels) all of the entries in the structure. If no entries are currently selected, the call is ignored. (Use the DXmSvnGetSelections, DXmSvnClearSelections, and DXmSvnShowSelections routines when getting, clearing, and showing selections.) Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnDeleteEntries

Deletes entries from the structure.

Format

```
void DXmSvnDeleteEntries(widget, after, count)
      Widget  widget;
      int     after;
      int     count;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

after
An integer that represents the entry number after which a specified number of entries are to be deleted.

count
An integer that represents the number of entries to delete from the array.

Description

The DXmSvnDeleteEntries routine enables an application to process the removal (collapse) of an entry. This routine specifies how many entries to delete and after which entry those deletions should begin. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnDisableDisplay

Temporarily disables the SVN widget from generating a display.

Format

```
void DXmSvnDisableDisplay(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnDisableDisplay routine allows your application to make changes to the SVN widget without the user making additional changes. For example, the DXmSvnDisableDisplay routine must be called when the user has selected to either expand or collapse an entry in the structure. No further user actions will be processed (because the SVN widget is temporarily disabled) until that expand or collapse operation has been completed and the DXmSvnEnableDisplay routine has been called (which re-enables the SVN widget and makes the changes visible to the user).

The kinds of changes that can be implemented while the widget is disabled include the following:

- Calling the DXmSvnSelectAll or DXmSvnClearSelections routine to change selections
- Calling the DXmSvnAddEntries or DXmSvnDeleteEntries routine to change the underlying structure
- Changing the appearance of an entry (adding highlighting or sensitivity, for example)

This routine is required only when the changes are not in response to an SVN callback. (The SVN widget automatically disables the SVN widget prior to issuing the callback and automatically enables the widget upon return.)

See the DXmSvnEnableDisplay routine for related information.

DXmSvnEnableDisplay

Reenables the SVN widget.

Format

```
void DXmSvnEnableDisplay(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnEnableDisplay routine re-enables the SVN widget, which then determines whether to update the display (make changes visible to the user) based on activities recorded while the SVN widget was in the disabled mode. These activities can include changes to selections, the addition or deletion of entries, or changes to the appearance of entries.

Your application must call this routine following a call to the DXmSvnDisableDisplay routine. (See the description of that routine for related information.)

DXmSvnFlushEntry

Displays the specified entry on the screen if it is the next logically displayed entry.

Format

```
void DXmSvnFlushEntry(widget, entry)
      Widget  widget;
      int     entry;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the specified entry.

Description

The DXmSvnFlushEntry routine displays (or ‘‘flushes’’) an entry on the screen (in outline mode only) if the entry number passed to the routine is one integer greater than the entry number of the last displayed entry *and* there is enough space for the entry to fit in the display. If there is not enough blank space for the entry to fit, DXmSvnFlushEntry permits the SVN widget to scroll to the entry.

DXmSvnGetComponentNumber

Retrieves (returns) the component number containing a supplied tag value.

Format

```
int DXmSvnGetComponentNumber(widget, tag)
     Widget     widget;
     XtPointer  tag;
```

Return Value

An integer that represents the component number. (Returns 0 if the supplied tag value is not found.)

Arguments

widget
The identifier (widget ID) of the SVN widget.

tag
A value that represents the component tag.

Description

The DXmSvnGetComponentNumber routine is called whenever the application requires the component number that is associated with a particular tag value. If the specified tag value is not found, then a value of 0 is returned.

See the descriptions of the DXmSvnSetComponentTag and DXmSvnInsertComponent routines for information about setting the tag value for a component.

DXmSvnGetComponentTag

Retrieves (returns) a component tag for a specified component number.

Format

```
XtPointer DXmSvnGetComponentTag(widget, comp_number)
     Widget        widget;
     unsigned int  comp_number;
```

Return Value

An integer that represents the tag value (returns 0 if the component tag has not been set).

Arguments

widget
The identifier (widget ID) of the SVN widget.

comp_number
An integer that represents the component number.

Description

The DXmSvnGetComponentTag routine retrieves (returns) a component tag for a specified component number whenever the application requires that information. You can set the value for this tag by using the DXmSvnInsertComponent or DXmSvnSetComponentTag routine.

DXmSvnGetComponentText

Retrieves (returns) the address of a compound string.

Format

```
XmString DXmSvnGetComponentText(widget, entry_number, comp_number)
          Widget  widget;
          int     entry_number;
          int     comp_number;
```

Return Value

A pointer to the compound string.

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry_number
An integer that represents the number of the entry.

comp_number
An integer that represents the number of the component.

Description

The DXmSvnGetComponentText routine returns the address of a compound string that was previously stored from a call to the DXmSvnSetComponentText routine.

DXmSvnGetComponentWidth

Retrieves (returns) the width of a specified component.

Format

```
int DXmSvnGetComponentWidth(widget, comp_number)
     Widget  widget;
     int     comp_number;
```

Return Value

An integer that represents the width of the component.

Arguments

widget
The identifier (widget ID) of the SVN widget.

comp_number
An integer that represents the component number.

Description

The DXmSvnGetComponentWidth routine is called whenever the application requires the component width that is associated with a particular component number. You can set the value for the width by using the DXmSvnInsertComponent or DXmSvnSetComponentWidth routine. Note that the SVN widget automatically increases the width field whenever it encounters a component that is wider than the current width value.

DXmSvnGetDisplayed

Retrieves (returns) displayed entry numbers and related information necessary to draw a corresponding display.

Format

```
void DXmSvnGetDisplayed(widget, entries, tags, ys, len)
      Widget        widget;
      int           *entries[];
      unsigned int  *tags[];
      int           *ys[];
      int           len;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entries
A pointer to an array of integers to receive the entry numbers of the entries being displayed.

tags
A pointer to an array of longwords to receive the application's **entry_tag** value for each entry displayed. If tags are not required, a Null pointer may be passed.

ys
A pointer to an array of longwords to receive the y-coordinates for each entry displayed. If y-coordinates are not required, a Null pointer can be passed.

len
The number of entries allocated in the provided array.

Description

The DXmSvnGetDisplayed routine returns information about the entries that the SVN widget is currently displaying. This information can then be used to keep a simultaneous display up to date with the SVN widget window (in the case of a dialog box, for example, which might contain totals for the entries being displayed).

The application is responsible for managing the memory used to return this list of entries. As such, note the following:

- At the minimum, the number of entries in the array should be capable of holding at least the number of entries indicated by the value returned from the DXmSvnGetNumDisplayed routine.
- If there are more entries in the array than the application will need, the SVN widget will set the value for those extra entries to 0.
- If the capacity of the arrays passed is less than the number of selected entries, only the number of entries allocated in the provided array (the value for **len**) will be returned.

DXmSvnGetEntryLevel

Retrieves (returns) the level of an entry in the display.

Format

```
int DXmSvnGetEntryLevel(widget, entry)
     Widget  widget;
     int     entry;
```

Return Value

An integer that represents the entry level.

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry number.

Description

The DXmSvnGetEntryLevel routine is called whenever the application requires the level number for a given entry number.

DXmSvnGetEntryNumber

Retrieves (returns) an entry number for a specified entry tag value.

Format

```
int DXmSvnGetEntryNumber(widget, tag)
     Widget        widget;
     XtPointer     tag;
```

Return Value

An integer that represents the entry number (returns 0 if the entry tag value does not correspond to any entry number).

Arguments

widget
The identifier (widget ID) of the SVN widget.

tag
A value that represents the entry tag.

Description

The DXmSvnGetEntryNumber routine retrieves (returns) an entry number for a specified entry tag value whenever the application requires that information. See the DXmSvnAddEntries and DXmSvnSetEntry routines for related information.

DXmSvnGetEntryPosition

Retrieves (returns) the position of an entry in the display.

Format

```
void DXmSvnGetEntryPosition(widget, entry, mode, x, y)
      Widget   widget;
      int      entry;
      Boolean  mode;
      int      *x;
      int      *y;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which position information is required.

mode
A Boolean value that, when True, specifies that position information returned is relative to the upper left corner of the SVN window. When False, the position information returned is internal SVN information only (for column and outline mode) or indicates the position of the entry in the entire tree (tree mode).

x
A pointer to the *x* position in the tree.

y
A pointer to the *y* position in the tree.

Description

The DXmSvnGetEntryPosition routine allows an application to determine the position of entries. The routine interprets the x- and y-coordinates differently, depending on the application's mode:

- Outline and Column mode

  When the application is in outline or column mode (the DXmSvnNdisplayMode resource has been set to either DXmSvnKdisplayOutline or DXmSvnKdisplayColumn) and **mode** for the DXmSvnGetEntryPosition routine has been set to True, the x- and y-coordinates indicate, in pixels, the distance of the displayed entry from the upper left corner of the SVN window.

  However, if the specified entry is *not* visible in the SVN window, the routine returns -1 (an invalid value) for both the x- and y-coordinates, because the SVN widget cannot determine the position of such an entry when the application is in outline or column mode.

  When **mode** is False, the values returned for the x- and y-coordinates are only meaningful internally to the SVN widget.

- Tree mode

  When the application is in tree mode (the DXmSvnNdisplayMode resource is set to DXmSvnKdisplayTree) and the DXmSvnNtreeStyle resource is set to DXmSvnKuserDefinedTree (the application controls the format), the x- and y-coordinates indicate, in pixels, the distance of the displayed entry from the upper left corner of the SVN window, when **mode** is True.

  If the entry is *not* visible in the SVN window, the **mode** is False. However, in tree mode, the returned x- and y-coordinates indicate the position of the entry in the entire tree, even though that position extends beyond the limit of what can currently be displayed in the SVN window.

Note that for position values to be valid, the SVN widget must have been enabled prior to any changes made to the contents of the display.

DXmSvnGetEntrySensitivity

Retrieves (returns) the sensitivity of an entry in the display.

Format

```
int DXmSvnGetEntrySensitivity(widget, entry)
     Widget  widget;
     int     entry;
```

Return Value

An integer that represents the entry sensitivity value, as follows:

Value	Description
0	The entry is not sensitive (the user cannot select it).
1	The entry is sensitive (the user can select it).

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry number.

Description

The DXmSvnGetEntrySensitivity routine returns the sensitivity state (which indicates whether the user can select an entry) whenever the application requires that information for a given entry.

DXmSvnGetEntryTag

Retrieves (returns) an entry's tag for a specified entry number.

Format

```
XtPointer DXmSvnGetEntryTag(widget, entry)
     Widget  widget;
     int     entry;
```

Return Value

Represents the entry tag.

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry number.

Description

The DXmSvnGetEntryTag routine retrieves an entry tag value for a specified entry number. The routine returns a value of 0 if the entry tag value has never been specified for the given entry number in a call to either the DXmSvnAddEntries or DXmSvnSetEntry routine.

DXmSvnGetHighlighted

Retrieves (returns) a list of entries currently highlighted by the user.

Format

```
void DXmSvnGetHighlighted(widget, entries, tags, len)
      Widget        widget;
      int           *entries[];
      unsigned int  tags[];
      int           len;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entries
A pointer to an array of integers to receive the highlighted entry numbers.

tags
A pointer to an array of longwords to receive the application's **entry_tag** value for each highlighted entry. If tags are not required, a null pointer may be passed.

len
An integer that represents the number of entries allocated in the provided array.

Description

The DXmSvnGetHighlighted routine returns a list of entries that are currently highlighted by the user. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay to ensure that the highlighted list is not changing.

The application is responsible for managing the memory used to return this list of highlighted entries. As such, note the following:

- At the minimum, the number of entries in the array should be capable of holding the number of entries indicated by the value returned from the DXmSvnGetNumHighlighted routine.
- If there are more entries in the array than the application will need, the SVN widget will set the value for those extra entries to 0.

- If the capacity of the arrays passed is less than the number of selected entries, only the number of entries allocated in the provided array (the value for **len**) will be returned.

DXmSvnGetNumDisplayed

Retrieves (returns) the number of entries currently being displayed in the window.

Format

```
int DXmSvnGetNumDisplayed(widget)
     Widget  widget;
```

Return Value

An integer that represents the number of entries currently being displayed.

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnGetNumDisplayed routine returns the number of entries currently being displayed in the SVN window. Use this routine to determine the number of entries required in the arrays being passed to the DXmSvnGetDisplayed routine.

Note that this routine is not supported in tree mode (it will return a value of 0).

DXmSvnGetNumHighlighted

Retrieves (returns) the number of highlighted entries.

Format

```
int DXmSvnGetNumHighlighted(widget)
     Widget  widget;
```

Return Value

An integer that represents the number of highlighted entries.

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnGetNumHighlighted routine returns the number of entries currently highlighted by the user. Use this routine to determine the number of entries required in the array being passed to the DXmSvnGetHighlighted routine. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay to ensure that the highlighted list is not changing.

DXmSvnGetNumSelections

Retrieves (returns) the number of selected entries.

Format

```
int DXmSvnGetNumSelections(widget)
     Widget  widget;
```

Return Value

An integer that represents the number of selected entries.

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnGetNumSelections routine returns the number of entries currently selected by the user. Use this routine to determine the number of entries required in the array being passed to the DXmSvnGetSelections routine. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay to ensure that the selected list is not changing.

DXmSvnGetPrimaryWorkWidget

Retrieves (returns) the identifier (widget ID) of the primary SVN work widget.

Format

```
Widget DXmSvnGetPrimaryWorkWidget(widget)
        Widget  widget;
```

Return Value

The identifier (widget ID) of the primary SVN work widget.

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnGetPrimaryWorkWidget routine is a read-only routine that returns the identifier (widget ID) of the primary work widget (window).

DXmSvnGetSecondaryWorkWidget

Retrieves (returns) the identifier (widget ID) of the secondary SVN work widget.

Format

```
Widget DXmSvnGetSecondaryWorkWidget(widget)
        Widget  widget;
```

Return Value

The identifier (widget ID) of the secondary SVN work widget.

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnGetSecondaryWorkWidget routine is a read-only routine that returns the identifier (widget ID) of the secondary work widget (window).

DXmSvnGetSelections

Retrieves (returns) a list of selected entry numbers.

Format

```
void DXmSvnGetSelections(widget, entries, comps, tags, len)
      Widget         widget;
      int            *entries[];
      int            *comps[];
      unsigned int   *tags[];
      int            len;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entries
A pointer to an array of integers to receive the selected entry numbers.

comps
A pointer to an array of integers to receive the component numbers on which the entry was selected. A component number of 0 means that the entry was selected using range selection and not by directly clicking on the entry. If components are not needed, a Null pointer may be passed.

tags
A pointer to an array of longwords to receive the application tag value for each entry selected. If tags are not needed, a Null pointer may be passed.

len
An integer that represents the number of entries allocated in the provided array.

Description

The DXmSvnGetSelections routine returns a list of entries that are currently selected by the user. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay to ensure that the selected list is not changing.

The application is responsible for managing the memory used to return this list of highlighted entries. As such, note the following:

- At the minimum, the number of entries in the array should be capable of holding the number of entries indicated by the value returned from the DXmSvnGetNumHighlighted routine.
- If there are more entries in the array than the application will need, the SVN widget will set the value for those extra entries to 0.
- If the capacity of the arrays passed is less than the number of selected entries, only the number of entries allocated in the provided array (the value for **len**) will be returned.

DXmSvnGetTreePosition

Retrieves (returns) the position of the display window for future use.

Format

```
void DXmSvnGetTreePosition(widget, x, y)
      Widget  widget;
      int     *x;
      int     *y;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

x
A pointer to the *x* position in the tree.

y
A pointer to the *y* position in the tree.

Description

The DXmSvnGetTreePosition routine obtains the upper left coordinate position of the display window that is within the virtual display. It thus provides a way for applications to determine the exact position of the display window so that window can be restored at a later time (using the DXmSvnSetTreePosition routine).

The application can call this routine *only* when the DXmSvnNdisplayMode resource is set to DXmSvnKdisplayTree. Note as well that the x- and y-coordinates returned are only meaningful internally to the SVN widget (they are used by the DXmSvnSetTreePosition routine when the SVN widget restores the display window to its original position).

DXmSvnHideHighlighting

Specifies that the SVN widget show highlighted entries in the default format, without using highlighting graphics.

Format

```
void DXmSvnHideHighlighting(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnHideHighlighting routine specifies that the SVN widget show highlighted entries in the default format, without drawing highlighting graphics around those entries. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay to ensure that the highlighted list is not changing.

To draw highlight graphics around highlighted entries, use the DXmSvnShowHighlighting routine.

DXmSvnHideSelections

Specifies that the SVN widget show selected entries in the default format, without drawing selection graphics on those entries.

Format

```
void DXmSvnHideSelections(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnHideSelections routine specifies that the SVN widget show selected entries in the default format (reverse video in outline mode), without drawing selection graphics on the entries even if the entries are selected. Use this routine if you want to hide selections when the application loses ownership of the global selection. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

To redraw entries with selection graphics, use the DXmSvnShowSelections routine.

DXmSvnHighlightAll

Highlights all entries.

Format

```
void DXmSvnHighlightAll(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnHighlightAll routine allows an application to highlight all entries in the structure using a command from the application's menu. The entries are displayed in a rectangular box (drawn with dashes). Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnHighlightEntry

Highlights a specified entry.

Format

```
void DXmSvnHighlightEntry(widget, entry)
      Widget  widget;
      int     entry;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry to be highlighted.

Description

The DXmSvnHighlightEntry routine designates a particular entry as highlighted. If the entry is already highlighted, this call is ignored. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnInsertComponent

Inserts a new component into each SVN entry.

Format

```
void DXmSvnInsertComponent(widget, comp_number, width, tag)
      Widget        widget;
      int           comp_number;
      Dimension     width;
      unsigned int  tag;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

comp_number
An integer that represents the component number to insert.

width
The width of the component. This value will be inserted into the widget's component width array. You can set or examine this value by using the DXmSvnSetComponentWidth and DXmSvnGetComponentWidth routines.

tag
The tag for the component. This value will be inserted into the widget's component tag array. You can set or examine this value by using the DXmSvnSetComponentTag and DXmSvnGetComponentTag routines.

Description

The DXmSvnInsertComponent routine inserts a new component for each entry currently recognized by the SVN widget. All entries that are currently valid become invalid during this operation.

Note the following:

- A warning message is generated if an attempt is made to insert a component that is not the next in the sequence of components in the entry (for example, you cannot add component 5 if there are only 3 components in the entry; you must add component 4).
- An entry can hold a maximum of 30 components.

DXmSvnInvalidateEntry

Specifies that the SVN widget redisplay (update) an entry.

Format

```
void DXmSvnInvalidateEntry(widget, entry)
      Widget  widget;
      int     entry;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry number to invalidate.

Description

The DXmSvnInvalidateEntry routine allows an application to modify an underlying entry (for example, changing the text or highlighting) by directing the SVN widget to issue the DXmSvnNgetEntryCallback to obtain new information about the specified entry.

Note that if an entry is selected prior to a call to the DXmSvnInvalidateEntry routine, that entry will remain selected after the entry is redisplayed. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnMapPosition

Associates the *x* and *y* position with an entry number.

Format

```
void DXmSvnMapPosition(widget, x, y, entry, comp_number, tag)
      Widget        widget;
      int           x;
      int           y;
      int           *entry;
      int           *comp_number;
      unsigned int  *tag;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

x
An integer that represents the *x* position within the widget's window where the mouse button was pressed or released.

y
An integer that represents the *y* position within the widget's window where the mouse button was pressed or released.

entry
A pointer to the entry number.

comp_number
A pointer to the component number within the entry. If the component number is not needed, a Null pointer may be passed.

tag
A pointer to the application's tag associated with the entry. If the tag value is not required by the application, a Null pointer may be passed.

Description

The DXmSvnMapPosition routine uses a supplied x- and y-coordinates to determine which entry the user was positioned over when the mouse button was pressed or released.

DXmSvnPositionDisplay

Displays a specified entry.

Format

```
int DXmSvnPositionDisplay(widget, entry, position)
     Widget  widget;
     int     entry;
     int     position;
```

Return Value

An integer that returns as 1 (True). The routine returns a value of 0 (False) only when the specified position of the entry is SvnPositionNextPage and the user is already at the bottom of the page (display).

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry.

position
The position of the entry, indicated by one of the following values:

Value	Description
DXmSvnKpositionBottom	The entry is positioned at the bottom of the display.
DXmSvnKpositionTop	The entry is positioned at the top of the display.
DXmSvnKpositionMiddle	The entry is positioned in the middle of the display.
DXmSvnKpositionPreviousPage	The entry is positioned on the page (display) of entries previously displayed.
DXmSvnKpositionNextPage	The entry is positioned on the page (display) of entries to be displayed next.

Description

The DXmSvnPositionDisplay routine ensures that the entries being displayed contain a specified entry. This routine allows the application to position a particular entry at the top, middle, or bottom of the display regardless of the number of entries being displayed.

DXmSvnRemoveComponent

Removes a component from each SVN entry.

Format

```
void DXmSvnRemoveComponent(widget, comp_number)
      Widget  widget;
      int     comp_number;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

comp_number
An integer that represents the component number to be removed.

Description

The DXmSvnRemoveComponent routine removes a new component for any entry currently recognized by the SVN widget. All entries that are currently valid continue to be valid after the component is removed.

Note the following:

- A warning message is generated if an attempt is made to remove a component that is not within the range of the number of components in the entry (for example, you cannot remove component 3 if there are only 2 components in the entry).
- During the remove operation, the corresponding tag and width values for the component being removed are deleted from the widget's component width and tag arrays. Those values are discarded and cannot be retrieved.

DXmSvnSelectAll

Selects all entries.

Format

```
void DXmSvnSelectAll(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnSelectAll routine allows the application to designate all entries in the structure as selected (they are displayed using reverse video) using a command from the application's menu. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnSelectComponent

Selects an entry using a specific component.

Format

```
void DXmSvnSelectComponent(widget, entry, comp_number)
      Widget  widget;
      int     entry;
      int     comp_number;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry to select.

comp_number
An integer that represents the component to select.

Description

The DXmSvnSelectComponent routine designates a specified entry as selected. It differs from the DXmSvnSelectEntry routine because it associates a specific component with the selection. If the entry is already selected with this component, this call has no effect. Use this routine when different selection modes are being used and make the call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnSelectEntry

Selects a specified entry.

Format

```
void DXmSvnSelectEntry(widget, entry)
      Widget  widget;
      int     entry;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry to select.

Description

The DXmSvnSelectEntry routine designates a particular entry as selected. If the entry is already selected, this call is ignored. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnSetApplDragging

Turns application dragging mode on or off in the SVN widget.

Format

```
void DXmSvnSetApplDragging(widget, on_off)
      Widget  widget;
      int     on_off;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

on_off
An integer that sets the application dragging mode to be either on (1) or off (0).

Description

The DXmSvnSetApplDragging routine either turns application dragging mode on or off. When application dragging mode is on (**on_off** is set to 1), the application controls dragging operations by calling the DXmSvnCRDragging callback (when the user presses MB2 and drags the mouse downward) and the DXmSvnCRDraggingEnd callback (when the dragging operation is completed).

When application dragging mode is off (**on_off** is set to 0), the SVN widget controls MB2 dragging operations with DXmSvnNselectionsDraggedCallback.

DXmSvnSetComponentHidden

Controls which modes the SVN widget uses to display a particular component.

Format

```
void DXmSvnSetComponentHidden(widget, entry, comp_number, mode)
      Widget  widget;
      int     entry;
      int     comp_number;
      int     mode;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which information is to be set or changed.

comp_number
An integer that represents the number of the component for which information is to be set or changed. This cannot be a component of the type **Widget**.

mode
The display mode in which this component should not be displayed, indicated by one of the following values:

Value	Description
DXmSvnKdisplayNone	Display in all modes (hidden in none)
DXmSvnKdisplayOutline	Do not display in outline mode
DXmSvnKdisplayTree	Do not display in tree mode
DXmSvnKdisplayAllModes	Do not display in outline, tree, or column mode
DXmSvnKdisplayColumns	Do not display in column mode

Description

The DXmSvnSetComponentHidden routine enables an application to prevent a component (excluding subwidgets) from being displayed in a particular mode, thus making that component ‘‘hidden.’’ By default, components are visible in all modes. However, you can use this routine to prevent a component from displaying in one mode (outline, tree, or column) or all modes.

Using this routine can reduce the amount of information displayed when the application is in tree mode, making the application less complex and easier to use. This routine is also useful if you want your application to display different icons in each of the two modes.

Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnSetComponentPixmap

Adds a pixmap component to an entry.

Format

```
void DXmSvnSetComponentPixmap(widget, entry, comp_number, x, y,
                              pixmap, width, height)
      Widget  widget;
      int     entry;
      int     comp_number;
      int     x;
      int     y;
      Pixmap  pixmap;
      int     width;
      int     height;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry that this component belongs to.

comp_number
An integer that represents the component number within the entry.

x
An integer that represents the *x* offset within the entry where this component should be placed.

y
An integer that represents the *y* offset within the entry where this component should be placed. (Note that if you want the SVN widget to automatically create a layout of the entry for you, set the value of this argument to 0.)

pixmap
The Pixmap value.

width
An integer that represents the width specified when the pixmap was created.

height
An integer that represents the height specified when the pixmap was created.

Description

The DXmSvnSetComponentPixmap routine sets a component of an entry to be a pixmap. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay. If your application passes 0 for the value of the *y* offset, the SVN widget will automatically create a layout of the entry.

DXmSvnSetComponentTag

Sets the tag of a component.

Format

```
void DXmSvnSetComponentTag(widget, comp_number, tag)
      Widget           widget;
      int              comp_number;
      XtPointer        tag;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

comp_number
An integer that represents the component number.

tag
A value that represents the entry tag.

Description

The DXmSvnSetComponentTag routine sets the widget level component tag data. (The SVN widget never examines this tag.) This tag may be lost if the corresponding component is removed with the DXmSvnRemoveComponent routine.

Use the DXmSvnGetComponentTag routine to retrieve the current value of the tag; use the DXmSvnGetComponentNumber routine to return the component number associated with the supplied tag.

DXmSvnSetComponentText

Adds a read-only component string to an entry.

Format

```
void DXmSvnSetComponentText(widget, entry, comp_number,
                            x, y, text, font)
      Widget       widget;
      int          entry;
      int          comp_number;
      int          x;
      int          y;
      XmString     text;
      XmFontList   font;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry that this component belongs to.

comp_number
An integer that represents the component number within the entry.

x
An integer that specifies the *x* position of the component within the entry.

y
An integer that specifies the *y* position of the component within the entry. (Note that if you want the SVN widget to automatically create a layout of the entry for you, set the value of this argument to 0.)

text
A compound string.

font
A valid XmFontList.

Description

The DXmSvnSetComponentText routine sets a component of an entry to be a read-only compound string. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay. If your application passes 0 for the value of the *y* offset, the SVN widget will automatically create a layout of the entry.

DXmSvnSetComponentWidget

Adds an SVN subwidget component to an entry.

Format

```
void DXmSvnSetComponentWidget(widget, entry, comp_number, x, y, subw)
      Widget  widget;
      int     entry;
      int     comp_number;
      int     x;
      int     y;
      Widget  subw;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the entry number that this component belongs to.

comp_number
An integer that represents the component number within the entry.

x
An integer that represents the *x* offset within the entry where this component should be placed.

y
An integer that represents the *y* offset within the entry where this component should be placed. (Note that if you want the SVN widget to automatically create a layout of the entry for you, set the value of this argument to 0.)

subw
The identifier (widget ID) of the subwidget. This subwidget will be a managed child of the SVN widget.

Description

The DXmSvnSetComponentWidget routine sets a component of an entry to be a subwidget, which enables an application to put read-write text or a push button in an entry. The application is responsible for creating this subwidget and destroying it when the entry is deleted. If your application passes 0 for the value of the *y* offset, the SVN widget will automatically create a layout of the entry.

DXmSvnSetComponentWidth

Sets the width of a component.

Format

```
void DXmSvnSetComponentWidth(widget, comp_number, width)
      Widget     widget;
      int        comp_number;
      Dimension  width;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

comp_number
An integer that represents the component number.

width
The width to set.

Return Value

An integer that represents the width of the component.

Description

The DXmSvnSetComponentWidth routine sets the widget level component width data. This width is used only when the widget is displaying in column mode. If a component found in this column is larger than the value currently set, the component width is automatically adjusted (which makes setting the component width optional).

New components inserted with the DXmSvnInsertComponent routine may include the component width as an argument to that routine.

DXmSvnSetEntry

Provides the SVN widget with new entry level information.

Format

```
void DXmSvnSetEntry(widget, entry, width, height, number, sens,
                    tag, index)

      Widget        widget;
      int           entry;
      int           width;
      int           height;
      int           number;
      Boolean       sens;
      unsigned int  tag;
      Boolean       index;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which information will be set or changed.

width
An integer that represents the width of the entry. This value is used to allocate sufficient display space for the entry. If 0 is specified, the SVN widget will calculate the width based on the components and font.

height
An integer that represents the height of the entry. This value is used to allocate sufficient display space for the entry. If 0 is specified, the SVN widget will calculate the height based on the components and font.

number
An integer that represents the number of components in this entry.

sens
A Boolean value that, when True, makes this entry sensitive (the user can select it). When False, the display for that entry will be dimmed to indicate that the entry is not sensitive (the user cannot select it).

tag
The value that the application will associate with the entry. This value is returned in all callbacks that set the **entry_number** field in the DXmSvnCallback structure.

index
A Boolean value that, when True, indicates that all of the entries being added should be displayed in the scroll index window when the user drags the slider. Note that if you set this argument to True, the DXmNliveScrolling resource must be set to False.

Description

The DXmSvnSetEntry routine enables an application to respond to a DXmSvnCRGetEntry callback requesting the information for an entry. You can specify a subset of the information allowed on this call by using the following routines:

- DXmSvnSetEntryNumComponents
- DXmSvnSetEntryTag
- DXmSvnSetEntryIndexWindow
- DXmSvnSetEntrySensitivity
- DXmSvnSetEntry

Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay. After making a call to this routine, your application should call the DXmSvnSetComponentText, DXmSvnSetComponentPixmap, and DXmSvnSetComponentWidget routines to individually supply the component information.

Note that if your application calls the DXmSvnSetEntry routine to modify an existing entry and the number of entries changes, all previous component information is discarded.

DXmSvnSetEntryIndexWindow

Controls whether the SVN widget displays the entry in the index window.

Format

```
void DXmSvnSetEntryIndexWindow(widget, entry, index)
      Widget   widget;
      int      entry;
      Boolean  index;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which information will be set or changed.

index
A Boolean value that, when True, indicates that all of the entries being added will be displayed either in the scroll index window when the user drags the slider (in outline or column mode) or in the Navigation Window (in tree mode).

If you set this argument to True, the DXmNliveScrolling resource must be set to False. Note as well that if the display mode is DXmSvnKdisplayTree and you have set the DXmSvnNTreeIndexAll resource to True, this specification will be overridden.

Description

The DXmSvnSetEntryIndexWindow routine enables an application to respond to a DXmSvnCRGetEntry callback requesting the information for an entry. It provides some of the functions of the DXmSvnSetEntry routine, but only requires a subset of the information.

DXmSvnSetEntryNumComponents

Provides the SVN widget with the number of components for an entry.

Format

```
void DXmSvnSetEntryNumComponents(widget, entry, number)
      Widget  widget;
      int     entry;
      int     number;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which information will be set or changed.

number
An integer that represents the number of components that make up this entry.

Description

The DXmSvnSetEntryNumComponents routine enables an application to respond to a DXmSvnCRGetEntry callback requesting information for an entry. It provides some of the functions of the DXmSvnSetEntry routine, but requires only a subset of the information.

Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay. After making a call to this routine, the application should call the DXmSvnSetComponentText, DXmSvnSetComponentPixmap, and DXmSvnSetComponentWidget routines to individually supply component information.

Note that if the number of entries changes, all previous component information is discarded.

DXmSvnSetEntryPosition

Sets the position of an entry in user-defined tree mode.

Format

```
void DXmSvnSetEntryPosition(widget, entry, mode, x, y)
      Widget   widget;
      int      entry;
      Boolean  mode;
      int      x;
      int      y;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which information will be set or changed.

mode
A Boolean value that, when True, specifies that position information returned should be interpreted relative to the upper left corner of the SVN window. When False, the position information returned should be interpreted relative internally to the SVN widget.

x
An integer that represents the *x* position of the entry. If the value for **mode** is True, the value for **x** represents the distance from the corner of the SVN window in pixels. If the value for **mode** is False, the value for **x** represents the position in the entire tree, even though that position extends beyond the limit of what can currently be displayed in the SVN window.

y
An integer that represents the *y* position of the entry. If the value for **mode** is True, the value for **y** represents the distance from the corner of the SVN window in pixels. If the value for **mode** is False, the value for **y** represents the position in the entire tree, even though that position extends beyond the limit of what can currently be displayed in the SVN window.

Description

The DXmSvnSetEntryPosition routine allows an application to implement the layout of the tree when the tree style is DXmSvnKuserDefinedTree. The application can explicitly position an entry, whether that entry is currently visible in the display or not. This call is valid only if the display mode is set to DXmSvnKdisplayTree and the tree style to DXmSvnKuserDefinedTree. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnSetEntrySensitivity

Controls whether the user can select a particular entry.

Format

```
void DXmSvnSetEntrySensitivity(widget, entry, sens)
     Widget   widget;
     int      entry;
     Boolean  sens;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry whose sensitivity will be set or changed.

sens
A Boolean value that, when True, means the entry is sensitive (the user can select it). If the value is False, the SVN widget will dim the display of the entry to indicate that the entry is not sensitive (the user cannot select it).

Description

The DXmSvnSetEntrySensitivity routine changes the sensitivity of an entry (''sensitivity'' means whether the SVN widget will allow the user to select a particular entry). When using this routine, note the following:

- Make the call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.
- If the newly specified sensitivity is the same as the current sensitivity, the call is ignored.
- If the newly specified sensitivity for the entry is different from the current sensitivity, that entry is redisplayed if necessary.
- If an entry is being changed to no longer be sensitive (the user cannot select the entry) *and* that entry is currently selected, the entry will be cleared (canceled) and then redisplayed.

DXmSvnSetEntryTag

Assigns an entry tag to a specified entry.

Format

```
void DXmSvnSetEntryTag(widget, entry, tag)
      Widget        widget;
      int           entry;
      unsigned int  tag;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

entry
An integer that represents the number of the entry for which information will be set or changed.

tag
A value that the application will associate with this entry. This value is returned in all callbacks that set the **entry_number** field in the DXmSvnCallback structure.

Description

The DXmSvnSetEntryTag routine assigns an entry tag to a specified entry, enabling an application to respond to a DXmSvnNgetEntryCallback callback requesting information about that entry. The function of this routine is similar to the DXmSvnSetEntry routine but is more limited in scope, requiring only a subset of the information you would specify for the DXmSvnSetEntry routine.

See the DXmSvnSetEntry and DXmCreateSvn routines for related information.

DXmSvnSetTreePosition

Sets the position of the tree in tree display mode.

Format

```
void DXmSvnSetTreePosition(widget, x, y)
      Widget  widget;
      int     x;
      int     y;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

x
An integer that represents the *x* position for the tree. This value is obtained by a previous call to the DXmSvnGetTreePosition routine.

y
An integer that represents the *y* position for tree. This value is obtained by a previous call to the DXmSvnGetTreePosition routine.

Description

The DXmSvnSetTreePosition routine enables applications to set the exact position of the display from previously stored coordinates. It is valid to call this routine only when the DXmSvnNdisplayMode resource is set to DXmSvnKdisplayTree. (The DXmSvnGetTreePosition routine retrieves the x- and y-coordinates from the SVN widget.)

Note that if the application does not currently have the exact same characteristics (for example, the same mode, fonts, or size and number of entries) as when the DXmSvnGetPosition routine was last called, calling the DXmSvnSetTreePosition routine may generate an incorrect display.

Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnShowHighlighting

Specifies that the SVN widget use highlight graphics when showing highlighted entries.

Format

```
void DXmSvnShowHighlighting(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnShowHighlighting routine shows all highlighting by displaying a box (drawn with dashes) around an entry. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnShowSelections

Specifies that the SVN widget use selection graphics when displaying selected entries.

Format

```
void DXmSvnShowSelections(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnShowSelections routine allows the SVN widget to use reverse video selection graphics when displaying selected entries. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

DXmSvnValidateAll

Ensures that all entries are valid.

Format

```
void DXmSvnValidateAll(widget)
      Widget  widget;
```

Arguments

widget
The identifier (widget ID) of the SVN widget.

Description

The DXmSvnValidateAll routine examines all entries to ensure that each is valid. If other entries are added while a particular entry is being validated, the routine validates those new entries as well. (The routine calls DXmSvnNgetEntryCallback to obtain entries from the application.) This routine will not complete its operation until all entries are valid. Make this call in your application between calls to DXmSvnDisableDisplay and DXmSvnEnableDisplay.

3
DECwindows Extensions to UIL

This chapter explains how to use the extensions that Digital has added to the Motif User Interface Language (UIL). Specifically, it explains how to use nested compound strings, the new command line qualifiers in the UIL compiler, and the Widget Meta Language (WML) binary databases.

In addition, Appendix A documents the UIL built-in tables (which include the additional routines provided by Digital) that you can use to check whether your UIL specification is consistent with the Motif Toolkit. This checking operation occurs during compilation.

3.1 Nested Compound Strings

A nested compound string allows you to nest a given compound string within another relative to the direction in which the string will be displayed. Use the NESTED_COMPOUND_STRING function when one compound string segment needs to be read right-to-left and another needs to be read left to right.

The result of the NESTED_COMPOUND_STRING function is a compound string containing a nested direction. The NESTED_COMPOUND_STRING has one of the following as its value:

- A concatenated nested_string_expression
- A directional string expression
- A string expression

The NESTED_COMPOUND_STRING function has the following syntax:

NESTED_COMPOUND_STRING(*nested_string_expression ...*)

Where:

```
nested_string_expression :==
      {nested_string_expression nested_string_expression ...   ❶
      |RTOL (nested_string_expression)   ❷
      |LTOR (nested_string_expression)   ❸
      |string_expression [SEPARATE=Boolean_expression]};       ❹
```

❶ The nested string expression is the concatenation of two or more nested_ string_expressions into one nested_string_expression. Each nested_string_ expression consists of one of the following:

- A concatenated string expression
- A directional string expression (either RTOL or LTOR)
- A string expression

❷ RTOL formats the nested string expression from right to left.

❸ LTOR formats the nested string expression from left to right.

❹ The string expression is any string allowed in the character set you have chosen. The string expression must be a null-terminated string. It can optionally contain the SEPARATE clause. The SEPARATE clause appends a separator to the end of the compound string if the Boolean expression is TRUE. If you omit the SEPARATE clause, the resulting string does not have a separator.

The compiler evaluates each string expression to see if it is one of the previously described four types, and then concatenates them all into one compound string with nested directional settings.

The following example shows UIL code for a nested compound string:

```
(RTOL ("Hello"
      LTOR ("123", separate=TRUE)
      "World"
      ))
```

This example compound string would be displayed as follows to the end user:

```
123 olleH
    dlroW
```

3.2 Command Qualifiers to the UIL Compiler

The UIL compiler uses the OSF/Motif Release 1.2 and DXm widgets by default. However, you may use other versions or other widgets by using different compiler command options. The following sections describe how to use the UIL compiler on OpenVMS, UNIX, and Windows NT operating systems.

3.2.1 UIL Compiler Command Qualifiers on OpenVMS Systems

Table 3–1 lists the qualifiers for the UIL compiler on OpenVMS systems.

Table 3–1 OpenVMS Compiler Command Qualifiers

Qualifier	Default	Description
/MOTIF	/MOTIF	Uses the OSF/Motif and Digital extended widget database.
/LOCALE	no value	Instructs the compiler to use the current locale when creating localized compound strings. (See the *OSF/Motif Programmer's Guide* for a description of the UIL -s flag.)
/WIDGET_META_ DESCRIPTION=filespec		Selects the Widget Meta Description (WMD) file to be used when processing the UIL source. UIL comes with two WMD files: a standard OSF/Motif widget set, and a Digital extended widget set. To use the standard OSF/Motif widget set, supply the following file specification with this qualifier: DECW$SYSTEM_DEFAULTS: DECW$UIL_OSF_MOTIF_1_1.WMD

(continued on next page)

Table 3–1 (Cont.) OpenVMS Compiler Command Qualifiers

Qualifier	Default	Description
		If you want to use the Digital extended widget set, use this qualifier with the following file specification: DECW$SYSTEM_DEFAULTS: DECW$UIL_DXM_MOTIF.WMD (Ask your system manager if there are additional WMD files that you can use.)
/[NO]LIST	/NOLIST for interactive mode /LIST for batch mode	Directs the compiler to produce a listing file. The listing file goes in the current default directory and has the same name as the input file but with a .LIS file extension. If you include a file specification with the /LIST qualifier, the compiler uses that specification to name the listing file.
/[NO]MACHINE_CODE	/MACHINE_CODE	Functions like the /MACHINE_CODE (or equivalent) qualifier available for most high-level language compilers. If you specify both /LIS and /MACHINE_CODE on the UIL command line, the compiler places in the listing file a description of the records that it wrote to the UID file. This is analogous to a compiler placing the machine code it generates for a high-level language in the listing. The primary purpose of this listing is to isolate errors in the UIL compiler.

(continued on next page)

Table 3–1 (Cont.) OpenVMS Compiler Command Qualifiers

Qualifier	Default	Description
/[NO]OUTPUT	/OUTPUT	Directs the compiler to produce a User Interface Definition (UID) file. By default, /OUTPUT create a UID file in the current default directory with the same name as the input file, but with the UID extension. If you include a file specification with the /OUTPUT qualifier, the compiler uses that specification to name the UID file. The compiler does not produce a UID file when you use the /NOOUTPUT qualifier or when the compiler issues any diagnostics categorized as error or severe. You can use the /NOOUTPUT qualifier when you want to verify only that your UIL module is syntactically correct.
/[NO]WARNINGS	/WARNINGS	Directs the UIL compiler to suppress warning or informational messages or both.

3.2.2 UIL Compiler Command Flags on UNIX and Windows NT Systems

On UNIX and Windows NT operating systems, you use the following command format to invoke the OSF/Motif UIL compiler:

uil [-*flags*] *filename*

Table 3–2 lists the uil command flags.

Table 3–2 Motif UIL Compiler Command Flags on UNIX and Windows NT Systems

Flag	Default	Description
-s		Instructs the compiler to use the current locale when creating localized compound strings.

(continued on next page)

Table 3–2 (Cont.) Motif UIL Compiler Command Flags on UNIX and Windows NT Systems

Flag	Default	Description

-wmd *filename*		Selects the Widget Meta Description (WMD) file to be used when processing the UIL source. UIL comes with two WMD files: a standard OSF/Motif widget set and a Digital extended widget set. If you want to use the standard OSF/Motif widget set, use the following file name with the -wmd flag. The location of this file varies.
		`motif.wmd`
		If you want to use the Digital extended widget set, you can use the following file with the -wmd flag. The location of this file varies.
		`dxm-motif.wmd`
		(Ask your system administrator if there are additional WMD files that you can use.)

-o *filename*	a.uid	Directs the compiler to produce a User Interface Definition (UID) file. By default, UIL creates a UID file with the name a.uid. No UID file is produced if the compiler issues any diagnostics categorized as error or severe.
-v *filename*	No listing is generated	Directs the compiler to produce a listing file. If the -v flag is not present, the compiler does not generate a listing.
-m	No machine code is listed	Directs the compiler to place in the listing file a description of the records that it added to the UID file. This feature helps you to isolate errors.
-w	Warning messages are generated	Directs the compiler to suppress all warning and informational messages. Only error messages and severe messages are generated.

(continued on next page)

Table 3–2 (Cont.) Motif UIL Compiler Command Flags on UNIX and Windows NT Systems

Flag	Default	Description
-I[*pathname*]	/usr/include	If you specify the -I flag followed by a full pathname, the compiler searches for include files whose names do not begin with a slash (/) in the following order: • In the directory path of the main file • In the directory specified by the -I flag • In the standard directory (/usr/include) If you specify -I alone, the compiler searches only the directory in which the main file is located; it does not search for include files in the standard directory (/usr /include).

3.3 WML Binary Databases

The version of Motif supplied by Digital supports the Widget Meta Language (WML) in a binary form called Widget Meta Description (WMD). WML generates a WMD file describing widgets and other attributes for use with the UIL compiler. Digital has included two WML source files on each operating system. You can copy these files to your own subdirectory and modify them so that you can refer to your own widgets by name without making them user-defined.

The following sections describe WML support provided on OpenVMS, UNIX, and Windows NT operating systems. For additional information about the WML, see the *OSF/Motif Programmer's Guide*.

3.3.1 WML Files on OpenVMS Systems

Digital includes the following files to support WML databases on OpenVMS operating systems:

- DECW$UIL_OSF_MOTIF_1_1.WML specifies the Motif toolkit as shipped from OSF.
- DECW$UIL_DXM_MOTIF.WML describes the OSF/Motif toolkit with the Digital extensions to the toolkit.

When adding new widgets or changing widget characteristics, start with a copy of one of the two WML files. Use DECW$DXM_MOTIF.WML if you are using any Digital widgets. Use the OSF/Motif WML files if your application requires a strict OSF widget set.

Digital has also included the following two WMD files (which correspond to the two WML files):

- DECW$UIL_OSF_MOTIF_1_1.WMD specifies the Motif toolkit as shipped from OSF.
- DECW$UIL_DXM_MOTIF.WMD describes the OSF/Motif toolkit with the Digital extensions.

3.3.2 Supporting Additional Widgets on OpenVMS Systems

To allow you to add UIL support for additional widgets, Digital has provided you with the following files:

- Templates

  DECW$SYSTEM_DEFAULTS:DECW$UIL_OSF_MOTIF_1_1.WML
  DECW$SYSTEM_DEFAULTS:DECW$UIL_DXM_MOTIF.WML

  Use either of the previous files as a template for your own WML file.
- DECW$EXAMPLES:UILWMDCREATE.C
- DECW$EXAMPLES:UILWMDCREATE.COM

The procedure is as follows:

1. Extend the DECW$UIL_DXM_MOTIF.WML file to add support for your new widgets.
2. Copy DECW$EXAMPLES:UILWMDCREATE.C to your directory.
3. Add your widget's public include file to UILWMDCREATE.C. In the C file there is a comment section that indicates where in the file you add public include files.
4. Execute UILWMDCREATE.COM by entering a command in the following format:

   ```
   @DECW$EXAMPLES:UILWMDCREATE filename.WML
   ```

   This command procedure performs three major functions:

   1. Runs WML to create .h files.
   2. Compiles, links, and runs a modified UILWMDCREATE.C.

3. Cleans up. (If the command procedure does not finish successfully, it will not clean up the extra files generated. Delete all the new files and subdirectories that the command procedure creates before trying to run the command procedure again.)

After you execute this command procedure, you will have a WMD file that the UIL compiler will use instead of the internal database.

Two other files, an MM and an SDML file, will be created; these are documentation files. The MM file is a troff MM macro format file; the SDML file is a VAX DOCUMENT file. You can use the MM file on a UNIX system, or the SDML file on an OpenVMS for VAX system to create a copy of the UIL built-in tables for your own widget set.

5. Invoke the UIL compiler by entering a command at the dollar sign ($) prompt, using the following format:

```
UIL/MOTIF/WIDGET_META_LANGUAGE=filename.WMD filename.UIL
```

3.3.3 WML Files on UNIX and Windows NT Systems

Digital includes the following files to support WML databases on UNIX and Windows NT operating systems:

- motif.wml specifies the Motif toolkit as shipped from the OSF.
- dxm-motif.wml describes the OSF/Motif toolkit with the additional Digital extensions.

When adding new widgets or changing widget characteristics, start with a copy of one of the two WML files. Use dxm-motif.wml if you are using any Digital widgets. Use the motif.wml files if your application requires a strict OSF/Motif widget set.

Digital has also included the following two WMD files (which correspond to the two WML files):

- motif.wmd specifies the Motif toolkit as shipped from the OSF.
- dxm-motif.wmd describes the OSF/Motif toolkit with the Digital extensions to the toolkit.

3.3.4 Supporting Additional Widgets on UNIX and Windows NT Systems

To allow you to add UIL support for additional widgets, Digital has provided you with the following files for UNIX and Windows NT systems:

- Templates (found in /usr/examples/motif/wml)

  motif.wml
  dxm-motif.wml

  Use either of the previous files as a template for your own wml file.
- wmldbcreate.c
- A script file

To add UIL support for additional widgets, follow these steps:

1. Copy the wml files into your directory.
2. Extend the wml file to add support for your new widgets.
3. Add your widget's public include file to the wmldbcreate.c file.
4. Execute the script file by entering a command in following format:

   xmkmf *filename*
5. Issue the **make all** command to produce the makefile. This makefile performs three major functions:

   1. Runs wml to create .h files.
   2. Compiles, links, and runs a modified wmldbcreate.c.
   3. Cleans up. (If the command procedure does not finish successfully, it will not clean up the extra files generated. Delete any extra files before running the command procedure again.)

   After you execute this script file, you will have a wmd file that the UIL compiler will use instead of the internal database.

   An mm documentation file is created. This file is a troff mm macro format file. You can use the mm file on UNIX systems to create a copy of the UIL built-in tables for your own widget set.
6. Invoke the UIL compiler by entering a command in the following format:

   uil -d *filenamenew*.wmd *filename*.uil -o *filename*.uid

A
UIL Built-In Tables

This appendix lists all the UIL built-in tables used to check that your UIL specification is consistent with the DECwindows Motif Toolkit. The Digital specific arguments are listed first in each table, followed by the OSF/Motif arguments.

For each object in the DECwindows Motif toolkit, this appendix contains a table that lists the reasons, controls (children), and arguments (widget resources) that UIL supports. It also lists the data type and default value of each supported argument.

The default values have been included to indicate the behavior of the object when you do not specify particular arguments in the UIL module. For example, if you do not specify the highlight color for an object, the object is displayed with black as the highlight color.

The UIL name for an argument is identical to the DECwindows Motif Toolkit name for that argument. Callback resources in the DECwindows Motif Toolkit are represented by reasons in UIL. In cases where the DECwindows Motif toolkit resource name is a UIL keyword, the UIL name for the argument has been configured to avoid ambiguity; the argument name is appended to the object name (or an abbreviated form of it).

Some arguments take a symbolic reference to a widget as their value. These arguments are denoted by the term **object reference** in the type column for the argument. To specify a symbolic reference to a widget ID, supply the type and name of an object that is defined in your UIL module.

For example:

```
arguments {
     XmNdefaultButton = XmPushButton My_push_button;
};
```

A.1 DXmColorMix

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNokCallback

UIL Argument Name	Argument Type	Default Value
DXmNbackBlueValue	integer	32767
DXmNbackGreenValue	integer	32767
DXmNbackRedValue	integer	32767
DXmNblackLabel	compound_string	Black
DXmNblueLabel	compound_string	Blue
DXmNbrowserItemCount	integer	5
DXmNbrowserLabel	compound_string	Browser
DXmNclearLabel	compound_string	Clear
DXmNcolorModel	integer	DXmColorModelPicker
DXmNcoolerLabel	compound_string	Cooler
DXmNdarkerLabel	compound_string	Darker
DXmNdisplayColWinHeight	integer	80
DXmNdisplayColWinWidth	integer	80
DXmNdisplayLabel	compound_string	Null
DXmNdisplayWindow	widget_ref	default
DXmNdispWinMargin	integer	20
DXmNearthtoneLabel	compound_string	Earthtones
DXmNfullLabel	compound_string	Full
DXmNgrayLabel	compound_string	Gray
DXmNgreenLabel	compound_string	Green

UIL Argument Name	Argument Type	Default Value
DXmNgreyscaleLabel	compound_string	Grayscale
DXmNgreyscaleOnGreyscale	boolean	True
DXmNhelpLabel	compound_string	Help
DXmNhlsLabel	compound_string	HLS
DXmNhueLabel	compound_string	Hue:
DXmNinterpTileCount	integer	10
DXmNinterpTileHeight	integer	30
DXmNinterpTileWidth	integer	30
DXmNinterpTitleLabel	compound_string	Interpolator
DXmNlighterLabel	compound_string	Lighter
DXmNlightLabel	compound_string	Lightness:
DXmNlightnessIncrement	integer	5
DXmNmainLabel	compound_string	Null
DXmNmatchColors	boolean	True
DXmNmixerLabel	compound_string	Null
DXmNmixerWindow	widget_ref	default
DXmNnewBlueValue	integer	0
DXmNnewGreenValue	integer	0
DXmNnewRedValue	integer	0
DXmNoptionLabel	compound_string	Color Model:
DXmNorigBlueValue	integer	0
DXmNorigGreenValue	integer	0
DXmNorigRedValue	integer	0
DXmNpastelLabel	compound_string	Pastels
DXmNpickerLabel	compound_string	Picker
DXmNpickerTileHeight	integer	30
DXmNpickerTileWidth	integer	30
DXmNpickerTitleLabel	compound_string	Spectrum
DXmNredLabel	compound_string	Red
DXmNresetLabelString	compound_string	Reset
DXmNrgbLabel	compound_string	RGB

UIL Built-In Tables
A.1 DXmColorMix

UIL Argument Name	Argument Type	Default Value
DXmNsatLabel	compound_string	Saturation:
DXmNscratchPadInfoLabel	compound_string	Save colors here...
DXmNscratchPadLabel	compound_string	Scratch Pad...
DXmNsetMixerColorProc	any	default
DXmNsetNewColorProc	any	default
DXmNsliderLabel	compound_string	Percentage
DXmNsmearLabel	compound_string	Smear
DXmNspectrumLabel	compound_string	Spectrum
DXmNundoLabel	compound_string	Undo
DXmNuserPaletteLabel	compound_string	User palette
DXmNvalueLabel	compound_string	Value
DXmNvividLabel	compound_string	Vivids
DXmNwarmerLabel	compound_string	Warmer
DXmNwarmthIncrement	integer	5000
DXmNwhiteLabel	compound_string	White
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Apply
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null

UIL Argument Name	Argument Type	Default Value
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP

UIL Argument Name	Argument Type	Default Value
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNworkWindow	widget_ref	Null
XmNx	integer	0
XmNy	integer	0

A.2 DXmColorMixDialog

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
DXmNbackBlueValue	integer	32767
DXmNbackGreenValue	integer	32767
DXmNbackRedValue	integer	32767
DXmNblackLabel	compound_string	Black
DXmNblueLabel	compound_string	Blue
DXmNbrowserItemCount	integer	5
DXmNbrowserLabel	compound_string	Browser
DXmNclearLabel	compound_string	Clear

UIL Argument Name	Argument Type	Default Value
DXmNcolorModel	integer	DXmColorModelPicker
DXmNcoolerLabel	compound_string	Cooler
DXmNdarkerLabel	compound_string	Darker
DXmNdispWinMargin	integer	20
DXmNdisplayColWinHeight	integer	80
DXmNdisplayColWinWidth	integer	80
DXmNdisplayLabel	compound_string	Null
DXmNdisplayWindow	widget_ref	default
DXmNearthtoneLabel	compound_string	Earthtones
DXmNfullLabel	compound_string	Full
DXmNgrayLabel	compound_string	Gray
DXmNgreenLabel	compound_string	Green
DXmNgreyscaleLabel	compound_string	Grayscale
DXmNgreyscaleOnGreyscale	boolean	True
DXmNhelpLabel	compound_string	Help
DXmNhlsLabel	compound_string	HLS
DXmNhueLabel	compound_string	Hue:
DXmNinterpTileCount	integer	10
DXmNinterpTileHeight	integer	30
DXmNinterpTileWidth	integer	30
DXmNinterpTitleLabel	compound_string	Interpolator
DXmNlighterLabel	compound_string	Lighter
DXmNlightLabel	compound_string	Lightness:
DXmNlightnessIncrement	integer	5
DXmNmainLabel	compound_string	Null
DXmNmatchColors	boolean	True
DXmNmixerLabel	compound_string	Null
DXmNmixerWindow	widget_ref	default
DXmNnewBlueValue	integer	0
DXmNnewGreenValue	integer	0
DXmNnewRedValue	integer	0

UIL Argument Name	Argument Type	Default Value
DXmNoptionLabel	compound_string	Color Model:
DXmNorigBlueValue	integer	0
DXmNorigGreenValue	integer	0
DXmNorigRedValue	integer	0
DXmNpastelLabel	compound_string	Pastels
DXmNpickerLabel	compound_string	Picker
DXmNpickerTileHeight	integer	30
DXmNpickerTileWidth	integer	30
DXmNpickerTitleLabel	compound_string	Spectrum
DXmNredLabel	compound_string	Red
DXmNresetLabelString	compound_string	Reset
DXmNrgbLabel	compound_string	RGB
DXmNsatLabel	compound_string	Saturation:
DXmNscratchPadInfoLabel	compound_string	Save colors here...
DXmNscratchPadLabel	compound_string	Scratch Pad...
DXmNsetMixerColorProc	any	default
DXmNsetNewColorProc	any	default
DXmNsliderLabel	compound_string	Percentage
DXmNsmearLabel	compound_string	Smear
DXmNspectrumLabel	compound_string	Spectrum
DXmNundoLabel	compound_string	Undo
DXmNuserPaletteLabel	compound_string	User palette
DXmNvalueLabel	compound_string	Value
DXmNvividLabel	compound_string	Vivids
DXmNwarmerLabel	compound_string	Warmer
DXmNwarmthIncrement	integer	5000
DXmNwhiteLabel	compound_string	White
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Apply

UIL Argument Name	Argument Type	Default Value
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNworkWindow	widget_ref	Null
XmNx	integer	0
XmNy	integer	0

A.3 DXmCSText

Controls	Reasons
No children are supported	DXmNnoFontCallback
	MrmNcreateCallback
	XmNactivateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNlosingFocusCallback

Controls	Reasons
	XmNmodifyVerifyCallback
	XmNmotionVerifyCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
DXmNbidirectionalCursor	boolean	False
DXmNeditingPath	integer	DXmDIRECTION_RIGHT_DOWN
DXmNtextPath	integer	DXmDIRECTION_RIGHT_DOWN
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNautoShowCursorPosition	boolean	True
XmNbackground	color	dynamic
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNblinkRate	integer	500
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNcolormap	identifier	dynamic
XmNcolumns	integer	20
XmNcursorPosition	integer	0
XmNcursorPositionVisible	boolean	True
XmNdepth	identifier	dynamic
XmNeditable	boolean	True
XmNeditMode	integer	XmSINGLE_LINE_EDIT
XmNfontList	font_table	DXmDefaultFont

UIL Argument Name	Argument Type	Default Value
XmNforeground	color	XtDefaultForeground
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	6
XmNmarginWidth	integer	6
XmNmaxLength	integer	MAXINT
XmNnavigationType	integer	XmTAB_GROUP
XmNpendingDelete	boolean	True
XmNresizeHeight	boolean	False
XmNresizeWidth	boolean	False
XmNrows	integer	1
XmNscreen	identifier	dynamic
XmNselectionArray	any	array of XmTextScanType: XmSELECT_POSITION, XmSELECT_WORD, XmSELECT_LINE, XmSELECT_ALL
XmNselectionArrayCount	integer	4
XmNselectThreshold	integer	5
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopPosition	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null

UIL Argument Name	Argument Type	Default Value
XmNtraversalOn	boolean	True
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	Null
XmNwidth	integer	0
XmNwordWrap	boolean	False
XmNx	integer	0
XmNy	integer	0

A.4 DXmHelpDialog

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
DXmNaddtopicLabel	compound_string	Additional topics
DXmNapplicationName	compound_string	NULL
DXmNbadframeMessage	compound_string	Couldn't find frame !CS
DXmNbadlibMessage	compound_string	Couldn't open library !CS
DXmNcacheHelpLibrary	boolean	False
DXmNcloseLabel	compound_string	Exit
DXmNcols	integer	55
DXmNcopyLabel	compound_string	Copy
DXmNcopyLabelMnem	keysym	C
DXmNcopyLabelMnemCS	string	dynamic
DXmNdefaultPosition	boolean	True
DXmNdismissLabel	compound_string	Dismiss

UIL Argument Name	Argument Type	Default Value
DXmNeditLabel	compound_string	Edit
DXmNeditLabelMnem	keysym	E
DXmNeditLabelMnemCS	string	dynamic
DXmNerroropenMessage	compound_string	Error opening file !CS
DXmNexitLabel	compound_string	Exit
DXmNexitLabelMnem	keysym	E
DXmNexitLabelMnemCS	string	dynamic
DXmNfileLabel	compound_string	File
DXmNfileLabelMnem	keysym	F
DXmNfileLabelMnemCS	string	dynamic
DXmNfirstTopic	compound_string	NULL
DXmNglossaryTopic	compound_string	NULL
DXmNgobackLabel	compound_string	Go Back
DXmNgobackLabelMnem	keysym	B
DXmNgobackLabelMnemCS	string	dynamic
DXmNgobacktopicLabel	compound_string	Go Back
DXmNgooverLabel	compound_string	Go To Overview
DXmNgooverLabelMnem	keysym	O
DXmNgooverLabelMnemCS	string	dynamic
DXmNgotoLabel	compound_string	Go To
DXmNgototopicLabel	compound_string	Go To Topic
DXmNgototopicLabelMnem	keysym	T
DXmNgototopicLabelMnemCS	string	dynamic
DXmNhelpAcknowledgeLabel	compound_string	OK
DXmNhelphelpLabel	compound_string	On Window ...
DXmNhelphelpLabelMnem	keysym	W
DXmNhelphelpLabelMnemCS	string	dynamic
DXmNhelpLabel	compound_string	Using Help
DXmNhelpLabelMnem	keysym	U
DXmNhelpLabelMnemCS	string	dynamic
DXmNhelpOnHelpTitle	compound_string	Using Help

UIL Argument Name	Argument Type	Default Value
DXmNhelpontitleLabel	compound_string	Help On
DXmNhelptitleLabel	compound_string	Help
DXmNhistoryboxLabel	compound_string	Search Topic History
DXmNhistoryLabel	compound_string	History...
DXmNhistoryLabelMnem	keysym	H
DXmNhistoryLabelMnemCS	string	dynamic
DXmNkeywordLabel	compound_string	Keyword...
DXmNkeywordLabelMnem	keysym	K
DXmNkeywordLabelMnemCS	string	dynamic
DXmNkeywordsLabel	compound_string	Keyword
DXmNlibrarySpec	compound_string	NULL
DXmNlibraryType	integer	DXmTextLibrary
DXmNnokeywordMessage	compound_string	Couldn't find keyword !CS
DXmNnotitleMessage	compound_string	No title to match string !CS
DXmNnulllibMessage	compound_string	No library specified
DXmNoverviewTopic	compound_string	NULL
DXmNrows	integer	20
DXmNsaveasLabel	compound_string	Save As...
DXmNsaveasLabelMnem	keysym	A
DXmNsaveasLabelMnemCS	string	dynamic
DXmNsearchapplyLabel	compound_string	Apply
DXmNsearchkeywordboxLabel	compound_string	Search Topic Keywords
DXmNsearchLabel	compound_string	Search
DXmNsearchLabelMnem	keysym	S
DXmNsearchLabelMnemCS	string	dynamic
DXmNsearchtitleboxLabel	compound_string	Search Topic Titles
DXmNselectallLabel	compound_string	Select All
DXmNselectallLabelMnem	keysym	S
DXmNselectallLabelMnemCS	string	dynamic
DXmNtitleLabel	compound_string	Title...
DXmNtitleLabelMnem	keysym	T

UIL Argument Name	Argument Type	Default Value
DXmNtitleLabelMnemCS	string	dynamic
DXmNtitlesLabel	compound_string	Title
DXmNtopictitlesLabel	compound_string	Topic Titles
DXmNviewLabel	compound_string	View
DXmNviewLabelMnem	keysym	V
DXmNviewLabelMnemCS	string	dynamic
DXmNvisitglosLabel	compound_string	Visit Glossary
DXmNvisitglosLabelMnem	keysym	G
DXmNvisitglosLabelMnemCS	string	dynamic
DXmNvisitLabel	compound_string	Visit
DXmNvisittopicLabel	compound_string	Visit Topic
DXmNvisittopicLabelMnem	keysym	V
DXmNvisittopicLabelMnemCS	string	dynamic
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic

UIL Argument Name	Argument Type	Default Value
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNnavigationType	integer	XmTAB_GROUP
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtextFontList	font_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.5 DXmPrintBox

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback

Controls	Reasons
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
DXmNautoPagination	boolean	false
DXmNdefaultPrinter	compound_string	NULL
DXmNdeleteFile	boolean	false
DXmNdoubleSpacing	boolean	false
DXmNfileBurstSheet	integer	DXmFILE_SHEET_ DEFAULT
DXmNfileEndSheet	integer	DXmFILE_SHEET_ DEFAULT
DXmNfileNameCount	integer	0
DXmNfileNameList	string_table	NULL
DXmNfileStartSheet	integer	DXmFILE_SHEET_ DEFAULT
DXmNheader	boolean	false
DXmNholdJob	boolean	false
DXmNinputTray	integer	DXmINPUT_TRAY_ DEFAULT
DXmNjobName	compound_string	NULL
DXmNlayupDefinition	compound_string	NULL
DXmNmessageLog	integer	DXmMESSAGE_LOG_ DEFAULT
DXmNnotify	boolean	true
DXmNnumberCopies	integer	0
DXmNnumberUp	integer	0
DXmNoperatorMessage	compound_string	NULL
DXmNoptionsDialogTitle	compound_string	Print: Options
DXmNorientation	integer	DXmORIENTATION_ DEFAULT
DXmNoutputTray	integer	DXmOUTPUT_TRAY_ DEFAULT
DXmNpageRangeFrom	compound_string	NULL

UIL Argument Name	Argument Type	Default Value
DXmNpageRangeTo	compound_string	NULL
DXmNpageSize	integer	DXmSIZE_DEFAULT
DXmNpassAll	boolean	false
DXmNprintAfter	compound_string	Now
DXmNprinterChoice	compound_string	NULL
DXmNprinterCount	integer	0
DXmNprinterFormChoice	compound_string	NULL
DXmNprinterFormCount	integer	0
DXmNprinterFormList	string_table	NULL
DXmNprinterList	string_table	NULL
DXmNprintFormatChoice	compound_string	NULL
DXmNprintFormatCount	integer	0
DXmNprintFormatList	string_table	NULL
DXmNpriority	integer	0
DXmNsetup	compound_string	NULL
DXmNsheetCount	integer	0
DXmNsheetSize	integer	DXmSIZE_DEFAULT
DXmNsides	integer	DXmSIDES_DEFAULT
DXmNstartSheetComment	compound_string	NULL
DXmNsuppressOptionsMask	integer	DXmSUPPRESS_NONE
DXmNunmanageOnCancel	boolean	false
DXmNunmanageOnOk	boolean	false
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP

UIL Argument Name	Argument Type	Default Value
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextFontList	font_table	Null

UIL Argument Name	Argument Type	Default Value
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.6 DXmPrintDialog

Controls	Reasons
AllWidgetsAndGadgets	
	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
DXmNautoPagination	boolean	false
DXmNdefaultPrinter	compound_string	NULL
DXmNdeleteFile	boolean	false
DXmNdoubleSpacing	boolean	false

UIL Argument Name	Argument Type	Default Value
DXmNfileBurstSheet	integer	DXmFILE_SHEET_ DEFAULT
DXmNfileEndSheet	integer	DXmFILE_SHEET_ DEFAULT
DXmNfileNameCount	integer	0
DXmNfileNameList	string_table	NULL
DXmNfileStartSheet	integer	DXmFILE_SHEET_ DEFAULT
DXmNheader	boolean	false
DXmNholdJob	boolean	false
DXmNinputTray	integer	DXmINPUT_TRAY_ DEFAULT
DXmNjobName	compound_string	NULL
DXmNlayupDefinition	compound_string	NULL
DXmNmessageLog	integer	DXmMESSAGE_LOG_ DEFAULT
DXmNnotify	boolean	true
DXmNnumberCopies	integer	0
DXmNnumberUp	integer	0
DXmNoperatorMessage	compound_string	NULL
DXmNoptionsDialogTitle	compound_string	Print: Options
DXmNorientation	integer	DXmORIENTATION_ DEFAULT
DXmNoutputTray	integer	DXmOUTPUT_TRAY_ DEFAULT
DXmNpageRangeFrom	compound_string	NULL
DXmNpageRangeTo	compound_string	NULL
DXmNpageSize	integer	DXmSIZE_DEFAULT
DXmNpassAll	boolean	false
DXmNprintAfter	compound_string	Now
DXmNprinterChoice	compound_string	NULL
DXmNprinterCount	integer	0
DXmNprinterFormChoice	compound_string	NULL
DXmNprinterFormCount	integer	0

UIL Argument Name	Argument Type	Default Value
DXmNprinterFormList	string_table	NULL
DXmNprinterList	string_table	NULL
DXmNprintFormatChoice	compound_string	NULL
DXmNprintFormatCount	integer	0
DXmNprintFormatList	string_table	NULL
DXmNpriority	integer	0
DXmNsetup	compound_string	NULL
DXmNsheetCount	integer	0
DXmNsheetSize	integer	DXmSIZE_DEFAULT
DXmNsides	integer	DXmSIDES_DEFAULT
DXmNstartSheetComment	compound_string	NULL
DXmNsuppressOptionsMask	integer	DXmSUPPRESS_NONE
DXmNunmanageOnCancel	boolean	false
DXmNunmanageOnOk	boolean	false
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null

UIL Argument Name	Argument Type	Default Value
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null

UIL Argument Name	Argument Type	Default Value
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.7 DXmScrolledCSText

Controls	Reasons
No children are supported	DXmNnoFontCallback
	MrmNcreateCallback
	XmNactivateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNlosingFocusCallback
	XmNmodifyVerifyCallback
	XmNmotionVerifyCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
DXmNbidirectionalCursor	boolean	False
DXmNeditingPath	integer	DXmDIRECTION_RIGHT_DOWN
DXmNtextPath	integer	DXmDIRECTION_RIGHT_DOWN
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNautoShowCursorPosition	boolean	True
XmNbackground	color	dynamic
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNblinkRate	integer	500
XmNborderColor	color	XtDefaultForeground

UIL Argument Name	Argument Type	Default Value
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNcolumns	integer	20
XmNcursorPosition	integer	0
XmNcursorPositionVisible	boolean	True
XmNdepth	identifier	dynamic
XmNeditable	boolean	True
XmNeditMode	integer	XmSINGLE_LINE_EDIT
XmNfontList	font_table	DXmDefaultFont
XmNforeground	color	XtDefaultForeground
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	6
XmNmarginWidth	integer	6
XmNmaxLength	integer	MAXINT
XmNnavigationType	integer	XmTAB_GROUP
XmNpendingDelete	boolean	True
XmNresizeHeight	boolean	False
XmNresizeWidth	boolean	False
XmNrows	integer	1
XmNscreen	identifier	dynamic
XmNscrollHorizontal	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNscrollLeftSide	boolean	False
XmNscrollTopSide	boolean	False
XmNscrollVertical	boolean	False
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ DEFAULT
XmNtopPosition	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	True
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	Null
XmNwidth	integer	0
XmNwordWrap	boolean	False
XmNx	integer	0
XmNy	integer	0

A.8 DXmSvn

Controls	Reasons
AllWidgetsAndGadgets	DXmSvnNattachToSourceCallback
	DXmSvnNdetachFromSourceCallback
	DXmSvnNdisplayChangedCallback
	DXmSvnNdraggingCallback
	DXmSvnNdraggingEndCallback
	DXmSvnNentrySelectedCallback
	DXmSvnNentryTransferCallback
	DXmSvnNentryUnselectedCallback

Controls	Reasons
	DXmSvnNextendConfirmCallback
	DXmSvnNgetEntryCallback
	DXmSvnNhelpRequestedCallback
	DXmSvnNpopupMenuCallback
	DXmSvnNselectAndConfirmCallback
	DXmSvnNselectionsDraggedCallback
	DXmSvnNtransitionsDoneCallback
	MrmNcreateCallback
	XmNdestroyCallback

UIL Argument Name	Argument Type	Default Value
DXmSvnNcolumnLines	boolean	False
DXmSvnNdefaultSpacing	integer	12
DXmSvnNdisplayMode	integer	DXmSvnKdisplayOutline
DXmSvnNexpectHighlighting	boolean	False
DXmSvnNfixedWidthEntries	boolean	True
DXmSvnNfontList	font_table	DXmDefaultFont
DXmSvnNfontListLevel0	font_table	Null
DXmSvnNfontListLevel1	font_table	Null
DXmSvnNfontListLevel2	font_table	Null
DXmSvnNfontListLevel3	font_table	Null
DXmSvnNfontListLevel4	font_table	Null
DXmSvnNforceSeqGetEntry	boolean	False
DXmSvnNghostHeight	integer	0
DXmSvnNghostPixmap	pixmap	Null
DXmSvnNghostWidth	integer	0
DXmSvnNghostX	integer	0
DXmSvnNghostY	integer	0
DXmSvnNindentMargin	integer	16
DXmSvnNliveScrolling	boolean	True

UIL Argument Name	Argument Type	Default Value
DXmSvnNmultipleSelections	boolean	True
DXmSvnNnavWindowTitle	compound_string	Null
DXmSvnNnumberOfEntries	integer	0
DXmSvnNoutlineHScrollWidget	widget_ref	Null
DXmSvnNpaneWidget	widget_ref	Null
DXmSvnNprimaryPercentage	integer	50
DXmSvnNprimaryWindowWidget	widget_ref	Null
DXmSvnNsecondaryBaseX	integer	0
DXmSvnNsecondaryComponentsUnmapped	boolean	False
DXmSvnNsecondaryWindowWidget	widget_ref	Null
DXmSvnNselectionMode	integer	DXmSvnKselectEntry
DXmSvnNshowPathToRoot	boolean	True
DXmSvnNstartColumnComponent	integer	0
DXmSvnNstartLocationCursor	integer	1
DXmSvnNtreeArcWidth	integer	15
DXmSvnNtreeCenteredComponents	boolean	False
DXmSvnNtreeEntryOutlines	boolean	True
DXmSvnNtreeEntryShadows	boolean	True
DXmSvnNtreeIndexAll	boolean	True
DXmSvnNtreeLevelSpacing	integer	5
DXmSvnNtreePerpendicularLines	boolean	True
DXmSvnNtreeSiblingSpacing	integer	5
DXmSvnNtreeStyle	integer	DXmSvnKoutlineTree
DXmSvnNtruncateText	boolean	False
DXmSvnNuseScrollButtons	boolean	True
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black

UIL Argument Name	Argument Type	Default Value
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	XtExtdefaultforeground
XmNheight	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	2
XmNmarginWidth	integer	2
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNtranslations	translation_table	Null
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.9 XmArrowButton

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNarrowDirection	integer	XmDYNAMIC
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmultiClick	integer	XmMULTICLICK_ DISCARD
XmNnavigationType	integer	XmNONE
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null

UIL Argument Name	Argument Type	Default Value
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.10 XmArrowButtonGadget

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNancestorSensitive	boolean	True
XmNarrowDirection	integer	XmDYNAMIC
XmNborderWidth	integer	1
XmNheight	integer	0
XmNhighlightOnEnter	boolean	False
XmNhighlightThickness	integer	0
XmNmultiClick	integer	XmMULTICLICK_ DISCARD
XmNnavigationType	integer	XmNONE
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS

UIL Argument Name	Argument Type	Default Value
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.11 XmBulletinBoard

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null

UIL Argument Name	Argument Type	Default Value
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.12 XmBulletinBoardDialog

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null

A.12 XmBulletinBoardDialog

UIL Argument Name	Argument Type	Default Value
XmNcancelButton	widget_ref	Null
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null

UIL Argument Name	Argument Type	Default Value
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.13 XmCascadeButton

Controls	Reasons
XmPulldownMenu	MrmNcreateCallback
	XmNactivateCallback
	XmNcascadingCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcascadePixmap	pixmap	menu-cascade
XmNcolormap	identifier	dynamic

UIL Argument Name	Argument Type	Default Value
XmNdepth	identifier	dynamic
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmappedWhenManaged	boolean	True
XmNmappingDelay	integer	100
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R

UIL Argument Name	Argument Type	Default Value
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.14 XmCascadeButtonGadget

Controls	Reasons
XmPulldownMenu	MrmNcreateCallback
	XmNactivateCallback
	XmNcascadingCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True
XmNborderWidth	integer	1
XmNcascadePixmap	pixmap	menu-cascade
XmNfontList	font_table	Fixed
XmNheight	integer	0
XmNhighlightOnEnter	boolean	False
XmNhighlightThickness	integer	0

A.14 XmCascadeButtonGadget

UIL Argument Name	Argument Type	Default Value
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmappingDelay	integer	100
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.15 XmCommand

Controls	Reasons
No children are supported	MrmNcreateCallback

Controls	Reasons
	XmNcommandChangedCallback
	XmNcommandEnteredCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNcommand	compound_string	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_COMMAND
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black

UIL Argument Name	Argument Type	Default Value
XmNhighlightPixmap	pixmap	dynamic
XmNhistoryItemCount	integer	0
XmNhistoryItems	string_table	Null
XmNhistoryMaxItems	integer	100
XmNhistoryVisibleItemCount	integer	8
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNlistItemCount	integer	0
XmNlistItems	string_table	Null
XmNlistVisibleItemCount	integer	8
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNpromptString	compound_string	>
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNselectionLabelString	compound_string	Selection
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextAccelerators	translation_table	
XmNtextColumns	integer	20
XmNtextFontList	font_table	Null
XmNtextString	compound_string	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic

UIL Argument Name	Argument Type	Default Value
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.16 XmDrawingArea

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNexposeCallback
	XmNhelpCallback
	XmNinputCallback
	XmNresizeCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic

UIL Argument Name	Argument Type	Default Value
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.17 XmDrawnButton

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNexposeCallback
	XmNhelpCallback
	XmNresizeCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNacceleratorText	compound_string	Null
XmNaccelerators	translation_table	Null
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0

A.17 XmDrawnButton

UIL Argument Name	Argument Type	Default Value
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmappedWhenManaged	boolean	True
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNmultiClick	integer	XmMULTICLICK_ DISCARD
XmNnavigationType	integer	XmNONE
XmNpushButtonEnabled	boolean	False
XmNrecomputeSize	boolean	True
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_ETCHED_IN
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic

UIL Argument Name	Argument Type	Default Value
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.18 XmErrorDialog

Controls	Reasons
XmMessageBox	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black

A.18 XmErrorDialog

UIL Argument Name	Argument Type	Default Value
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.19 XmFileSelectionBox

Controls	Reasons
AllWidgets	MrmNcreateCallback
	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback

A.19 XmFileSelectionBox

Controls	Reasons
	XmNhelpCallback
	XmNmapCallback
	XmNnoMatchCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Filter
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null

UIL Argument Name	Argument Type	Default Value
XmNdialogType	integer	dynamic
XmNdirListItemCount	integer	0
XmNdirListItems	string_table	dynamic
XmNdirListLabelString	compound_string	Directories
XmNdirMask	compound_string	*
XmNdirSearchProc	any	default
XmNdirSpec	compound_string	Null
XmNdirectory	compound_string	current working directory
XmNfileListItemCount	integer	0
XmNfileListItems	string_table	NULL
XmNfileListLabelString	compound_string	Files
XmNfileSearchProc	any	default
XmNfileTypeMask	integer	XmFILE_REGULAR
XmNfilterLabelString	compound_string	File Filter
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNlistItemCount	integer	0
XmNlistItems	string_table	Null
XmNlistLabelString	compound_string	Files
XmNlistUpdated	boolean	True
XmNlistVisibleItemCount	integer	8
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNminimizeButtons	boolean	False

A.19 XmFileSelectionBox

UIL Argument Name	Argument Type	Default Value
XmNmustMatch	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoMatchString	compound_string	[ ]
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNpattern	compound_string	*
XmNqualifySearchDataProc	any	default
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNselectionLabelString	compound_string	Selection
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextAccelerators	translation_table	
XmNtextColumns	integer	20
XmNtextFontList	font_table	Null
XmNtextString	compound_string	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.20 XmFileSelectionDialog

Controls	Reasons
AllWidgets	MrmNcreateCallback
	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNnoMatchCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Filter
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	cancel

A.20 XmFileSelectionDialog

UIL Argument Name	Argument Type	Default Value
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	dynamic
XmNdirListItemCount	integer	0
XmNdirListItems	string_table	dynamic
XmNdirListLabelString	compound_string	Directories
XmNdirMask	compound_string	*
XmNdirSearchProc	any	default
XmNdirSpec	compound_string	Null
XmNdirectory	compound_string	current working directory
XmNfileListItemCount	integer	0
XmNfileListItems	string_table	NULL
XmNfileListLabelString	compound_string	Files
XmNfileSearchProc	any	default
XmNfileTypeMask	integer	XmFILE_REGULAR
XmNfilterLabelString	compound_string	File Filter
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNlistItemCount	integer	0
XmNlistItems	string_table	Null
XmNlistLabelString	compound_string	Files

UIL Argument Name	Argument Type	Default Value
XmNlistUpdated	boolean	True
XmNlistVisibleItemCount	integer	8
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNminimizeButtons	boolean	False
XmNmustMatch	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoMatchString	compound_string	[ ]
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNpattern	compound_string	*
XmNqualifySearchDataProc	any	default
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNselectionLabelString	compound_string	Selection
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextAccelerators	translation_table	
XmNtextColumns	integer	20
XmNtextFontList	font_table	Null
XmNtextString	compound_string	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS

UIL Argument Name	Argument Type	Default Value
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.21 XmForm

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomAttachment	integer	XmATTACH_NONE
XmNbottomOffset	integer	0
XmNbottomPosition	integer	0
XmNbottomShadowColor	color	dynamic

UIL Argument Name	Argument Type	Default Value
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbottomWidget	widget_ref	Null
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNfractionBase	integer	100
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNhorizontalSpacing	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNleftAttachment	integer	XmATTACH_NONE
XmNleftOffset	integer	0
XmNleftPosition	integer	0
XmNleftWidget	widget_ref	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNresizable	boolean	True
XmNresizePolicy	integer	XmRESIZE_ANY
XmNrightAttachment	integer	XmATTACH_NONE

UIL Argument Name	Argument Type	Default Value
XmNrightOffset	integer	0
XmNrightPosition	integer	0
XmNrightWidget	widget_ref	Null
XmNrubberPositioning	boolean	False
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopAttachment	integer	XmATTACH_NONE
XmNtopOffset	integer	0
XmNtopPosition	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtopWidget	widget_ref	Null
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNverticalSpacing	integer	0
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.22 XmFormDialog

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback

Controls	Reasons
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomAttachment	integer	XmATTACH_NONE
XmNbottomOffset	integer	0
XmNbottomPosition	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbottomWidget	widget_ref	Null
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic

A.22 XmFormDialog

UIL Argument Name	Argument Type	Default Value
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNforeground	color	dynamic
XmNfractionBase	integer	100
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNhorizontalSpacing	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNleftAttachment	integer	XmATTACH_NONE
XmNleftOffset	integer	0
XmNleftPosition	integer	0
XmNleftWidget	widget_ref	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNresizable	boolean	True
XmNresizePolicy	integer	XmRESIZE_ANY
XmNrightAttachment	integer	XmATTACH_NONE
XmNrightOffset	integer	0
XmNrightPosition	integer	0
XmNrightWidget	widget_ref	Null
XmNrubberPositioning	boolean	False
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT

UIL Argument Name	Argument Type	Default Value
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopAttachment	integer	XmATTACH_NONE
XmNtopOffset	integer	0
XmNtopPosition	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtopWidget	widget_ref	Null
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNverticalSpacing	integer	0
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.23 XmFrame

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White

A.23 XmFrame

UIL Argument Name	Argument Type	Default Value
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	0
XmNmarginWidth	integer	0
XmNnavigationType	integer	XmTAB_GROUP
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	dynamic
XmNshadowType	integer	XmSHADOW_ETCHED_IN
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS

UIL Argument Name	Argument Type	Default Value
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.24 XmInformationDialog

Controls	Reasons
XmMessageBox	MrmNcreateCallback
XmPushButton	XmNcancelCallback
user_defined	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP

A.24 XmInformationDialog

UIL Argument Name	Argument Type	Default Value
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True

UIL Argument Name	Argument Type	Default Value
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.25 XmLabel

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNaccelerators	translation_table	Null
XmNacceleratorText	compound_string	Null
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True

UIL Argument Name	Argument Type	Default Value
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmappedWhenManaged	boolean	True
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2

UIL Argument Name	Argument Type	Default Value
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.26 XmLabelGadget

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNacceleratorText	compound_string	Null
XmNalignment	integer	XmALIGNMENT_CENTER

UIL Argument Name	Argument Type	Default Value
XmNancestorSensitive	boolean	True
XmNborderWidth	integer	1
XmNfontList	font_table	Fixed
XmNheight	integer	0
XmNhighlightOnEnter	boolean	False
XmNhighlightThickness	integer	0
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.27 XmList

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNbrowseSelectionCallback
	XmNdefaultActionCallback
	XmNdestroyCallback
	XmNextendedSelectionCallback
	XmNhelpCallback
	XmNmultipleSelectionCallback
	XmNsingleSelectionCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNautomaticSelection	boolean	False
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNdoubleClickInterval	integer	250
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black

A.27 XmList

UIL Argument Name	Argument Type	Default Value
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNitemCount	integer	0
XmNitems	string_table	Null
XmNlistMarginHeight	integer	0
XmNlistMarginWidth	integer	0
XmNlistSizePolicy	integer	XmVARIABLE
XmNlistSpacing	integer	0
XmNmappedWhenManaged	boolean	True
XmNnavigationType	integer	XmTAB_GROUP
XmNscreen	identifier	dynamic
XmNselectedItemCount	integer	0
XmNselectedItems	string_table	Null
XmNselectionPolicy	integer	XmBROWSE_SELECT
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopItemPosition	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvisibleItemCount	integer	1
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.28 XmMainWindow

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNcommandWindow	widget_ref	Null
XmNcommandWindowLocation	integer	XmCOMMAND_ABOVE_ WORKSPACE
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNhorizontalScrollBar	widget_ref	Null
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNmainWindowMarginHeight	integer	0
XmNmainWindowMarginWidth	integer	0

A.28 XmMainWindow

UIL Argument Name	Argument Type	Default Value
XmNmappedWhenManaged	boolean	True
XmNmenuBar	widget_ref	Null
XmNmessageWindow	widget_ref	Null
XmNnavigationType	integer	XmTAB_GROUP
XmNscreen	identifier	dynamic
XmNscrollBarDisplayPolicy	integer	XmSTATIC
XmNscrollBarPlacement	integer	XmBOTTOM_RIGHT
XmNscrolledWindowMarginHeight	integer	0
XmNscrolledWindowMarginWidth	integer	0
XmNscrollingPolicy	integer	XmAPPLICATION_ DEFINED
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshowSeparator	boolean	False
XmNspacing	integer	4
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNverticalScrollBar	widget_ref	Null
XmNvisualPolicy	integer	XmVARIABLE
XmNwidth	integer	0
XmNworkWindow	widget_ref	Null
XmNx	integer	0
XmNy	integer	0

A.29 XmMenuBar

Controls	Reasons
XmCascadeButton	MrmNcreateCallback
XmCascadeButtonGadget	XmNdestroyCallback
XmDrawnButton	XmNentryCallback
XmLabel	XmNhelpCallback
XmLabelGadget	XmNmapCallback
XmPushButton	XmNunmapCallback
XmPushButtonGadget	
XmSeparator	
XmSeparatorGadget	
XmToggleButton	
XmToggleButtonGadget	
user_defined	

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic

A.29 XmMenuBar

UIL Argument Name	Argument Type	Default Value
XmNentryAlignment	integer	dynamic
XmNentryBorder	integer	dynamic
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True
XmNisHomogeneous	boolean	dynamic
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA

UIL Argument Name	Argument Type	Default Value
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.30 XmMessageBox

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

A.30 XmMessageBox

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null

UIL Argument Name	Argument Type	Default Value
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.31 XmMessageDialog

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON

UIL Argument Name	Argument Type	Default Value
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null

UIL Argument Name	Argument Type	Default Value
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.32 XmOptionMenu

Controls	Reasons
XmPulldownMenu	MrmNcreateCallback
	XmNdestroyCallback
	XmNentryCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black

UIL Argument Name	Argument Type	Default Value
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNentryAlignment	integer	dynamic
XmNentryBorder	integer	dynamic
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True
XmNisHomogeneous	boolean	dynamic
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic

UIL Argument Name	Argument Type	Default Value
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.33 XmPanedWindow

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowResize	boolean	False
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	3
XmNmarginWidth	integer	3
XmNnavigationType	integer	XmTAB_GROUP
XmNpaneMaximum	integer	1000
XmNpaneMinimum	integer	1
XmNrefigureMode	boolean	True
XmNsashHeight	integer	10
XmNsashIndent	integer	-10
XmNsashShadowThickness	integer	2
XmNsashWidth	integer	10

UIL Argument Name	Argument Type	Default Value
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNseparatorOn	boolean	True
XmNshadowThickness	integer	0
XmNskipAdjust	boolean	False
XmNspacing	integer	8
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.34 XmPopupMenu

Controls	Reasons
XmCascadeButton	MrmNcreateCallback
XmCascadeButtonGadget	XmNdestroyCallback
XmDrawnButton	XmNentryCallback
XmLabel	XmNhelpCallback
XmLabelGadget	XmNmapCallback
XmPushButton	XmNunmapCallback
XmPushButtonGadget	
XmSeparator	
XmSeparatorGadget	
XmToggleButton	

Controls	Reasons
XmToggleButtonGadget	
user_defined	

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNentryAlignment	integer	dynamic
XmNentryBorder	integer	dynamic
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True
XmNisHomogeneous	boolean	dynamic

A.34 XmPopupMenu

UIL Argument Name	Argument Type	Default Value
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.35 XmPromptDialog

Controls	Reasons
XmPushButton	MrmNcreateCallback
user_defined	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNnoMatchCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Apply
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black

A.35 XmPromptDialog

UIL Argument Name	Argument Type	Default Value
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNlistItemCount	integer	0
XmNlistItems	string_table	Null
XmNlistLabelString	compound_string	Null
XmNlistVisibleItemCount	integer	8
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNminimizeButtons	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNmustMatch	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNselectionLabelString	compound_string	Selection
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextAccelerators	translation_table	
XmNtextColumns	integer	20
XmNtextFontList	font_table	Null
XmNtextString	compound_string	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.36 XmPulldownMenu

Controls	Reasons
XmCascadeButton	MrmNcreateCallback

A.36 XmPulldownMenu

Controls	Reasons
XmCascadeButtonGadget	XmNdestroyCallback
XmDrawnButton	XmNentryCallback
XmLabel	XmNhelpCallback
XmLabelGadget	XmNmapCallback
XmPushButton	XmNunmapCallback
XmPushButtonGadget	
XmSeparator	
XmSeparatorGadget	
XmToggleButton	
XmToggleButtonGadget	
user_defined	

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNentryAlignment	integer	dynamic
XmNentryBorder	integer	dynamic

UIL Argument Name	Argument Type	Default Value
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True
XmNisHomogeneous	boolean	dynamic
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True

UIL Argument Name	Argument Type	Default Value
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.37 XmPushButton

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNaccelerators	translation_table	Null
XmNacceleratorText	compound_string	Null

UIL Argument Name	Argument Type	Default Value
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True
XmNarmColor	color	dynamic
XmNarmPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdefaultButtonShadowThickness	integer	0
XmNdepth	identifier	dynamic
XmNfillOnArm	boolean	True
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING

A.37 XmPushButton

UIL Argument Name	Argument Type	Default Value
XmNmappedWhenManaged	boolean	True
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNmultiClick	integer	XmMULTICLICK_ DISCARD
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshowAsDefault	integer	0
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.38 XmPushButtonGadget

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNacceleratorText	compound_string	Null
XmNalignment	integer	XmALIGNMENT_ CENTER
XmNancestorSensitive	boolean	True
XmNarmColor	color	dynamic
XmNarmPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNdefaultButtonShadowThickness	integer	0
XmNfillOnArm	boolean	True
XmNfontList	font_table	Fixed
XmNheight	integer	0
XmNhighlightOnEnter	boolean	False
XmNhighlightThickness	integer	0
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2

A.38 XmPushButtonGadget

UIL Argument Name	Argument Type	Default Value
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNmultiClick	integer	XmMULTICLICK_ DISCARD
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshowAsDefault	integer	0
XmNstringDirection	integer	XmSTRING_ DIRECTION_L_TO_R
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.39 XmQuestionDialog

Controls	Reasons
XmMessageBox	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback

Controls	Reasons
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help

A.39 XmQuestionDialog

UIL Argument Name	Argument Type	Default Value
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null

UIL Argument Name	Argument Type	Default Value
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.40 XmRadioBox

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNentryCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNentryAlignment	integer	dynamic

A.40 XmRadioBox

UIL Argument Name	Argument Type	Default Value
XmNentryBorder	integer	dynamic
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True
XmNisHomogeneous	boolean	dynamic
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA
XmNscreen	identifier	dynamic

UIL Argument Name	Argument Type	Default Value
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.41 XmRowColumn

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNentryCallback
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True

A.41 XmRowColumn

UIL Argument Name	Argument Type	Default Value
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNentryAlignment	integer	dynamic
XmNentryBorder	integer	dynamic
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True
XmNisHomogeneous	boolean	dynamic
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null

UIL Argument Name	Argument Type	Default Value
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.42 XmScale

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNdragCallback
	XmNhelpCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdecimalPoints	integer	0
XmNdepth	identifier	dynamic
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True

UIL Argument Name	Argument Type	Default Value
XmNinsertPosition	identifier	Null
XmNmappedWhenManaged	boolean	True
XmNmaximum	integer	100
XmNminimum	integer	0
XmNnavigationType	integer	XmTAB_GROUP
XmNorientation	integer	XmVERTICAL
XmNprocessingDirection	integer	XmMAX_ON_TOP
XmNscaleHeight	integer	0
XmNscaleMultiple	integer	1
XmNscaleWidth	integer	0
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshowValue	boolean	False
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtitleString	compound_string	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	0
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.43 XmScrollBar

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNdecrementCallback
	XmNdestroyCallback
	XmNdragCallback
	XmNhelpCallback
	XmNincrementCallback
	XmNpageDecrementCallback
	XmNpageIncrementCallback
	XmNtoBottomCallback
	XmNtoTopCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black

UIL Argument Name	Argument Type	Default Value
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNincrement	integer	1
XmNinitialDelay	integer	250
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmaximum	integer	0
XmNminimum	integer	0
XmNnavigationType	integer	XmSTICKY_TAB_GROUP
XmNorientation	integer	XmVERTICAL
XmNpageIncrement	integer	10
XmNprocessingDirection	integer	XmMAX_ON_BOTTOM
XmNrepeatDelay	integer	50
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNshowArrows	boolean	True
XmNsliderSize	integer	10
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	dynamic
XmNtroughColor	color	dynamic
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.44 XmScrolledList

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNbrowseSelectionCallback
	XmNdefaultActionCallback
	XmNdestroyCallback
	XmNextendedSelectionCallback
	XmNhelpCallback
	XmNmultipleSelectionCallback
	XmNsingleSelectionCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNautomaticSelection	boolean	False
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNdoubleClickInterval	integer	250
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black

UIL Argument Name	Argument Type	Default Value
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNhorizontalScrollBar	widget_ref	Null
XmNinitialResourcesPersistent	boolean	True
XmNitemCount	integer	0
XmNitems	string_table	Null
XmNlistMarginHeight	integer	0
XmNlistMarginWidth	integer	0
XmNlistSizePolicy	integer	XmVARIABLE
XmNlistSpacing	integer	0
XmNmappedWhenManaged	boolean	True
XmNnavigationType	integer	XmTAB_GROUP
XmNscreen	identifier	dynamic
XmNscrollBarDisplayPolicy	integer	XmAS_NEEDED
XmNscrollBarPlacement	integer	XmBOTTOM_RIGHT
XmNscrolledWindowMarginHeight	integer	0
XmNscrolledWindowMarginWidth	integer	0
XmNselectedItemCount	integer	0
XmNselectedItems	string_table	Null
XmNselectionPolicy	integer	XmBROWSE_SELECT
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNspacing	integer	4
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopItemPosition	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False

A.44 XmScrolledList

UIL Argument Name	Argument Type	Default Value
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNverticalScrollBar	widget_ref	Null
XmNvisibleItemCount	integer	1
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.45 XmScrolledText

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNgainPrimaryCallback
	XmNhelpCallback
	XmNlosePrimaryCallback
	XmNlosingFocusCallback
	XmNmodifyVerifyCallback
	XmNmotionVerifyCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNautoShowCursorPosition	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNblinkRate	integer	500

UIL Argument Name	Argument Type	Default Value
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNcolumns	integer	20
XmNcursorPosition	integer	0
XmNcursorPositionVisible	boolean	True
XmNdepth	identifier	dynamic
XmNeditMode	integer	XmSINGLE_LINE_EDIT
XmNeditable	boolean	True
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	3
XmNmarginWidth	integer	3
XmNmaxLength	integer	MAXINT
XmNnavigationType	integer	XmTAB_GROUP
XmNpendingDelete	boolean	True
XmNresizeHeight	boolean	False
XmNresizeWidth	boolean	False
XmNrows	integer	1
XmNscreen	identifier	dynamic

A.45 XmScrolledText

UIL Argument Name	Argument Type	Default Value
XmNscrolledWindowMarginHeight	integer	0
XmNscrolledWindowMarginWidth	integer	0
XmNscrollHorizontal	boolean	True
XmNscrollLeftSide	boolean	True
XmNscrollTopSide	boolean	False
XmNscrollVertical	boolean	False
XmNselectionArray	any	array of XmTextScanType: XmSELECT_POSITION, XmSELECT_WORD, XmSELECT_LINE, XmSELECT_ALL
XmNselectionArrayCount	integer	4
XmNselectThreshold	integer	5
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNspacing	integer	4
XmNtopCharacter	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	Null
XmNverifyBell	boolean	True
XmNwidth	integer	0
XmNwordWrap	boolean	False
XmNx	integer	0
XmNy	integer	0

A.46 XmScrolledWindow

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNhorizontalScrollBar	widget_ref	Null
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNmappedWhenManaged	boolean	True
XmNnavigationType	integer	XmTAB_GROUP
XmNscreen	identifier	dynamic
XmNscrollBarDisplayPolicy	integer	XmSTATIC

UIL Argument Name	Argument Type	Default Value
XmNscrollBarPlacement	integer	XmBOTTOM_RIGHT
XmNscrolledWindowMarginHeight	integer	0
XmNscrolledWindowMarginWidth	integer	0
XmNscrollingPolicy	integer	XmAPPLICATION_ DEFINED
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	4
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNverticalScrollBar	widget_ref	Null
XmNvisualPolicy	integer	XmVARIABLE
XmNwidth	integer	0
XmNworkWindow	widget_ref	Null
XmNx	integer	0
XmNy	integer	0

A.47 XmSelectionBox

Controls	Reasons
XmPushButton	MrmNcreateCallback
user_defined	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback

Controls	Reasons
	XmNmapCallback
	XmNnoMatchCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Apply
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	dynamic

A.47 XmSelectionBox

UIL Argument Name	Argument Type	Default Value
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNlistItemCount	integer	0
XmNlistItems	string_table	Null
XmNlistLabelString	compound_string	Null
XmNlistVisibleItemCount	integer	8
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNminimizeButtons	boolean	False
XmNmustMatch	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNselectionLabelString	compound_string	Selection
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtextAccelerators	translation_table	default
XmNtextColumns	integer	20
XmNtextFontList	font_table	Null
XmNtextString	compound_string	Null

UIL Argument Name	Argument Type	Default Value
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.48 XmSelectionDialog

Controls	Reasons
XmPushButton	MrmNcreateCallback
user_defined	XmNapplyCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNnoMatchCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNapplyLabelString	compound_string	Apply

A.48 XmSelectionDialog

UIL Argument Name	Argument Type	Default Value
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNlistItemCount	integer	0
XmNlistItems	string_table	Null
XmNlistLabelString	compound_string	Null

UIL Argument Name	Argument Type	Default Value
XmNlistVisibleItemCount	integer	8
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNminimizeButtons	boolean	False
XmNmustMatch	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNselectionLabelString	compound_string	Selection
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNtextAccelerators	translation_table	default
XmNtextColumns	integer	20
XmNtextFontList	font_table	Null
XmNtextString	compound_string	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.49 XmSeparator

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmargin	integer	0
XmNnavigationType	integer	XmNONE
XmNorientation	integer	XmHORIZONTAL

UIL Argument Name	Argument Type	Default Value
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNseparatorType	integer	XmSHADOW_ETCHED_IN
XmNshadowThickness	integer	2
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.50 XmSeparatorGadget

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNdestroyCallback
	XmNhelpCallback

UIL Argument Name	Argument Type	Default Value
XmNancestorSensitive	boolean	True
XmNborderWidth	integer	1
XmNheight	integer	0
XmNhighlightOnEnter	boolean	False
XmNhighlightThickness	integer	0
XmNmargin	integer	0
XmNnavigationType	integer	XmNONE
XmNorientation	integer	XmHORIZONTAL

UIL Argument Name	Argument Type	Default Value
XmNsensitive	boolean	True
XmNseparatorType	integer	XmSHADOW_ETCHED_IN
XmNshadowThickness	integer	2
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.51 XmText

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNgainPrimaryCallback
	XmNhelpCallback
	XmNlosePrimaryCallback
	XmNlosingFocusCallback
	XmNmodifyVerifyCallback
	XmNmotionVerifyCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNautoShowCursorPosition	boolean	True
XmNbackground	color	White

UIL Argument Name	Argument Type	Default Value
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNblinkRate	integer	500
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNcolumns	integer	20
XmNcursorPosition	integer	0
XmNcursorPositionVisible	boolean	True
XmNdepth	identifier	dynamic
XmNeditMode	integer	XmSINGLE_LINE_EDIT
XmNeditable	boolean	True
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	3
XmNmarginWidth	integer	3
XmNmaxLength	integer	MAXINT
XmNnavigationType	integer	XmTAB_GROUP
XmNpendingDelete	boolean	True
XmNresizeHeight	boolean	False

UIL Argument Name	Argument Type	Default Value
XmNresizeWidth	boolean	False
XmNrows	integer	1
XmNscreen	identifier	dynamic
XmNselectThreshold	integer	5
XmNselectionArray	any	array of XmTextScanType: XmSELECT_POSITION, XmSELECT_WORD, XmSELECT_LINE, XmSELECT_ALL
XmNselectionArrayCount	integer	4
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNtopCharacter	integer	0
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	Null
XmNverifyBell	boolean	True
XmNwidth	integer	0
XmNwordWrap	boolean	False
XmNx	integer	0
XmNy	integer	0

A.52 XmTextField

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNactivateCallback
	XmNdestroyCallback

Controls	Reasons
	XmNgainPrimaryCallback
	XmNhelpCallback
	XmNlosePrimaryCallback
	XmNlosingFocusCallback
	XmNmodifyVerifyCallback
	XmNmotionVerifyCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNblinkRate	integer	500
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNcolumns	integer	20
XmNcursorPosition	integer	0
XmNcursorPositionVisible	boolean	True
XmNdepth	identifier	dynamic
XmNeditable	boolean	True
XmNfontList	font_table	Fixed
XmNforeground	color	dynamic
XmNheight	integer	0

A.52 XmTextField

UIL Argument Name	Argument Type	Default Value
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNinitialResourcesPersistent	boolean	True
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	3
XmNmarginWidth	integer	3
XmNmaxLength	integer	MAXINT
XmNnavigationType	integer	XmTAB_GROUP
XmNpendingDelete	boolean	True
XmNresizeWidth	boolean	False
XmNscreen	identifier	dynamic
XmNselectThreshold	integer	5
XmNselectionArray	any	array of XmTextScanType: XmSELECT_POSITION, XmSELECT_WORD, XmSELECT_LINE, XmSELECT_ALL
XmNselectionArrayCount	integer	4
XmNsensitive	boolean	True
XmNshadowThickness	integer	2
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvalue	any	Null
XmNverifyBell	boolean	True
XmNwidth	integer	0
XmNx	integer	0

UIL Argument Name	Argument Type	Default Value
XmNy	integer	0

A.53 XmToggleButton

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNhelpCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNaccelerators	translation_table	Null
XmNacceleratorText	compound_string	Null
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNfillOnSelect	boolean	True
XmNfontList	font_table	Fixed

A.53 XmToggleButton

UIL Argument Name	Argument Type	Default Value
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhighlightColor	color	Black
XmNhighlightOnEnter	boolean	False
XmNhighlightPixmap	pixmap	dynamic
XmNhighlightThickness	integer	0
XmNindicatorOn	boolean	True
XmNindicatorSize	integer	XmINVALID_DIMENSION
XmNindicatorType	integer	XmN_OF_MANY
XmNinitialResourcesPersistent	boolean	True
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmappedWhenManaged	boolean	True
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNscreen	identifier	dynamic
XmNselectColor	color	dynamic
XmNselectInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNselectPixmap	pixmap	XmUNSPECIFIED_ PIXMAP

UIL Argument Name	Argument Type	Default Value
XmNsensitive	boolean	True
XmNset	boolean	False
XmNshadowThickness	integer	0
XmNspacing	integer	4
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvisibleWhenOff	boolean	True
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.54 XmToggleButtonGadget

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNarmCallback
	XmNdestroyCallback
	XmNdisarmCallback
	XmNhelpCallback
	XmNvalueChangedCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerator	string	Null
XmNacceleratorText	compound_string	Null

A.54 XmToggleButtonGadget

UIL Argument Name	Argument Type	Default Value
XmNalignment	integer	XmALIGNMENT_CENTER
XmNancestorSensitive	boolean	True
XmNborderWidth	integer	1
XmNfillOnSelect	boolean	True
XmNfontList	font_table	Fixed
XmNheight	integer	0
XmNhighlightOnEnter	boolean	False
XmNhighlightThickness	integer	0
XmNindicatorOn	boolean	True
XmNindicatorSize	integer	XmINVALID_DIMENSION
XmNindicatorType	integer	XmN_OF_MANY
XmNlabelInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNlabelString	compound_string	Null
XmNlabelType	integer	XmSTRING
XmNmarginBottom	integer	0
XmNmarginHeight	integer	2
XmNmarginLeft	integer	0
XmNmarginRight	integer	0
XmNmarginTop	integer	0
XmNmarginWidth	integer	2
XmNmnemonic	keysym	NULL
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	XmNONE
XmNrecomputeSize	boolean	True
XmNselectColor	color	dynamic
XmNselectInsensitivePixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNselectPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNsensitive	boolean	True

UIL Argument Name	Argument Type	Default Value
XmNset	boolean	False
XmNshadowThickness	integer	0
XmNspacing	integer	4
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNvisibleWhenOff	boolean	True
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.55 XmWarningDialog

Controls	Reasons
No children are supported	MrmNcreateCallback
	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White

A.55 XmWarningDialog

UIL Argument Name	Argument Type	Default Value
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING

UIL Argument Name	Argument Type	Default Value
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.56 XmWorkArea

Controls	Reasons
AllWidgetsAndGadgets	MrmNcreateCallback
	XmNdestroyCallback
	XmNentryCallback

A.56 XmWorkArea

Controls	Reasons
	XmNhelpCallback
	XmNmapCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNadjustLast	boolean	True
XmNadjustMargin	boolean	True
XmNancestorSensitive	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderWidth	integer	0
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNcolormap	identifier	dynamic
XmNdepth	identifier	dynamic
XmNentryAlignment	integer	dynamic
XmNentryBorder	integer	dynamic
XmNentryClass	class_rec_name	dynamic
XmNforeground	color	dynamic
XmNheight	integer	dynamic
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNisAligned	boolean	True

UIL Argument Name	Argument Type	Default Value
XmNisHomogeneous	boolean	dynamic
XmNlabelString	compound_string	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	dynamic
XmNmarginWidth	integer	3
XmNmenuAccelerator	string	dynamic
XmNmenuHelpWidget	widget_ref	Null
XmNmenuHistory	widget_ref	Null
XmNmenuPost	string	Null
XmNmnemonic	keysym	dynamic
XmNmnemonicCharSet	string	dynamic
XmNnavigationType	integer	dynamic
XmNnumColumns	integer	dynamic
XmNorientation	integer	dynamic
XmNpacking	integer	dynamic
XmNpopupEnabled	boolean	True
XmNradioAlwaysOne	boolean	True
XmNradioBehavior	boolean	False
XmNresizeHeight	boolean	True
XmNresizeWidth	boolean	True
XmNrowColumnType	integer	XmWORK_AREA
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNspacing	integer	dynamic
XmNstringDirection	integer	XmSTRING_DIRECTION_ L_TO_R
XmNsubMenuId	widget_ref	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNtranslations	translation_table	Null

UIL Argument Name	Argument Type	Default Value
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwhichButton	integer	dynamic
XmNwidth	integer	dynamic
XmNx	integer	0
XmNy	integer	0

A.57 XmWorkingDialog

Controls	Reasons
XmPushButton	MrmNcreateCallback
user_defined	XmNcancelCallback
	XmNdestroyCallback
	XmNfocusCallback
	XmNhelpCallback
	XmNmapCallback
	XmNokCallback
	XmNunmapCallback

UIL Argument Name	Argument Type	Default Value
XmNaccelerators	translation_table	Null
XmNallowOverlap	boolean	True
XmNancestorSensitive	boolean	True
XmNautoUnmanage	boolean	True
XmNbackground	color	White
XmNbackgroundPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNborderColor	color	Black
XmNborderPixmap	pixmap	XmUNSPECIFIED_ PIXMAP

UIL Argument Name	Argument Type	Default Value
XmNborderWidth	integer	1
XmNbottomShadowColor	color	dynamic
XmNbottomShadowPixmap	pixmap	XmUNSPECIFIED_ PIXMAP
XmNbuttonFontList	font_table	Null
XmNcancelButton	widget_ref	Null
XmNcancelLabelString	compound_string	Cancel
XmNcolormap	identifier	dynamic
XmNdefaultButton	widget_ref	Null
XmNdefaultButtonType	integer	XmDIALOG_OK_BUTTON
XmNdefaultPosition	boolean	True
XmNdepth	identifier	dynamic
XmNdialogStyle	integer	dynamic
XmNdialogTitle	compound_string	Null
XmNdialogType	integer	XmDIALOG_MESSAGE
XmNforeground	color	dynamic
XmNheight	integer	0
XmNhelpLabelString	compound_string	Help
XmNhighlightColor	color	Black
XmNhighlightPixmap	pixmap	dynamic
XmNinitialResourcesPersistent	boolean	True
XmNinsertPosition	identifier	Null
XmNlabelFontList	font_table	Null
XmNmappedWhenManaged	boolean	True
XmNmarginHeight	integer	10
XmNmarginWidth	integer	10
XmNmessageAlignment	integer	XmALIGNMENT_ BEGINNING
XmNmessageString	compound_string	Null
XmNminimizeButtons	boolean	False
XmNnavigationType	integer	XmTAB_GROUP
XmNnoResize	boolean	False

A.57 XmWorkingDialog

UIL Argument Name	Argument Type	Default Value
XmNokLabelString	compound_string	OK
XmNresizePolicy	integer	XmRESIZE_ANY
XmNscreen	identifier	dynamic
XmNsensitive	boolean	True
XmNshadowThickness	integer	0
XmNshadowType	integer	XmSHADOW_OUT
XmNstringDirection	integer	XmSTRING_DIRECTION_L_TO_R
XmNsymbolPixmap	pixmap	XmDEFAULT_PIXMAP
XmNtextFontList	font_table	Null
XmNtextTranslations	translation_table	Null
XmNtopShadowColor	color	dynamic
XmNtopShadowPixmap	pixmap	XmUNSPECIFIED_PIXMAP
XmNtranslations	translation_table	Null
XmNtraversalOn	boolean	False
XmNunitType	integer	XmPIXELS
XmNuserData	any	Null
XmNwidth	integer	0
XmNx	integer	0
XmNy	integer	0

A.58 user_defined

Controls	Reasons
AllWidgets	

A.59 All Widgets and Gadgets

- DXmColorMix
- DXmColorMixDialog
- DXmCSText
- DXmHelpDialog

- DXmPrintBox
- DXmPrintDialog
- DXmScrolledCSText
- DXmSvn
- XmArrowButton
- XmArrowButtonGadget
- XmBulletinBoard
- XmBulletinBoardDialog
- XmCascadeButton
- XmCascadeButtonGadget
- XmCommand
- XmDrawingArea
- XmDrawnButton
- XmErrorDialog
- XmFileSelectionBox
- XmFileSelectionDialog
- XmForm
- XmFormDialog
- XmFrame
- XmInformationDialog
- XmLabel
- XmLabelGadget
- XmList
- XmMenuBar
- XmMessageBox
- XmMessageDialog
- XmOptionMenu
- XmPanedWindow
- XmPopupMenu

- XmPromptDialog
- XmPulldownMenu
- XmPushButton
- XmPushButtonGadget
- XmQuestionDialog
- XmRadioBox
- XmRowColumn
- XmScale
- XmScrollBar
- XmScrolledList
- XmScrolledText
- XmScrolledWindow
- XmSelectionBox
- XmSelectionDialog
- XmSeparator
- XmSeparatorGadget
- XmText
- XmTextField
- XmToggleButton
- XmToggleButtonGadget
- XmWarningDialog
- XmWorkArea
- XmWorkingDialog
- user_defined

A.60 All Widgets

- DXmColorMix
- DXmColorMixDialog
- DXmCSText
- DXmHelpDialog
- DXmPrintBox
- DXmPrintDialog
- DXmScrolledCSText
- DXmSvn
- XmArrowButton
- XmBulletinBoard
- XmBulletinBoardDialog
- XmCascadeButton
- XmCommand
- XmDrawingArea
- XmDrawnButton
- XmErrorDialog
- XmFileSelectionBox
- XmFileSelectionDialog
- XmForm
- XmFormDialog
- XmFrame
- XmInformationDialog
- XmLabel
- XmList
- XmMenuBar
- XmMessageBox
- XmMessageDialog

- XmOptionMenu
- XmPanedWindow
- XmPopupMenu
- XmPromptDialog
- XmPulldownMenu
- XmPushButton
- XmQuestionDialog
- XmRadioBox
- XmRowColumn
- XmScale
- XmScrollBar
- XmScrolledList
- XmScrolledText
- XmScrolledWindow
- XmSelectionBox
- XmSelectionDialog
- XmSeparator
- XmText
- XmTextField
- XmToggleButton
- XmWarningDialog
- XmWorkArea
- XmWorkingDialog
- user_defined

B

OpenVMS DECwTermPort Routine

This appendix describes the DECwTermPort routine, which allows you to create a DECterm window on a local or remote node. This routine works only on OpenVMS systems.

DECwTermPort

Creates a DECterm window on a local or remote node.

Format

```
int DECwTermPort(display,setup_file,customization,result_dev,
                 result_len[,controller][,char_buff]
                 [,char_chng_buff])
      char              *display;
      char              *setup_file;
      char              *customization;
      char              *result_dev;
      short             *result_len;
      char              *controller;
      struct tt_chars   *char_buff;
      struct tt_chars   *char_chng_buff;
```

Return Value

An OpenVMS status code that indicates the success or failure of the window creation operation. The following table lists the the status codes most commonly returned:

Status Code	Description
SS$_NORMAL	Window was created successfully.
CANT_OPEN_DISPLAY	Window display could not be opened.
CANT_OPEN_DRM	DRM database could not be opened.
TIMEOUT_CONTROLLER	Timeout occurred before the DECterm controller could start.

Arguments

display

A character string that identifies the server and screen on which the created DECterm appears. If the string address is 0, the default display is used.

setup_file

A character string that specifies the name of the setup file. The setup file changes DECterm's initial settings. (See the **customization** argument for

information about the syntax of a setup file.) If the string address is 0, the default setup file, DECW$USER_DEFAULTS:DECW$TERMINAL_DEFAULT.DAT, is used.

customization

A character string that specifies setup options that override the default values established in resource and setup files. If the string address is 0, default values are not to be overridden. The syntax is the same as the syntax for resource and setup files:

```
"param:  value \n param:  value \n param:  value...."
```

You can create a customization file using the Application menu of the session manager. To do this, create a DECterm using the Applications menu, use the Customize option to change settings, and save the new settings in a file (be sure to use a nondefault file name). You can use the name of this file as the value of the **customization** argument.

result_dev

A character string that specifies the virtual terminal device name for the created DECterm. This argument is intended for applications that want to assign the created DECterm or pass the name to a new process.

result_len

The address of a 16-bit word into which the length of the returned device name is written.

controller

An optional argument that is a character string that specifies which controller should be used with the DECterm window. For example, you can specify a foreign language variant of DECterm. The default is SYS$SYSTEM:DECW$TERMINAL.EXE.

char_buff

An optional argument that is the address of a 12-byte terminal characteristic buffer in which the terminal characteristics of the DECterm are specified. See the *VMS I/O User's Reference Manual: Part I* for further information.

char_chng_buff

An optional argument that is the address of a 12-byte terminal characteristic buffer that specifies which characteristics are set in the **char_buff** argument. This argument must be specified with the **char_buff** argument. Only those terminal characteristics that have nonzero values in the **char_chng_buff** buffer are set to the values specified in the **char_buff** argument. Otherwise,

the terminal characteristic is not changed. See the *VMS I/O User's Reference Manual: Part I* for further information.

Description

The DECwTermPort routine creates a DECterm window on a local or remote node, returning an OpenVMS status code that indicates the success or failure of the window creation operation.

C

SVN and Compound String Text Widget Translations

This appendix contains Structured Visual Navigation (SVN) and Compound String Text Widget translation information. Table C–1 describes the button syntax translations for the DXmSvn widget.

Table C–1 DXmSvn Widget Button Syntax Translations

Event	Action Routine
BSelectPress:	svn_button1_down
BSelectRelease:	svn_button1_up
BSelectDrag:	svn_button1_motion
BExtendPress:	svn_shiftbutton1_down
BExtendRelease:	svn_button1_up
BExtendDrag:	svn_button1_motion
BTogglePress:	svn_ctrlbutton1_down
BToggleRelease:	svn_button1_up
BToggleDrag:	svn_button1_motion
BDragPress:	svn_button2_down
BDragRelease:	svn_button2_up
BDragDrag:	svn_button2_motion
BDragToggle:	(MB2-ctrl-click) svn_button2_down
BDrag:	(MB2-Alt-click) svn_button2_down
BMenu:	svn_button3_down
BFocusIn:	svn_focus_in

(continued on next page)

SVN and Compound String Text Widget Translations

Table C–1 (Cont.) DXmSvn Widget Button Syntax Translations

Event	Action Routine
BFocusOut:	svn_focus_out
BHelp:	svnhelp

Table C–2 describes the keyboard syntax translations for the DXmSvn widget.

Table C–2 DXmSvn Widget Keyboard Syntax Translations

Event	Action Routine
KMenu:	svn_f4_menu
KDown:	svn_arrow_down
KUp:	svn_arrow_up
KPageDown:	svn_page_down
KPageUp:	svn_page_up
KEndData:	svn_scroll_to_top
KBeginData:	svn_scroll_to_bottom
KExtendUp:	svn_extend_up
KExtendDown:	svn_extend_down
MShiftKPageUp:	svn_extend_pageup
MShiftKPageDown:	svn_extend_pagedown
MShiftKBeginData:	svn_extend_to_top
MShiftKEndData:	svn_extend_to_bottom
KLeft:	svn_arrow_left
KRight:	svn_arrow_right
KBeginLine:	svn_scroll_to_left
KEndLine:	svn_scroll_to_right
KPageLeft:	svn_page_left
KPageUp:	svn_page_right
KActivate:	svn_activate_entry
MCtrlKActivate:	svn_activate_entry

(continued on next page)

Table C–2 (Cont.) DXmSvn Widget Keyboard Syntax Translations

Event	Action Routine
KSpace:	svn_activate_entry
MCtrlKSpace:	svn_activate_entry
KSelect:	svn_activate_entry
KHelp:	svnhelp
KSelectAll:	svn_select_all
KDeSelectAll:	svn_deselect_all
KNextColumn:	svn_next_column
KPrevColumn:	svn_prev_column
MShiftKTab:	LclTraversePrevTabGroup
KTab:	LclTraverseNextTabGroup
MCtrlKTab:	LclTraversePrevTabGroup
KCancel:	svn_cancel
KNextPara:	svn_next_level
KPrevPara:	svn_prev_level

Table C–3 further describes the action routines for the DXmSvn widget.

Table C–3 Action Routines for DXmSvn Widget

Action Routine	Description
svn_button1_down	Selects and moves the location cursor to the current entry. Deselects all other entries.
svn_button1_up	Completes selection started by the button1 press. Reports the DXmSvnNentrySelectedCallback to the application if specified. If this BtnUp was a completion of a double MB1 click, then the DXmSvnNselectAndConfirmCallback will be reported to the application, if specified.
svn_button1_motion	Extends selection to include entries that are passed over with the mouse motion. If the Ctrl modifier was pressed with this MB1 motion, then all entries passed over with the mouse selection state are toggled.

(continued on next page)

Table C–3 (Cont.) Action Routines for DXmSvn Widget

Action Routine	Description
svn_shiftbutton1_down	Extends selection to include all entries between the last selected entry and this current entry. Moves the location cursor to this current entry.
svn_ctrlbutton1_down	Toggles the selection state of the current entry and moves the location cursor to this entry.
svn_button2_down	Initiates a drag operation. If MB2 is clicked on a selected entry, all selected entries are dragged. If MB2 is clicked on a unselected entry, only that entry is dragged. If application dragging is set, via the DXmSvnSetApplDragging routine, the DXmSvnNdraggingCallback is reported, if specified.
svn_button2_up	Completes the drag operation. The DXmSvnNselectionsDragged callback is reported to the application if specified and the entries are selected. If application dragging is set, via the DXmSvnSetApplDragging routine, the DXmSvnNdraggingEndCallback is reported, if specified. If the MB2 click is made without mouse motion, the DXmSvnNentryTransferCallback is reported to the application, if specified.
svn_button2_motion	A ghost object is created and follows the mouse with MB2 motion.
svn_button3_down	Reports DXmSvnNpopupMenu callback to the application if specified for this entry.
svn_focus_in	Sets focus to the SVN widget. Moves location cursor to last selected entry in display.
svn_focus_out	Removes focus from the SVN widget. Also removes location cursor from last selected entry.
svnhelp	Reports the help callback DXmSvnNhelpCallback to the application, if specified.
svn_f4_menu	Reports the DXmNpopupMenuCallback to the application if specified for the entry with the location cursor.
svn_arrow_down	Selects and moves the location cursor to the next entry in the display, scrolling the display down, if necessary.

(continued on next page)

Table C–3 (Cont.) Action Routines for DXmSvn Widget

Action Routine	Description
svn_arrow_up	Selects and moves the location cursor to the previous entry in the display, scrolling the display up, if necessary.
svn_page_down	Scrolls the display to the next screen of entries. Selects and moves the location cursor to the entry located at the same relative position on the next screen.
svn_page_up	Scrolls the display to the previous screen of entries. Selects and moves the location cursor to the entry located at the same relative position on the previous screen.
svn_scroll_to_top	Scrolls to the first entry in the SVN display. Selects and moves the location cursor to that entry.
svn_scroll_to_bottom	Scrolls to the last entry in the SVN display. Selects and moves the location cursor to that entry.
svn_extend_up	Adds the previous entry to the extended selection. Selects and moves the location cursor to the previous entry, scrolling the display, if necessary.
svn_extend_down	Adds the next entry to the extended selection. Selects and moves the location cursor to that next entry, scrolling the display, if necessary.
svn_extend_pageup	Extends the selection to include all entries in the next screen. Moves the location cursor to the entry located at the same relative position on the next screen.
svn_extend_pagedown	Extends the selection to include all entries in the previous screen. Moves the location cursor to the entry located at the same relative position on the previous screen.
svn_extend_to_top	Extends selection from the currently selected entry to the top entry known to SVN. Moves the location cursor to that top entry.
svn_extend_to_bottom	Extends selection from the currently selected entry to the bottom entry known to SVN. Moves the location cursor to that bottom entry.
svn_arrow_left	Scrolls the display one character to the left.
svn_arrow_right	Scrolls the display one character to the right.

(continued on next page)

Table C–3 (Cont.) Action Routines for DXmSvn Widget

Action Routine	Description
svn_scroll_to_left	Scrolls to the extreme left of the display.
svn_scroll_to_right	Scrolls to the extreme right of the display.
svn_page_left	Scrolls the display horizontally one screen width to the left.
svn_page_right	Scrolls the display horizontally one screen width to the right.
svn_activate_entry	Selects and confirms the current entry. Equivalent to double clicking MB1.
svn_select_all	Selects all entries known to SVN.
svn_deselect_all	Deselects all entries known to SVN.
svn_next_column	Selects the next column component to the right.
svn_prev_column	Selects the previous column component to the left.
LclTraversePrevTabGroup	Moves the location cursor (focus) to the previous Tab group. In SVN, there are primary and secondary windows that are both scrollable and in separate Tab groups. This function allows the focus to move to another window in SVN.
LclTraverseNextTabGroup	Moves the location cursor to the next Tab group. In SVN, there are primary and secondary windows that are both scrollable and in separate Tab groups. This function allows the focus to move to other window in SVN.
svn_cancel	Cancels any dragging operation or extended selection in progress. Moves the location cursor to the entry it was on before the operation started.
svn_next_level	Scrolls to the entry at the next level in the SVN hierarchy.
svn_prev_level	Scrolls to the entry at the previous level in the SVN hierarchy.

Table C–4 describes the button syntax translations for the DXmCSText widget.

Table C–4 DXmCSText Widget Button Syntax Translations

Event	Action Routine
BSelect Press:	grab-focus( )
BSelect Motion:	extend-adjust( )
BSelect Release:	extend-end( )
BExtend Press:	extend-start( )
BExtend Motion:	extend-adjust( )
BExtend Release:	extend-end( )
BToggle Press:	move-destination( )
BDrag Press:	secondary-start( )
BDrag Motion:	secondary-adjust( )
BDrag Release:	copy-to( )
MCtrl BDrag Press:	secondary-start( )
MCtrl BDrag Motion:	secondary-adjust( )
MCtrl BDrag Release:	copy-to( )
MAlt BDrag Press:	secondary-start( )
MAlt BDrag Motion:	secondary-adjust( )
MAlt BDrag Release:	move-to( )
KUp:	process-up( )
MShift KUp:	process-shift-up( )
MCtrl KUp:	backward-paragraph( )
MShift MCtrl KUp:	backward-paragraph(extend)
KDown:	process-down( )
MShift KDown:	process-shift-down( )
MCtrl KDown:	forward-paragraph( )
MShift MCtrl KDown:	forward-paragraph(extend)
KLeft:	left-character( )
MShift KLeft:	key-select(left)
MCtrl KLeft:	left-word( )
MShift MCtrl KLeft:	left-word(extend)
KRight:	right-character( )

(continued on next page)

Table C–4 (Cont.) DXmCSText Widget Button Syntax Translations

Event	Action Routine
MShift KRight:	key-select(right)
MCtrl KRight:	right-word( )
MShift MCtrl KRight:	right-word(extend)
KPageUp:	previous-page( )
MShift KPageUp:	previous-page(extend)
KPageDown:	next-page( )
MShift KPageDown:	next-page(extend)
MCtrl KPageUp:	page-horizontal-backward( )
MCtrl KPageDown:	page-horizontal-forward( )
MAlt KLeft:	left-side-of-line( )
MShift Malt KLeft:	left-side-of-line(extend)
MAlt KRight:	right-side-of-line( )
MShift MAlt KRight:	right-side-of-line(extend)
MCtrl MAlt KLeft:	extreme-left-of-file( )
MShift MCtrl MAlt KLeft:	extreme-left-of-file(extend)
MCtrl MAlt KRight:	extreme-right-of-file( )
MShift MCtrl MAlt KRight:	extreme-right-of-file(extexd)
Malt <Key>F17:	toggle-text-path
<Key>F17:	toggle-editing-path-move( )
MShift <Key>F17:	toggle-editing-path-stay
KTab:	process-tab( )
KNextField:	next-tab-group( )
KPrevField:	prev-tab-group( )
KEnter:	process-return( )
KActivate:	activate( )
KDelete:	delete-next-character( )
KBackSpace:	delete-previous-character( )
KAddMode:	toggle-add-mode( )
KSpace:	self-insert( )

(continued on next page)

Table C–4 (Cont.) DXmCSText Widget Button Syntax Translations

Event	Action Routine
KSelect:	set-anchor( )
KExtend:	key-select( )
MAny KCancel:	process-cancel( )
KClear:	clear-selection( )
KSelectAll:	select-all( )
KDeselectAll:	deselect-all( )
KCut:	cut-clipboard( )
KCopy:	copy-clipboard( )
KPaste:	paste-clipboard( )
KPrimaryCut:	cut-primary( )
KPrimaryCopy:	copy-primary( )
KPrimaryPaste:	copy-primary( )
KQuickCut:	quick-cut-set( )
KQuickCopy:	quick-copy-set( )
KQuickPaste:	quick-copy-set( )
KQuickExtend:	do-quick-action( )
KHelp:	Help( )
KAny:	self-insert( )

Table C–5 further describes the action routines for the DXmCSText widget.

Table C–5 DXmCSText Widget Action Routines

Action Routine	Description
activate( ):	Calls the callbacks for XmNactivateCallback.
backward-character( ):	Moves the insertion cursor one character to the left.
backward-paragraph(extend):	If XmNeditMode is XmMULTI_LINE_EDIT and this action is called with no argument, moves the insertion cursor to the first nonwhitespace character following the first previous blank line or beginning of the text.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
	If the insertion cursor is already at the beginning of a paragraph, moves the insertion cursor to the beginning of the previous paragraph. If XmNeditMode is XmMULTI_LINE_EDIT and this action is called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
backward-word(extend):	If this action is called with no argument, moves the insertion cursor to the first nonwhitespace character after the first whitespace character to the left or the beginning of the line. If the insertion cursor is already at the beginning of a word, moves the insertion cursor to the beginning of the previous word. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
beep( ):	Causes the terminal to beep.
beginning-of-file(extend):	If this action is called with no argument, moves the insertion cursor to the beginning of the text. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
beginning-of-line(extend):	If this action is called with no argument, moves the insertion cursor to the beginning of the line. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
clear-selection( ):	Clears the current selection by replacing each character, except <Return>, with a <space> character.
copy-clipboard( ):	Copies the current selection to the clipboard.
copy-primary( ):	Copies the primary selection to the insertion cursor.
copy-to( ):	If a secondary selection exists, copies the secondary selection to the insertion cursor.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
	If no secondary selection exists, copies the primary selection to the pointer location.
cut-clipboard( ):	Cuts the current selection to the clipboard.
cut-primary( ):	Cuts the primary selection to the insertion cursor.
delete-next-character( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection; otherwise, deletes the character following the insert cursor.
delete-next-word( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection; otherwise, deletes the characters following the insertion cursor to the next space, tab, or end of line character.
delete-previous-character( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection; otherwise, deletes the character of text immediately preceding the insertion cursor.
delete-previous-word( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection; otherwise, deletes the characters preceding the insertion cursor to the next space, tab, or beginning of line character.
delete-selection( ):	Deletes the current selection.
delete-to-end-of-line( ):	Deletes the characters following the insertion cursor to the next end of line character.
delete-to-start-of-line( ):	Deletes the characters preceding the insertion cursor to the previous beginning of line character.
deselect-all( ):	Deselects the current selection.
do-quick-action( ):	Marks the end of a secondary selection. Performs the quick action initiated by the quick-copy-set or quick-cut-set action.
end-of-file(extend):	If this action is called with no argument, moves the insertion cursor to the end of the text.
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
end-of-line(extend):	If this action is called with no argument, moves the insertion cursor to the end of the line.
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
extend-adjust( ):	Selects text from the anchor to the pointer position and deselects text outside that range. Moving the pointer over several lines selects text from the anchor to the end of each line over which the pointer moves and up to the pointer position on the current line.
extend-end( ):	Moves the insertion cursor to the position of the pointer.
extend-start( ):	Adjusts the anchor using the balance-beam method. Selects text from the anchor to the pointer position and deselects text outside that range.
extreme-left-of-file(extend):	If this action is called with no argument and the current text path is left to right, moves the insertion cursor to the beginning of the text.
	If called with no argument and the current text path is right to left, moves the insertion cursor to the end of the text.
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
	The text path can be initially set using the DXmNtextPath resource or interactively (see toggle-text-path).
extreme-right-of-file(extend):	If this action is called with no argument and the current text path is left to right, moves the insertion cursor to the end of the text.
	If called with no argument and the current text path is right to left, moves the insertion cursor to the beginning of the text.
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
	The text path can be set initially using the DXmNtextPath resource or interactively (see toggle-text-path).
forward-character( ):	Moves the insertion cursor one character to the right.
forward-paragraph(extend):	If XmNeditMode is XmMULTI_LINE_EDIT, and this action is called with no argument, moves the insertion cursor to the first nonwhitespace character following the next blank line. If the insertion cursor is already at the beginning of a paragraph, moves the insertion cursor to the beginning of the next paragraph.
	If XmNeditMode is XmMULTI_LINE_EDIT and this action is called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
forward-word(extend):	If this action is called with no argument, moves the insertion cursor to the first whitespace character or end of line following the next nonwhitespace character. If the insertion cursor is already at the end of a word, moves the insertion cursor to the end of the next word.
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
grab-focus( ):	This key binding performs the action in the XmNselectArray, depending on the number of multiple mouse clicks. The default selection array ordering is one click to move the insertion cursor to the pointer position, two clicks to select a word, three clicks to select a line of text, and four clicks to select all text.
Help( ):	Calls the callbacks for XmNhelpCallback if any exist. If there are no help callbacks for this widget, this action calls the help callbacks for the nearest ancestor that has them.
insert-string(string):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Inserts "string" at the insertion cursor.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
key-select(direction):	If called with an argument of "right", moves the insertion cursor one character to the right and extends the selection. If called with an argument of "left", moves the insertion cursor one character to the left and extends the selection. If called with no argument, extends the selection.
kill-next-character( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Otherwise, kills the character following the insertion cursor and stores the character in the cut buffer.
kill-next-word( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Otherwise, kills the characters following the insertion cursor to the next space, tab, or end of line character, and stores the characters in the cut buffer.
kill-previous-character( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Otherwise, kills the character of text immediately preceding the insertion cursor and stores the character in the cut buffer.
kill-previous-word( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Otherwise, kills the characters preceding the insertion cursor to the next space, tab, or beginning of line character, and stores the characters in the cut buffer.
kill-selection( ):	Kills the currently selected text and stores the text in the cut buffer.
kill-to-end-of-line( ):	Kills the characters following the insertion cursor to the next end of line character and stores the characters in the cut buffer.
kill-to-start-of-line( ):	Kills the characters preceding the insertion cursor to the next beginning of line character and stores the characters in the cut buffer.
left-character( ):	Moves the insertion cursor one character to the left.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
left-side-of-line(extend):	If this action is called with no argument and the current text path is left to right, moves the insertion cursor to the beginning of the line. If called with no argument and the current text path is right to left, moves the insertion cursor to the end of the line. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection. Set the text path initially (using the DXmNtextPath resource) or interactively (see toggle-text-path).
left-word(extend):	If this action is called with no argument and the current text path is left to right, moves the insertion cursor to the first nonwhitespace character after the first whitespace character to the left or to the beginning of the line. If the insertion cursor is already at the beginning of a word, moves the insertion cursor to the beginning of the previous word. If this action is called with no argument and the current text path is right to left, moves the insertion cursor to the first whitespace character or end of line following the next nonwhitespace character. If the insertion cursor is already at the end of a word, moves the insertion cursor to the end of the next word. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection. Set the text path initially (using the DXmNtextPath resource) or interactively (see toggle-text-path).
move-destination( ):	Moves the insertion cursor to the pointer position without changing any existing selection. If no selection exists, also moves the destination cursor to the pointer position.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
move-to( ):	If a secondary selection exists, cuts the secondary selection to the insertion cursor. If no secondary selection exists, cuts the primary selection to the pointer location.
new-line( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Inserts a newline at the insertion cursor.
newline-and-backup( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Inserts a newline and repositions the insertion cursor to the end of the line before the newline.
newline-and-indent( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Inserts a newline and then the same number of whitespace characters as at the beginning of the previous line.
next-line( ):	Moves the insertion cursor to the next line.
next-page(extend):	If this action is called with now argument, moves the insertion cursor forward one page. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
next-tab-group( ):	Traverses to the next tab group.
page-horizontal-backward( ):	If the current text path is left to right, scrolls the viewing window left one path of text. If the current text path is right to left, scrolls the viewing window right one path of text. Set the text path initially (using the DXmNtextPath resource) or interactively (see toggle-text-path).
page-horizontal-forward( ):	If the current text path is left to right, scrolls the viewing window right one path of text. If the current text path is right to left, scrolls the viewing window left one path of text. Set the text path initially (using the DXmNtextPath resource) or interactively (see toggle-text-path).
page-left( ):	Scrolls the viewing window left one page of text.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
page-right( ):	Scrolls the viewing window right one page of text.
paste-clipboard( ):	Pastes the contents of the clipboard before the insertion cursor.
prev-tab-group( ):	Traverses to the previous tab group.
previous-line( ):	Moves the insertion cursor to the previous line.
previous-page(extend):	If this action is called with no argument, moves the insertion cursor back one page. If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
process-cancel( ):	Cancels the current extend-adjust( ) or secondary-adjust( ) operation and leaves the selection state as it was before the operation.
process-down( ):	If XmNeditMode is XmSINGLE_LINE_EDIT and XmNnavigationType is XmNONE, traverses to the widget below the current one in the tab group. If XmNeditMode is XmMULTI_LINE_EDIT, moves the insertion cursor down one line.
process-home( ):	Moves the insertion cursor to the beginning of the line.
process-return( ):	If XmNeditMode is XmSINGLE_LINE_EDIT, calls the callbacks for XmNactivateCallback. If XmNeditMode is XmMULTI_LINE_EDIT, inserts a new line.
process-shift-down( ):	If XmNeditMode is XmMULTI_LINE_EDIT, moves the insertion cursor down one line.
process-shift-up( ):	If XmNeditMode is XmMULTI_LINE_EDIT, moves the insertion cursor up one line.
process-tab( ):	If XmNeditMode is XmSINGLE_LINE_EDIT, traverses to the next tab group. If XmNeditMode is XmMULTI_LINE_EDIT, inserts a tab.
process-up( ):	If XmNeditMode is XmSINGLE_LINE_EDIT and XmNnavigationType is XmNONE, traverses to the widget above the current one in tab group.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
	If XmNeditMode is XmMULTI_LINE_EDIT, moves the insertion cursor up one line.
quick-copy-set( ):	Marks the beginning of a secondary selection for use in quick copy.
quick-cut-set( ):	Marks the beginning of a secondary selection for use in quick cut.
redraw-display( ):	Redraws the contents of the text windows.
right-character( ):	Moves the insertion cursor one character to the right.
right-side-of-line(extend):	If this action is called with no argument and the current text path is left to right, moves the insertion cursor to the end of the line.
	If called with no argument and the current text path is right to left, moves the insertion cursor to the beginning of the line.
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
	Set the text path initially (using the DXmNtextPath resource) or interactively (see toggle-text-path).
right-word(extend):	If this action is called with no argument and the current text path is left to right, moves the insertion cursor to the first whitespace character or end of line following the next nonwhitespace character. If the insertion cursor is already at the end of a word, moves the insertion cursor to the end of the next word.
	If this action is called with no argument and the current text path is right to left, moves the insertion cursor to the first nonwhitespace character after the first whitespace character to the left or the beginning of the line. If the insertion cursor is already at the beginning of a word, moves the insertion cursor to the beginning of the previous word.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
	If called with **extend**, moves the insertion cursor as in the case of no argument and extends the selection.
	Set the text path initially (using the DXmNtextPath resource) or interactively (see toggle-text-path).
scroll-one-line-down( ):	Scrolls the text area down one line.
scroll-one-line-up( ):	Scrolls the text area up one line.
secondary-adjust( ):	Extends the secondary selection to the pointer position.
secondary-notify( ):	Copies the secondary selection to the destination cursor.
secondary-start( ):	Marks the beginning of a secondary selection.
select-adjust( ):	Extends the selection. The amount of text selected depends on the number of mouse clicks, as specified by the XmNselectionArray resource.
select-all( ):	Select all text.
select-end( ):	Extends the selection. The amount of text selected depends on the number of mouse clicks, as specified by the XmNselectionArray resource.
select-start( ):	Marks the beginning of a new selection region.
self-insert( ):	If XmNpendingDelete is TRUE and the cursor is inside the selection, deletes the entire selection. Inserts the character associated with the key pressed at the insertion cursor.
set-anchor( ):	Resets the anchor point for extended selections. Resets the destination of secondary selection actions.
set-insertion-point( ):	Sets the insertion point.
set-selection-hint( ):	Sets the text source and location of the selection.
toggle-add-mode( ):	Toggles the state of Add Mode.
toggle-editing-path-move( ):	Toggles the editing path direction.
	If the direction is toggled for right to left editing, the insertion cursor stays at the current position.

(continued on next page)

Table C–5 (Cont.) DXmCSText Widget Action Routines

Action Routine	Description
	If the direction is toggled for left to right editing, the insertion cursor will move to the rightmost position of the previous right to left text.
toggle-editing-path-stay( ):	Toggles the editing path direction. The insertion cursor stays at the current position.
toggle-text-path( ):	Toggles the text path direction of the entire text. Inverts each segment's direction, creating a mirror image of the original text.
traverse-home( ):	Traverses to the first widget in the tab group.
traverse-next( ):	Traverses to the next widget in the tab group.
traverse-prev( ):	Traverses to the previous widget in the tab group.
unkill( ):	Restores last killed text to the position of the insertion cursor.

Table C–6 describes additional behavior for the DXmCSText widget.

Table C–6 Additional DXmCSText Behavior

Event	Description
<FocusIn>:	Draws the insertion cursor and starts the cursor blinking.
<FocusOut>:	Stops the cursor blinking.

Index

A

ASCII default values
  as locale-sensitive Motif compound string, 2–170
AST (asynchronous system trap), 1–2
Asynchronous system trap
  See AST

B

Bookreader
  using with DECwindows Motif Help System, 2–173, 2–175, 2–177
Button
  See Push button

C

Callable compiler
  See WML binary databases
Children
  retrieving number belonging to widget, 2–181
ColorMix widget
  See also specific DXmColorMix routines
  creating with dialog box, 2–18 to 2–37
    applying user actions, 2–24
    canceling user actions, 2–24
    confirming user actions, 2–24
    matching color values, 2–26
    specifying callbacks
      XmNapplyCallback, 2–24
      XmNcancelCallback, 2–24
ColorMix widget
  creating with dialog box
    specifying callbacks (cont'd)
      XmNokCallback, 2–24
    specifying color display subwidget, 2–21
    specifying color mixer subwidget, 2–22
    specifying color model, 2–27, 2–28
    specifying color palette, 2–30
    specifying color picker characteristics, 2–30, 2–31, 2–32
    specifying color values, 2–22, 2–23, 2–25
    specifying display window dimensions, 2–25
    specifying interpolator characteristics, 2–29
    specifying labels, 2–21, 2–22, 2–23, 2–24, 2–26, 2–27, 2–28, 2–29, 2–30, 2–31, 2–32, 2–33
    specifying lightness, 2–29
    specifying spectrum tile dimensions, 2–30, 2–31
    specifying visible browser items, 2–26
    specifying warmth, 2–33
    specifying work window, 2–24
    updating color display subwidget values, 2–24
    updating color mixer controls, 2–23
  creating without dialog box, 2–16 to 2–17, 2–37
  retrieving new color values, 2–12 to 2–13
  setting new color values, 2–14 to 2–15

ColorMixiwidget
creating with dialog box
specifying labels, 2–24
Command flags
Motif UIL on UNIX, 3–5 to 3–7
Motif UIL on Windows NT, 3–5 to 3–7
Command qualifiers
OpenVMS, 3–3 to 3–5
to the UIL Compiler, 3–3
Compound string
converting to DDIF string, 2–155
converting to file-compatible string, 2–156
converting to operating-system specific string, 2–157
displaying message, 2–162
locale-sensitive, 2–170
nested, 3–1
specifying direction, 2–7
Compound string text widget
See also specific DXmCSText routines
clearing a global selection, 2–122
copying text, 2–123
creating, 2–38 to 2–52
activating the widget, 2–41
changing value of widget, 2–46
deleting text, 2–43, 2–44
displaying text position, 2–46
editing text, 2–42
establishing input focus, 2–42
inserting text, 2–43
losing input focus, 2–43
moving cursor, 2–44
obtaining editing path direction, 2–40
obtaining text path direction, 2–41
resizing window, 2–44, 2–45
responding to font request, 2–41
responding to mouse click and drag, 2–46
responding to multiple mouse clicks, 2–46
selecting characters, 2–46
showing cursor in text, 2–41
specifying blinking cursor, 2–42
Compound string text widget
creating (cont'd)
specifying callbacks
DXmNnofontCallback, 2–41
XmNactivateCallback, 2–41
XmNfocusCallback, 2–42
XmNlosingFocusCallback, 2–43
XmNmodifyVerifyCallback, 2–43
XmNmotionVerifyCallback, 2–44
XmNvalueChangedCallback, 2–46
specifying cursor blink rate, 2–42
specifying cursor location, 2–42
specifying cursor shape, 2–40
specifying direction, 2–46
specifying fonts, 2–42
specifying keyboard bindings, 2–42
specifying scroll bars, 2–45
specifying text length, 2–43
specifying window dimensions, 2–42, 2–43, 2–45
specifying word wrapping, 2–46
with scroll bars, 2–91 to 2–92
deleting text, 2–124
displaying text, 2–152
enabling redisplay in text, 2–126
highlighting text, 2–149
inserting text, 2–137
moving insertion cursor, 2–144
obtaining ownership status of primary selection, 2–135
pasting clipboard data in text, 2–139
preventing redisplay in text, 2–125
removing selected text, 2–141
replacing text, 2–142
retrieving values
for text, 2–131, 2–133
of first text character position, 2–134
of insertion cursor position, 2–128
of last text character position, 2–129
of maximum text length, 2–130
of number of visible lines, 2–138
of selected text position, 2–132
of text character position, 2–140, 2–154

Compound string text widget
retrieving values (cont'd)
of text editing state, 2–127
scrolling text
horizontally, 2–136
vertically, 2–153
setting values
for Add Mode, 2–144
for displaying new text, 2–150
for editing state, 2–145
for first text character position, 2–151
for insertion cursor position, 2–147
for maximum text length, 2–148
for primary selection, 2–149
for text highlighting, 2–146
Converting strings
compound to DDIF, 2–155
compound to file-compatible, 2–156
compound to operating-system specific, 2–157
DDIF to compound, 2–158
file-compatible to compound, 2–159
OpenVMS string descriptor to null-terminated, 2–161
operating-system-specific to compound, 2–160
Copying text
in compound string text widget, 2–123
Creating a widget
See ColorMix widget
See Compound string text widget
See Help widget
See Print widget
See SVN widget
CSText widget
See Compound string text widget
Cursor
creating, 2–53
moving in compound string text widget, 2–144
retrieving position of, 2–128
retrieving position of in compound string text widget, 2–129
Cursor (cont'd)
setting position in compound string text widget, 2–147

D

DDIF string
converting compound string to, 2–155
converting to compound string, 2–158
DECterm window
creating, B–2
DECwindows Motif Help System
displaying topic or directory, 2–175
initialization operations, 2–177
DECwindows Motif Toolkit
initializing, 2–179
DECwTermPort routine, B–2
Deleting text
in compound string text widget, 2–124
Dialog shell widget
manipulating dialog shells, 2–1
DXmActivateWidget routine, 2–10
DXmChildren routine, 2–11
DXmColorMix widget
See also ColorMix widget
UIL table, A–2
DXmColorMixDialog widget
See also ColorMix widget
UIL table, A–6
DXmColorMixGetNewColor routine, 2–12
DXmColorMixSetNewColor routine, 2–14
DXmCreateColorMix routine, 2–16
DXmCreateColorMixDialog routine, 2–18
DXmCreateCSText routine, 2–38
DXmCreateCursor routine, 2–53
DXmCreateHelpDialog routine, 2–55
DXmCreatePrintBox routine, 2–71
DXmCreatePrintDialog routine, 2–73
DXmCreateScrolledCSText routine, 2–91
DXmCreateScrolledCSText widget
See Compound string text widget
DXmCreateSvn routine, 2–93

DXmCSContainsStringCharSet routine, 2–121
DXmCSText widget
See also Compound string text widget
UIL table, A–10
DXmCSTextClearSelection routine, 2–122
DXmCSTextCopy routine, 2–123
DXmCSTextCut routine, 2–124
DXmCSTextDisableRedisplay routine, 2–125
DXmCSTextEnableRedisplay routine, 2–126
DXmCSTextGetEditable routine, 2–127
DXmCSTextGetInsertionPosition routine, 2–128
DXmCSTextGetLastPosition routine, 2–129
DXmCSTextGetMaxLength routine, 2–130
DXmCSTextGetSelection routine, 2–131
DXmCSTextGetSelectionInfo routine, 2–132
DXmCSTextGetString routine, 2–133
DXmCSTextGetTopPosition routine, 2–134
DXmCSTextHasSelection routine, 2–135
DXmCSTextHorizontalScroll routine, 2–136
DXmCSTextInsert routine, 2–137
DXmCSTextNumLines routine, 2–138
DXmCSTextPaste routine, 2–139
DXmCSTextPosToXY routine, 2–140
DXmCSTextRemove routine, 2–141
DXmCSTextReplace routine, 2–142
DXmCSTextSetAddMode routine, 2–144
DXmCSTextSetEditable routine, 2–145
DXmCSTextSetHighlight routine, 2–146
DXmCSTextSetInsertionPosition routine, 2–147
DXmCSTextSetMaxLength routine, 2–148
DXmCSTextSetSelection routine, 2–149
DXmCSTextSetString routine, 2–150
DXmCSTextSetTopPosition routine, 2–151
DXmCSTextShowPosition routine, 2–152
DXmCSTextVerticalScroll routine, 2–153
DXmCSTextXYToPos routine, 2–154
DXmCvtCStoDDIF routine, 2–155
DXmCvtCStoFC routine, 2–156
DXmCvtCStoOS routine, 2–157
DXmCvtDDIFtoCS routine, 2–158
DXmCvtFCtoCS routine, 2–159
DXmCvtOStoCS routine, 2–160
DXmDescToNull routine, 2–161
DXmDisplayCSMessage routine, 2–162
DXmDisplayVMSMessage routine, 2–165
DXmFindFontFallback routine, 2–168
DXmFormSpaceButtonsEqually routine, 2–169
DXmGetComponentWidth routine, 2–202
DXmGetLocaleString routine, 2–170
DXmHelpDialog widget
See also Help widget
UIL table, A–13
DXmHelpOnContext routine, 2–172
DXmHelpSystemClose routine, 2–173
DXmHelpSystemDisplay routine, 2–175
DXmHelpSystemOpen routine, 2–177
DXmInitialize routine, 2–179
DXmLoadQueryFont routine, 2–180
DXmNautoUnrealize resource, 2–1
DXmNfitToScreenPolicy resource, 2–1
DXmNumChildren routine, 2–181
DXmPrint widget
See Print widget
DXmPrintBox widget
See also Print widget
UIL table, A–17
DXmPrintDialog widget
See also Print widget
UIL table, A–21
DXmPrintWgtAugmentList routine, 2–182
DXmPrintWgtPrintJob routine, 2–185
DXmScrolledCSText widget
See also Compound string text widget
UIL table, A–25
DXmSetComponentWidth routine, 2–244
DXmSvn widget
See also SVN widget
UIL table, A–27
DXmSvnAddEntries routine, 2–186
DXmSvnAutoScrollCheck routine, 2–188
DXmSvnAutoScrollDisplay routine, 2–190

DXmSvnClearHighlight routine, 2–191
DXmSvnClearHighlighting routine, 2–192
DXmSvnClearSelection routine, 2–193
DXmSvnClearSelections routine, 2–194
DXmSvnDeleteEntries routine, 2–195
DXmSvnDisableDisplay routine, 2–196
DXmSvnEnableDisplay routine, 2–197
DXmSvnFlushEntry routine, 2–198
DXmSvnGetComponentNumber routine, 2–199
DXmSvnGetComponentTag routine, 2–200
DXmSvnGetComponentText routine, 2–201
DXmSvnGetDisplayed routine, 2–203
DXmSvnGetEntryLevel routine, 2–205
DXmSvnGetEntryNumber routine, 2–206
DXmSvnGetEntryPosition routine, 2–207
DXmSvnGetEntrySensitivity routine, 2–209
DXmSvnGetEntryTag routine, 2–210
DXmSvnGetHighlighted routine, 2–211
DXmSvnGetNumDisplayed routine, 2–213
DXmSvnGetNumHighlighted routine, 2–214
DXmSvnGetNumSelections routine, 2–215
DXmSvnGetPrimaryWorkWidget routine, 2–216
DXmSvnGetSecondaryWorkWidget routine, 2–217
DXmSvnGetSelections routine, 2–218
DXmSvnGetTreePosition routine, 2–220
DXmSvnHideHighlighting routine, 2–221
DXmSvnHideSelections routine, 2–222
DXmSvnHighlightAll routine, 2–223
DXmSvnHighlightEntry routine, 2–224
DXmSvnInsertComponent routine, 2–225
DXmSvnInvalidateEntry routine, 2–226
DXmSvnMapPosition routine, 2–227
DXmSvnPositionDisplay routine, 2–228
DXmSvnRemoveComponent routine, 2–230
DXmSvnSelectAll routine, 2–231
DXmSvnSelectComponent routine, 2–232
DXmSvnSelectEntry routine, 2–233
DXmSvnSetApplDragging routine, 2–234
DXmSvnSetComponentHidden routine, 2–235
DXmSvnSetComponentPixmap routine, 2–237
DXmSvnSetComponentTag routine, 2–239
DXmSvnSetComponentText routine, 2–240
DXmSvnSetComponentWidget routine, 2–242
DXmSvnSetEntry routine, 2–245
DXmSvnSetEntryIndexWindow routine, 2–247
DXmSvnSetEntryNumComponents routine, 2–248
DXmSvnSetEntryPosition routine, 2–249
DXmSvnSetEntrySensitivity routine, 2–251
DXmSvnSetEntryTag routine, 2–252
DXmSvnSetTreePosition routine, 2–253
DXmSvnShowHighlighting routine, 2–254
DXmSvnShowSelections routine, 2–255
DXmSvnValidateAll routine, 2–256

E

Enabling redisplay of text
in compound string text widget, 2–126
Events
selecting asynchronous input types, 1–4
selecting asynchronous types, 1–2
Extensions
DECwTermPort routine, B–1
Digital supplied resources, 2–1 to 2–7
Digital supplied routines
summary, 2–8
nested compound strings, 3–1
to the UIL compiler command, 3–3
to UIL, 3–1 to 3–10
to Xlib for OpenVMS, 1–1 to 1–6
to XmStringDirectionCreate routine, 2–7
WML binary databases, 3–7

F

Font names
associating with new XLFD font name, 2–168
generating new, 2–180

Fonts
loading, 2–180

G

Gadgets
UIL list, A–138

H

Help
context-sensitive, 2–172
DECwindows Motif Help System
closing Bookreader windows, 2–173
displaying topic or directory, 2–175
initialization operations, 2–177
Help widget
creating, 2–55 to 2–70
responding to widget mapping call, 2–69
responding to widget unmapping call, 2–69
specifying callbacks
XmNmapCallback, 2–69
XmNunmapCallback, 2–69
specifying character set for defined keys, 2–59, 2–60, 2–61, 2–62, 2–63, 2–64, 2–65, 2–66, 2–67, 2–68, 2–69
specifying error messages, 2–58, 2–60, 2–65
specifying first topic, 2–61
specifying glossary topic, 2–61
specifying help topic library file, 2–64, 2–65
specifying keys, 2–59, 2–60, 2–61, 2–62, 2–63, 2–64, 2–65, 2–66, 2–67, 2–68
specifying labels, 2–58, 2–59, 2–60, 2–61, 2–62, 2–63, 2–64, 2–65, 2–66, 2–67, 2–68
specifying name in title bar, 2–58
specifying overview topic, 2–65
specifying title bar in Help-on-Help subwidget, 2–63
Help widget
creating (cont'd)
specifying widget position, 2–59
specifying window dimensions, 2–59, 2–65
storing help text in memory, 2–58
Highlighting
in compound string text widget
for primary selection, 2–149
setting, 2–146
in SVN widget
displaying highlighted entries, 2–221
highlighting entries, 2–223, 2–224, 2–254
removing, 2–192
removing highlighting, 2–191
retrieving highlighted entries, 2–211, 2–214

I

Initializing DECwindows Motif Toolkit, 2–179
ISO_LATIN_1 characters
detecting in XmString segments, 2–121

K

Keyboard bindings
SVN widget, 2–114

M

Messages
displaying compound string, 2–162
displaying OpenVMS, 2–165
specifying in Help widget, 2–58, 2–60, 2–65
specifying in Print widget, 2–80
Motif UIL UNIX command flags, 3–5 to 3–7
Motif UIL Windows NT command flags, 3–5 to 3–7
MRM
using for Digital supplied widgets, 2–179

N

Nested compound strings, 3–1
  specifying direction, 2–7

O

OpenVMS operating system
  compiler command qualifiers, 3–3 to 3–5
  displaying a message, 2–165
  files provided for UIL support, 3–7, 3–8

P

Performance
  and DXmNautoUnrealize resource, 2–1
Print formats
  defining, 2–183
Print widget
  adding options, 2–182
  creating
    deleting file, 2–76
    holding print job in queue, 2–78
    notifying user of job completion, 2–79
    passing control characters, 2–82
    printing header, 2–78
    printing image, 2–79
    specifying autopagination, 2–75
    specifying callbacks
      XmNcancelCallback, 2–89
      XmNokCallback, 2–89
    specifying default printer, 2–76
    specifying dialog box title, 2–80
    specifying double-spacing, 2–76
    specifying ending page, 2–81
    specifying error message processing, 2–79
    specifying File Burst Sheet, 2–76
    specifying File End Sheet, 2–77
    specifying File Start Sheet, 2–77
    specifying image size, 2–81
    specifying input tray, 2–78
    specifying number of copies, 2–79
    specifying number of files, 2–77
    specifying number of images, 2–79
Print widget
  creating (cont'd)
    specifying number of printers, 2–84
    specifying number of sheets, 2–86
    specifying number of sides to be printed, 2–86
    specifying operator messages, 2–80
    specifying output tray, 2–80
    specifying paper orientation, 2–80
    specifying print format, 2–77, 2–84, 2–85
    specifying print job name, 2–78
    specifying print job priority, 2–85
    specifying print queue, 2–84
    specifying printer, 2–83
    specifying printer form, 2–84
    specifying setup modules, 2–85
    specifying size of printing medium, 2–86
    specifying starting page, 2–81
    specifying text for File Start Sheet, 2–87
    specifying time and date, 2–83
    suppressing print widget features, 2–87
    unmanaging widget, 2–89
  creating with dialog box, 2–73 to 2–90
  creating without dialog box, 2–71 to 2–72
  defining additional print formats, 2–182
  submitting a print job, 2–185
Push button
  positioning and sizing, 2–169
  simulating push-button activation, 2–10

Q

Qualifiers
  to the UIL compiler, 3–3

R

Redisplay of text
  in compound string text widget, 2–125
Routines
  See DECwTermPort routine
  See Toolkit routines

Routines (cont'd)
See Xlib routines

S

Screens
small-screen window implementation, 2–1
Scrolled compound string text widget
See Compound string text widget
Scrolled window
with small-screen implementations, 2–1
Strings
converting compound to DDIF, 2–155
converting compound to file-compatible, 2–156
converting compound to operating-system specific, 2–157
converting DDIF to compound, 2–158
converting file-compatible to compound, 2–159
converting OpenVMS string descriptor to null-terminated, 2–161
converting operating-system-specific to compound, 2–160
Structured Visual Navigation widget
See SVN widget
SVN (Structured Visual Navigation) widget
associating entry position with entry number, 2–227
automatic scrolling, 2–190
calling DXmSvnAutoScrollDisplay, 2–188
canceling selection, 2–193, 2–194
clearing selection, 2–193, 2–194
column format, 2–115
creating, 2–93 to 2–120
attaching to source data module, 2–95
canceling a selection, 2–99
changing entry display, 2–96
copying an entry, 2–99
detaching from source data module, 2–96
dragging
SVN (Structured Visual Navigation) widget
creating
dragging (cont'd)
application-controlled, 2–97, 2–108
SVN-controlled, 2–107
generating callbacks sequentially, 2–101
highlighting an entry, 2–100
indenting entry level, 2–104
moving an entry, 2–99
responding to help request, 2–103
responding to MB2 click with no mouse movement, 2–99
retrieving entry information, 2–101
selecting a single entry, 2–106
selecting an entry, 2–98
selecting multiple entries, 2–100
separating columns, 2–95
spacing components, 2–96
specifying a pop-up menu, 2–105
specifying callbacks
DXmSvnNattachToSourceCallback, 2–95
DXmSvnNdetachFromSourceCallback, 2–96
DXmSvnNdisplayChangedCallback, 2–96
DXmSvnNdraggingCallback, 2–97
DXmSvnNdraggingEndCallback, 2–97
DXmSvnNentrySelectedCallback, 2–98
DXmSvnNentryTransferCallback, 2–99
DXmSvnNentryUnselectedCallback, 2–99
DXmSvnNextendConfirmCallback, 2–100
DXmSvnNgetEntryCallback, 2–101
DXmSvnNhelpRequestedCallback, 2–103
DXmSvnNpopupMenuCallback, 2–105

SVN (Structured Visual Navigation) widget
creating
specifying callbacks (cont'd)
DXmSvnNselectAndConfirmCallback, 2–106
DXmSvnNselectionsDraggedCallback, 2–107
DXmSvnNtransitionsDoneCallback, 2–108
specifying cursor location, 2–108
specifying display modes, 2–96
specifying entry width, 2–101
specifying fonts, 2–101
specifying highlighted dragging image (ghost), 2–102
specifying multiple selections, 2–104
specifying navigation window title, 2–104
specifying number of entries, 2–104
specifying pane widget, 2–104
specifying path to root, 2–108
specifying primary window widget, 2–104
specifying scroll bar, 2–104
specifying scroll bar arrows, 2–110
specifying scrolling, 2–104
specifying secondary column, 2–108
specifying secondary components, 2–105
specifying secondary window position, 2–105
specifying secondary window widget, 2–105
specifying selection mode, 2–107
specifying tree mode characteristics, 2–109, 2–110
truncating text, 2–110
deleting entries, 2–195
detecting automatic scrolling areas, 2–188
disabling, 2–196
display formats, 2–115
displaying an entry, 2–198
displaying entries, 2–226, 2–228
displaying highlighted entries, 2–221
SVN (Structured Visual Navigation) widget (cont'd)
displaying selected entries, 2–222
dragging
application-controlled, 2–188, 2–234
SVN-controlled, 2–234
drawing a display, 2–203
flushing an entry, 2–198
hiding components, 2–235
highlighting entries, 2–223, 2–224
inserting components, 2–225
keyboard bindings, 2–114
modes, 2–115
opening entries, 2–186
outline format, 2–115
re-enabling, 2–197
removing components, 2–230
removing highlighting, 2–191, 2–192
retrieving values
of component number, 2–199
of component tag, 2–200
of component width, 2–202
of compound string address, 2–201
of display window position, 2–220
of entry level, 2–205
of entry numbers, 2–203, 2–206
of entry position, 2–207
of entry sensitivity, 2–209
of entry tag, 2–210
of highlighted entries, 2–211
of number of displayed entries, 2–213
of number of highlighted entries, 2–214
of number of selected entries, 2–215
of primary work widget ID, 2–216
of secondary work widget ID, 2–217
of selected entry numbers, 2–218
selecting entries, 2–231, 2–232, 2–233
setting values
for application dragging mode, 2–234
for component display modes, 2–235
for component tag, 2–239
for component width, 2–244
for entry level information, 2–245

SVN (Structured Visual Navigation) widget
setting values (cont'd)
for entry position in user-defined tree mode, 2–249
for entry sensitivity, 2–251
for number of components for entry, 2–248
for tree position, 2–253
to add component string to entry, 2–240
to add pixmap component, 2–237
to add subwidget to entry, 2–242
to assign entry tag to entry, 2–252
to display entry in index window, 2–247
specifying highlighting graphics, 2–254
specifying selection graphics, 2–255
tree format, 2–115
updating entries, 2–226
updating the display, 2–197
validating entries, 2–256

T

Text
See also Compound string text widget
changing highlighting, 2–146
clearing a selection in compound string text widget, 2–122
copying in compound string text widget, 2–123
deleting in compound string text widget, 2–124
determining editing state, 2–127
displaying in compound string text widget, 2–150, 2–152
enabling redisplay in compound string text widget, 2–126
highlighting as primary selection in compound string text widget, 2–149
inserting text into compound string text widget, 2–137
moving insertion cursor, 2–144
obtaining ownership status of primary selection, 2–135
Text (cont'd)
pasting clipboard data in compound string text widget, 2–139
preventing redisplay in compound string text widget, 2–125
removing in compound string text widget, 2–141
replacing in compound string text widget, 2–142
retrieving character position, 2–140, 2–154
retrieving first character position, 2–134
retrieving in compound string text widget, 2–131, 2–133
retrieving insertion cursor position, 2–128
retrieving last character position, 2–129
retrieving maximum length of, 2–130
retrieving number of visible lines, 2–138
retrieving position of, 2–132
scrolling
horizontally, 2–136
vertically, 2–153
setting Add Mode, 2–144
setting editing state, 2–145
setting insertion cursor position, 2–147
setting maximum length, 2–148
setting position of first character, 2–151
Toolkit routines
DXmActivateWidget, 2–10
DXmChildren, 2–11
DXmColorMixGetNewColor, 2–12
DXmColorMixSetNewColor, 2–14
DXmCreateColorMix, 2–16
DXmCreateColorMixDialog, 2–18
DXmCreateCSText, 2–38
DXmCreateCursor, 2–53
DXmCreateHelpDialog, 2–55
DXmCreatePrintBox, 2–71
DXmCreatePrintDialog, 2–73
DXmCreateScrolledCSText, 2–91
DXmCreateSvn, 2–93
DXmCSContainsStringCharSet, 2–121
DXmCSTextClearSelection, 2–122
DXmCSTextCopy, 2–123

Toolkit routines (cont'd)
DXmCSTextCut, 2–124
DXmCSTextDisableRedisplay, 2–125
DXmCSTextEnableRedisplay, 2–126
DXmCSTextGetEditable, 2–127
DXmCSTextGetInsertionPosition, 2–128
DXmCSTextGetLastPosition, 2–129
DXmCSTextGetMaxLength, 2–130
DXmCSTextGetSelection, 2–131
DXmCSTextGetSelectionInfo, 2–132
DXmCSTextGetString, 2–133
DXmCSTextGetTopPosition, 2–134
DXmCSTextHasSelection, 2–135
DXmCSTextHorizontalScroll, 2–136
DXmCSTextInsert, 2–137
DXmCSTextNumLines, 2–138
DXmCSTextPaste, 2–139
DXmCSTextPosToXY, 2–140
DXmCSTextRemove, 2–141
DXmCSTextReplace, 2–142
DXmCSTextSetAddMode, 2–144
DXmCSTextSetEditable, 2–145
DXmCSTextSetHighlight, 2–146
DXmCSTextSetInsertionPosition, 2–147
DXmCSTextSetMaxLength, 2–148
DXmCSTextSetSelection, 2–149
DXmCSTextSetString, 2–150
DXmCSTextSetTopPosition, 2–151
DXmCSTextShowPosition, 2–152
DXmCSTextVerticalScroll, 2–153
DXmCSTextXYToPos, 2–154
DXmCvtCStoDDIF, 2–155
DXmCvtCStoFC, 2–156
DXmCvtCStoOS, 2–157
DXmCvtDDIFtoCS, 2–158
DXmCvtFCtoCS, 2–159
DXmCvtOStoCS, 2–160
DXmDescToNull, 2–161
DXmDisplayCSMessage, 2–162
DXmDisplayVMSMessage, 2–165
DXmFindFontFallback, 2–168
DXmFormSpaceButtonsEqually, 2–169
DXmGetComponentWidth, 2–202
DXmGetLocaleString, 2–170
DXmHelpOnContext, 2–172

Toolkit routines (cont'd)
DXmHelpSystemDisplay, 2–175
DXmHelpSystemOpen, 2–177
DXmInitialize, 2–179
DXmLoadQueryFont, 2–180
DXmNumChildren, 2–181
DXmPrintWgtAugmentList, 2–182
DXmPrintWgtPrintJob, 2–185
DXmSetComponentWidth, 2–244
DXmSvnAddEntries, 2–186
DXmSvnAutoScrollCheck, 2–188
DXmSvnAutoScrollDisplay, 2–190
DXmSvnClearHighlight, 2–191
DXmSvnClearHighlighting, 2–192
DXmSvnClearSelection, 2–193
DXmSvnClearSelections, 2–194
DXmSvnDeleteEntries, 2–195
DXmSvnDisableDisplay, 2–196
DXmSvnEnableDisplay, 2–197
DXmSvnFlushEntry, 2–198
DXmSvnGetComponentNumber, 2–199
DXmSvnGetComponentTag, 2–200
DXmSvnGetComponentText, 2–201
DXmSvnGetDisplayed, 2–203
DXmSvnGetEntryLevel, 2–205
DXmSvnGetEntryNumber, 2–206
DXmSvnGetEntryPosition, 2–207
DXmSvnGetEntrySensitivity, 2–209
DXmSvnGetEntryTag, 2–210
DXmSvnGetHighlighted, 2–211
DXmSvnGetNumDisplayed, 2–213
DXmSvnGetNumHighlighted, 2–214
DXmSvnGetNumSelections, 2–215
DXmSvnGetPrimaryWorkWidget, 2–216
DXmSvnGetSecondaryWorkWidget, 2–217
DXmSvnGetSelections, 2–218
DXmSvnGetTreePosition, 2–220
DXmSvnHideHighlighting, 2–221
DXmSvnHideSelections, 2–222
DXmSvnHighlightAll, 2–223
DXmSvnHighlightEntry, 2–224
DXmSvnInsertComponent, 2–225
DXmSvnInvalidateEntry, 2–226
DXmSvnMapPosition, 2–227

Toolkit routines (cont'd)
DXmSvnPositionDisplay, 2–228
DXmSvnRemoveComponent, 2–230
DXmSvnSelectAll, 2–231
DXmSvnSelectComponent, 2–232
DXmSvnSelectEntry, 2–233
DXmSvnSetApplDragging, 2–234
DXmSvnSetComponentHidden, 2–235
DXmSvnSetComponentPixmap, 2–237
DXmSvnSetComponentTag, 2–239
DXmSvnSetComponentText, 2–240
DXmSvnSetComponentWidget, 2–242
DXmSvnSetEntry, 2–245
DXmSvnSetEntryIndexWindow, 2–247
DXmSvnSetEntryNumComponents, 2–248
DXmSvnSetEntryPosition, 2–249
DXmSvnSetEntrySensitivity, 2–251
DXmSvnSetEntryTag, 2–252
DXmSvnSetTreePosition, 2–253
DXmSvnShowHighlighting, 2–254
DXmSvnShowSelections, 2–255
DXmSvnValidateAll, 2–256
XmStringDirectionCreate
specifying revert value, 2–7
Translation table, C–1

U

UIL extensions, 3–1 to 3–10
UIL tables, A–1 to A–142
UNIX operating system
compiler command flags
Motif UIL, 3–5 to 3–7
files provided for UIL support, 3–9, 3–10
invoking Motif UIL compiler, 3–5
user_defined
UIL table, A–138

W

Widget Meta-Language binary databases
See WML binary databases
Widgets
retrieving number of children, 2–181
returning list of children, 2–11
UIL list, A–138, A–141
Windows
creating and destroying with DXmNautoUnrealize, 2–1
creating DECterm window, B–2
sizing with DXmNfitToScreenPolicy, 2–1
Windows NT operating system
compiler command flags
Motif UIL, 3–5 to 3–7
files provided for UIL support, 3–9, 3–10
invoking Motif UIL compiler, 3–5
WML binary databases, 3–1, 3–7

X

X Logical Font Description
See XLFD
XLFD
and DXmFindFontFallback routine, 2–168
Xlib routines
XSelectAsyncEvent, 1–2
XSelectAsyncInput, 1–4
XmArrowButton
UIL table, A–30
XmArrowButtonGadget
UIL table, A–32
XmBulletinBoard
UIL table, A–33
XmBulletinBoardDialog
UIL table, A–35
XmCascadeButton
UIL table, A–37
XmCascadeButtonGadget
UIL table, A–39
XmCommand
UIL table, A–40
XmDrawingArea
UIL table, A–43
XmDrawnButton
UIL table, A–45

XmErrorDialog
UIL table, A–47
XmFileSelectionBox
UIL table, A–49
XmFileSelectionDialog
UIL table, A–53
XmForm
UIL table, A–56
XmFormDialog
UIL table, A–58
XmFrame
UIL table, A–61
XmInformationDialog
UIL table, A–63
XmLabel
UIL table, A–65
XmLabelGadget
UIL table, A–67
XmList
UIL table, A–69
XmMainWindow
UIL table, A–71
XmMenuBar
UIL table, A–73
XmMessageBox
UIL table, A–75
XmMessageDialog
UIL table, A–78
XmOptionMenu
UIL table, A–80
XmPanedWindow
UIL table, A–82
XmPopupMenu
UIL table, A–84
XmPromptDialog
UIL table, A–87
XmPulldownMenu
UIL table, A–89
XmPushButton
UIL table, A–92
XmPushButtonGadget
UIL table, A–95
XmQuestionDialog
UIL table, A–96
XmRadioBox
UIL table, A–99
XmRowColumn
UIL table, A–101
XmScale
UIL table, A–104
XmScrollBar
UIL table, A–106
XmScrolledList
UIL table, A–108
XmScrolledText
UIL table, A–110
XmScrolledWindow
UIL table, A–113
XmSelectionBox
UIL table, A–114
XmSelectionDialog
UIL table, A–117
XmSeparator
UIL table, A–120
XmSeparatorGadget
UIL table, A–121
XmStringDirectionCreate routine
specifying revert value, 2–7
XmText
UIL table, A–122
XmTextField
UIL table, A–124
XmToggleButton
UIL table, A–127
XmToggleButtonGadget
UIL table, A–129
XmWarningDialog
UIL table, A–131
XmWorkArea
UIL table, A–133
XmWorkingDialog
UIL table, A–136
XSelectAsyncEvent routine, 1–2
XSelectAsyncInput routine, 1–4