Preface

The *Motif Programmer's Reference* contains the reference pages for Motif® commands and functions, including toolkit, window manager, and user interface language commands and functions.

Audience

This document is written for programmers who want to write applications by using Motif® interfaces. This document assumes that the reader is familiar with the American National Standards Institute (ANSI) C programming language. It also assumes that the reader has a general understanding of the X Window System, the Xlib library, and the X Toolkit Intrinsics (Xt).

Applicability

This is Revision 2.0 of this document. It applies to Release 2.0 of the Motif software system.

Purpose

The purpose of this reference is to provide detailed descriptions of the Motif commands and functions.

Document Usage

This document is organized into one chapter and three appendixes:

- Chapter 1 contains all the reference pages for the Motif commands and functions.
- Appendix A contains a list of the constraint arguments and automatically created children for widgets available within UIL (User Interface Language).
- Appendix B contains a list of the reasons and controls, or children, that UIL supports for each Motif Toolkit object.
- Appendix C contains a list of the UIL arguments and their data types.
- Appendix D contains a list of the UIL compiler diagnostics messages.

Reference Page Format

The reference pages in this volume use the following format:

Purpose

This section gives a short description of the interface.

Synopsis

This section describes the appropriate syntax for using the interface.

Description

This section describes the behavior of the interface. On widget reference pages there are tables of resource values in the descriptions. These tables have the following headings:

Name

Contains the name of the resource. Each new resource is described following the new resources table.

Class

Contains the class of the resource.

Type

Contains the type of the resource.

Default

Contains the default value of the resource.

Access

Contains the access permissions for the resource. A *C* in this column means the resource can be set at widget creation time. An *S* means the resource can be set anytime. A *G* means the resource's value can be retrieved.

Examples

This sections gives practical examples for using the interface.

Return Value

This section lists the values returned by function interfaces.

Errors

This section describes the error conditions associated with using this interface.

Related Information

This section provides cross−references to related interfaces and header files described within this document.

Related Documents

For additional information about Motif, refer to the following documents:

- The *Application Environment Specification −− User Environment Volume* defines a stable set of routines for creating user interface applications.
- The *Motif Style Guide* explains the principles of user interface design for application developers.
- The *Motif User's Guide* explains how to interact with Motif applications.
- The *Motif Widget Writer's Guide* explains how to create Motif widgets.

For additional information about Xlib and Xt, refer to the following X Window System documents:

- *Xlib−−C Language X Interface* is the specification for Xlib.
- *X Toolkit Intrinsics−−C Language Interface* is the specification for Xt.

Typographic and Keying Conventions

This document uses the following typographic conventions:

Bold

> *Bold* words or characters represent system elements that an application or user must use literally, such as functions, data types, commands, flags, and pathnames. *Bold* words also indicate the first use of a term included in the glossary.

Italic

> *Italic* words or characters represent variable values and arguments that an application or user must supply. The *Italic* font is also used for emphasis.

`Constant width`

> Examples and information that the system displays appear in this typeface.

`< >`

> Angle brackets enclose the name of a key on the keyboard.

ComponentName

> Components of the user interface are represented by uppercase letters for each major word in the name of the component, such as PushButton.

Keyboard Conventions

Because not all keyboards are the same, it is difficult to specify keys that are correct for every manufacturer's keyboard. To solve this problem, this reference describes keys by using a *virtual key* mechanism. The term *virtual* implies that the keys as described do not necessarily correspond to a fixed set of actual keys. Instead, virtual keys are linked to actual keys by means of *virtual bindings*. A given virtual key may be bound to different physical keys for different keyboards.

See the *Motif Programmer's Guide* for information on the mechanism for binding virtual keys to actual keys. For details, see the **VirtualBindings**(3) reference page in this document.

Mouse Conventions

Mouse buttons are described in this reference by using a *virtual button* mechanism to better describe behavior independent from the number of buttons on the mouse. This guide assumes a 3–button mouse. On a 3–button mouse, the leftmost mouse button is usually defined as *BSelect*, the middle mouse button is usually defined as *BTransfer*, and the rightmost mouse button is usually defined as *BMenu*. For details about how virtual mouse buttons are usually defined, see the **VirtualBindings**(3) reference page in this document.

Problem Reporting

If you have any problems with the software or documentation, please contact your software vendor's customer service department.