

## Preface

### Preface to the Open Source Edition

Many thanks to all at O'Reilly and Associates for releasing this, Volume 6B, and the companion Volume 6A, the Motif Programming Manual, in open source. Both have been extensively revised for Motif 2.1; this, the Motif Reference Manual, has had several alterations to the 2nd edition as printed:

- all the function prototypes and examples have been converted to strict ANSI format
- the UIL sections have been restored
- the Xt Session Shell is documented
- many bug patches have been folded in
- new examples have been added to Motif 2.1 procedure sections
- the book sources have been converted from the original troff into FrameMaker and PDF formats

Removing the UIL portions from the original printed second edition was a hard decision; the Motif 2.1 toolkit was a much expanded library since previous versions of the book, and something had to give - the book was over a thousand pages as it was. However, an electronic copy does not have the same space restrictions as the printed tome, and so these materials, originally in the Motif 1.2 version of the manual, have been restored. They also have been reworked for Motif 2.1.

*Antony J. Fountain*

### Preface to the Second Edition

What to put in, and what to leave out, of this update to the Motif Reference Manual was the hardest decision of all. The guiding principle has been to consider for whom this material is intended. This is a Programmer's Reference, and not a Widget Author's handbook. Accordingly, those aspects of the new Trait mechanisms which an application programmer needs to know have been included, but the Xme utilities have not. Specifying a Trait as a well-defined piece of behaviour which a widget supports, it is enough to know which traits a Widget Class supports, and how this affects objects in the widget instance hierarchy. How a Trait is implemented, and which methods are associated with the given Trait, are generally the domain of the widget author. Hence it is recorded that the VendorShell holds the XmQTspecifyRenderTableTrait, and that this means that widget classes further down the widget instance hierarchy inherit default Render Table information from the VendorShell. This is all that the Application Programmer needs to know: the rest is silence.

Conversely, the Motif Input Method utilities have been included. Although mostly defined originally in the Motif 1.2 release, and although the Motif widget classes generally handle connections to an Input Method when and where this is required, there is an important exception. The Motif Drawing Area does not register itself with an Input Method automatically, and hence anyone who needs to directly implement internationalized input for this widget class most certainly would need to know about the XmIm functions. The World does not all speak English: for these reasons, the XmIm functions are included in the Manual.

A brief note concerning the status of Motif as the premier Unix toolkit. A number of alternative toolkits have arisen, particularly in the Linux domain, which offer an X-based windowing system for the Unix, and other, platforms. I refer principally to the likes of Qt, and GTK+. These on the whole dispense with the Xt layer, in order to provide small, lightweight GUI components which are, from the application programmer's perspective, relatively easy to port to non-Unix domains. Although admirable in many ways, these suffer from one crucial drawback, precisely because Xt has been excluded: there is no object component model associated with any of the objects which can be created in an interface<sup>1</sup>. Compare and contrast with something like JavaBeans, where a GUI builder can be designed which can dynamically load and query objects from whatever source, and from thence inspect the attributes of the object, construct resource panels, and generate code for the components, all without any external configuration. Based on Xt, Motif also has this important property: I can in principle dynamically load into my GUI builder any third party component, construct an internal attribute list, present resource panels for object configuration to the user, and from there generate source code. Just by interrogating the widget class. All the commercial GUI builders available for Motif support this.

The newer alternative Linux toolkits do not have this introspective quality. Writing GUI builders happens to be what I do for a living: sad to say, I cannot write one for these toolkits precisely because there is no component model at the object level. Not surprisingly, no third party component market exists for the toolkits either: there is no GUI builder into which these components can be dynamically slotted. Each needs the other, but there is nothing which allows them to talk. In the absence of either a commercial component market, or a dynamic GUI builder, there remains serious question marks concerning the scalability of the alternative toolkits, whatever merits they hold. The only alternatives are to write all the code by hand, or pass control of the

---

1. True at the moment of writing. It is still true that all the information required to dynamically introspect an object's entire resource set, particularly if user-defined and not built-in to the basic set, is not completely forthcoming. Introspecting third party components remains troublesome for a dynamic GUI builder.

## Preface

application to a private piece of hobbyware which masquerades as a support environment. Ironically, the advent of Java has cemented Motif: the JDK relies on Motif for the native implementation on the Unix platform. Until such time as a native toolkit surfaces which has this important introspective property, Motif remains what it has long been, the only native toolkit for Unix which supports large scale internationalized applications.<sup>1</sup>

### About the Motif Toolkit

The Motif toolkit, from the Open Software Foundation (OSF), is based on the X Toolkit Intrinsic (Xt), which is the standard mechanism on which many of the toolkits written for the X Window System are based. Xt provides a library of user-interface objects called widgets and gadgets, which provide a convenient interface for creating and manipulating X windows, colormap, events, and other cosmetic attributes of the display. In short, widgets can be thought of as building blocks that the programmer uses to construct a complete application.

However, the widgets that Xt provides are generic in nature and impose no user-interface policy whatsoever. Providing the look and feel of an interface is the job of a user-interface toolkit such as Motif. Motif provides a complete set of widgets that are designed to implement the application look and feel specified in the *Motif Style Guide* and the *Motif Application Environment Specification*. The Motif toolkit also includes a library of functions for creating and manipulating the widgets and other aspects of the user interface.

The Motif toolkit has other components in addition to the widget set and related functions. Motif provides a User Interface Language (UIL) for describing the initial state of a user interface. UIL is designed to permit rapid prototyping of the user interface for an application. The Motif Resource Manager (Mrm) functions provide the interface between C language application code and UIL. Motif also provides the Motif Window Manager (*mwm*). The appearance and behavior of this window manager is designed to be compatible with the appearance and behavior of the Motif widget set.

### About This Manual

This manual contains reference material on the Motif toolkit. This edition is based on Motif 2.1, which is the latest major release of the Motif toolkit. Motif 1.2 is based on

---

<sup>1</sup>The contents of this paragraph were true at the moment of writing. There is now a commercial GUI builder for the Linux toolkits; whether it survives in a free software environment remains to be seen. It is still true that the large scale commercial concerns continue to use Motif for their native Unix toolkit.

## Preface

Release 6 of the Xlib and Xt specifications (X11R6). This release of Motif provides many new features, including new widget classes and several new functions. In order to cover all of the material, it became necessary to split Volume Six into two separate manuals, a programming manual and a reference manual. Volume Six A is the *Motif Programming Manual* and Volume Six B is the *Motif Reference Manual*.

This manual is part of the sixth volume in the O'Reilly & Associates X Window System Series. It includes reference pages for each of the Motif functions and macros, for the Motif and Xt Intrinsic widget classes, for the Mrm functions, for the Motif clients, and for the UIL file format, data types, and functions. A permuted index and numerous quick reference appendices are also provided.

Volume Six B includes reference pages for all of the new functions and widgets in Motif 2.0 and 2.1. When the functionality of an existing routine or widget has changed in Motif 2.0 or 2.1, the reference page explains the differences between the two versions. Volume Six B also provides a complete set of reference material for UIL and Mrm, which was not covered in the previous edition.

Volumes Six A and B are designed to be used together. Volume Six A provides a complete programmer's guide to the Motif toolkit. Each chapter of the book covers a particular component of the Motif toolkit. Each chapter includes basic tutorial material about creating and manipulating the component, intermediate-level information about the configurable aspects of the component, and any advanced programming topics that are relevant. The chapters also provide numerous programming examples.

To get the most out of the examples in Volume Six A, you will need the exact calling sequences of each function from Volume Six B. To understand fully how to use each of the routines described in Volume Six B, all but the most experienced Motif programmers will need the explanations and examples in Volume Six A.

While the Motif toolkit is based on Xt, the focus of this manual is on Motif itself, not on the X Toolkit Intrinsic. Reference pages for the Xt widget classes are included here to provide a complete picture of the widget class hierarchy. Many reference pages mention related Xt routines, but the functionality of these routines is not described. Detailed information about Xt is provided by Volume 4, *X Toolkit Intrinsic Programming Manual, Motif Edition*, and Volume 5, *X Toolkit Intrinsic Reference Manual*.

### How This Manual is Organized

Volume Six B is designed to make it easy and fast to look up virtually any fact about the Motif toolkit. It contains reference pages and numerous helpful appendices.

## Preface

The book is organized as follows:

Preface	Describes the organization of the book and the conventions it follows.
Section 1	<i>Motif Functions and Macros</i> , contains reference pages for all of Motif functions and macros.
Section 2	<i>Motif and Xt Widget Classes</i> , contains reference pages for the widget classes defined by the Motif toolkit and the X Toolkit Intrinsics.
Section 3	<i>Mrm Functions</i> , contains reference pages for the Motif Resource Manager functions that are used in conjunctions with the User Interface Language.
Section 4	<i>Motif Clients</i> , contains reference pages for the Motif clients: <i>mwm</i> , <i>uil</i> , and <i>xmbind</i> .
Section 5	<i>UIL File Format</i> , contains reference pages that describe the file format of a User Interface Language module.
Section 6	<i>UIL Data Types</i> , contains reference pages for the data types supported by the User Interface Language.
Section 7	<i>UIL Functions</i> , contains reference pages for the User Interface Language functions.
Appendix A	<i>Function Summaries</i> , provides quick reference tables that list each Motif function alphabetically and also by functional groups.
Appendix B	<i>Data Types</i> , lists and explains in alphabetical order the structures, enumerated types, and other typedefs used for arguments to Motif and Mrm functions.
Appendix C	<i>Table of Motif Resources</i> , lists all of the resources provided by Motif and Xt widget classes, along with their types and the classes that define them.
Appendix D	<i>Table of UIL Objects</i> , lists all of the objects supported by the User Interface Language, along with their corresponding Motif widget classes.
Appendix E	<i>New Features in Motif 1.2</i> , lists the new functions, widget classes, and widget resources in Motif 1.2.
Index	Should help you to find what you need to know.

## Preface

### Assumptions

This book assumes that the reader is familiar with the C programming language and the concepts and architecture of the X Toolkit, which are presented in Volume 4, *X Toolkit Intrinsic Programming Manual, Motif Edition*, and Volume 5, *X Toolkit Intrinsic Reference Manual*. A basic understanding of the X Window System is also useful. For some advanced topics, the reader may need to consult Volume 1, *Xlib Programming Manual*, and Volume 2, *Xlib Reference Manual*.

### Related Documents

The following books on the X Window System are available from O'Reilly & Associates, Inc.:

Volume Zero	<i>X Protocol Reference Manual</i>
Volume One	<i>Xlib Programming Manual</i>
Volume Two	<i>Xlib Reference Manual</i>
Volume Three	<i>X Window System User's Guide, Motif Edition</i>
Volume Four <i>Edition</i>	<i>X Toolkit Intrinsic Programming Manual, Motif Edition</i>
Volume Five	<i>X Toolkit Intrinsic Reference Manual</i>
Volume Six A	<i>Motif Programming Manual</i>
Volume Seven including reference volume.	<i>XView Programming Manual</i> with accompanying reference volume.
Volume Eight	<i>X Window System Administrator's Guide</i>
	<i>PHIGS Programming Manual</i>
	<i>PHIGS Reference Manual</i>
	<i>PEXlib Programming Manual</i>
	<i>PEXlib Reference Manual</i>
Quick Reference	<i>The X Window System in a Nutshell</i>
Programming Supplement for Release 6 of the X Window System	

### Conventions Used in This Book

*Italic* is used for:

## Preface

- UNIX pathnames, filenames, program names, user command names, options for user commands, and variable expressions in syntax sections.
- New terms where they are defined.

Constant width Font is used for:

- Anything that would be typed verbatim into code, such as examples of source code and text on the screen.
- Variables, data structures (and fields), symbols (defined constants and bit flags), functions, macros, and a general assortment of anything relating to the C programming language.
- All functions relating to Motif, Xt, and Xlib.
- Names of subroutines in example programs.

*Constant Width Italic* Font is used for:

- Arguments to functions, since they could be typed in code as shown but are arbitrary names that could be changed.

*Helvetica Italic* is used for:

- Titles of examples, figures, and tables.

**Boldface** is used for:

- Chapter headings, section headings, and the names of buttons and menus.

## We'd Like to Hear From You

We have tested and verified all of the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing:

O'Reilly & Associates, Inc.  
103 Morris Street, Suite A  
Sebastopol, CA 95472  
1-800-998-9938 (in the US or Canada)  
1-707-829-0515 (international/local)  
1-707-829-0104 (FAX)

## Preface

### Acknowledgements

This book developed out of the realization that it would be impossible to update the first edition of Volume Six to cover Motif 1.2 without dividing the original book into two books. Dan Heller, David Flanagan, Adrian Nye, and Tim O'Reilly all provided valuable suggestions on how best to expand the original reference appendices into a full-fledged reference manual.

The Motif reference pages in this book are based on the reference appendices from the first edition, which were developed by Daniel Gilly. His work meant that I didn't have to start from scratch, and thus saved many hours of toil. The OSF/Motif reference material also provided a helpful foundation from which to explore the complexities of the Motif toolkit. Many of the Motif examples in the book were borrowed from the first edition of Volume Six. These examples were written by Dan Heller, although they have been updated for Motif 1.2.

Dave Brennan, of HaL Computer Systems, took on the unenviable task of learning everything there is to know about UIL and Mrm, so that he could write the UIL reference material. He did a great job.

Adrian Nye deserves special recognition for freeing me to work on this project, when I'm sure that he had other projects he would have liked to send my way. I don't think either one of us had any idea how involved this update project would become. The other inhabitants of the "writer's block" at O'Reilly & Associates, Valerie Quercia, Linda Mui, and Ellie Cutler, provided support that kept me sane while I was working on the book. Extra gratitude goes to Linda Mui for her work on the cross references and the reference tables; her knowledge of various tools prevented me from doing things the hard way. Tim O'Reilly also provided editorial support that improved the quality of the reference material.

Special thanks go to the people who worked on the production of this book. The final form of this book is the work of the staff at O'Reilly & Associates. The authors would like to thank Chris Reilly for the figures, Ellie Cutler for indexing, Lenny Muellner for tools support, Eileen Kramer for copy editing and production of the final copy, and Clairemarie Fisher O'Leary for final proofing and printing. Thanks also to Donna Woonteiler for her patience in answering my questions and helping me to understand the production process.

Despite the efforts of all of these people, the authors alone are responsible for any errors or omissions that remain.

*Paula M. Ferguson*



## **Preface**

### **Acknowledgements to the Motif 2.1 Edition**

Many thanks to all at IST who gave me the time and opportunity to perform this work. I would like to thank all those who reviewed the material, which in a Reference Manual of this type is a tedious but necessary task: a very big "Thank You" to Andy Bartlett who took the trouble of sitting down with the Motif sources whilst pouring over every technical detail, and to Tricia Lovell who reviewed the format at particularly short notice.

A special thanks also to Richard Offer and Doug Rand from Silicon Graphics, and Mark Riches for casting expert and independent eyes over the materials. I would also like to thank Andy Lovell and Derek Lambert for allowing and freeing me up to perform the task. To the rest of the company, who have had to wait whilst yet another batch of print jobs ran to completion, all I can say is "Sorry".

A very big "Thank You" indeed to all at O'Reilly for allowing me to undertake this important task, and especially to Paula Ferguson, my editor: I could not have done this without you.

But to my wife Emma, who put up with some seriously late nights over a long period, goes the biggest "Thank You" of all. This would not have happened without any of you, and I am extremely grateful.

*Antony J. Fountain*

### **Acknowledgements to the Open Source Edition**

Again, many thanks to all at IST who helped me convert the original troff to Frame and PDF formats. A special thank you to Denise Huxtable who enlightened me on the mysteries of Reference Pages, Indexes, and Tables of Contents. Denise also performed much of the cross-referencing in the manual. Thank you also to Ruth Lambert, who showed me how to mark up the document sources.

Again, a very big "Thank You" to all at O'Reilly, and Paula Ferguson in particular, for helping this open source edition come about.

And again, to my wife Emma: a big kiss, and I'll be home real soon now.

*Antony J. Fountain*

## Section 1 - Motif Functions and Macros

This page describes the format and contents of each reference page in Section 1, which covers the Motif functions and macros.

### Name

Function – a brief description of the function.

### Synopsis

This section shows the signature of the function: the names and types of the arguments, and the type of the return value. If header file other than `<Xm/Xm.h>` is needed to declare the function, it is shown in this section as well.

#### Inputs

This subsection describes each of the function arguments that pass information to the function.

#### Outputs

This subsection describes any of the function arguments that are used to return information from the function. These arguments are always of some pointer type, so you should use the C address-of operator (&) to pass the address of the variable in which the function will store the return value. The names of these arguments are sometimes suffixed with `_return` to indicate that values are returned in them. Some arguments both supply and return a value; they will be listed in this section and in the "Inputs" section above. Finally, note that because the list of function arguments is broken into "Input" and "Output" sections, they do not always appear in the same order that they are passed to the function. See the function signature for the actual calling order.

#### Returns

This subsection explains the return value of the function, if any.

### Availability

This section appears for functions that were added in Motif 2.0 and later, and also for functions that are now superseded by other, preferred, functions.

### Description

This section explains what the function does and describes its arguments and return value. If you've used the function before and are just looking for a refresher, this section and the synopsis above should be all you need.

### Usage

This section appears for most functions and provides less formal information about the function: when and how you might want to use it, things to watch out for, and related functions that you might want to consider.

**Example**

This section appears for some of the most commonly used Motif functions, and provides an example of their use.

**Structures**

This section shows the definition of any structures, enumerated types, typedefs, or symbolic constants used by the function.

**Procedures**

This section shows the syntax of any prototype procedures used by the function.

**See Also**

This section refers you to related functions, widget classes, and clients. The numbers in parentheses following each reference refer to the sections of this book in which they are found.