

Table of Contents

	Page
Preface	xxvii
Summary of Contents	xxvii
Assumptions	xxviii
Related Documents	xxviii
How to Use This Manual	xxix
Font Conventions Used in This Manual	xxxii
Request for Comments	xxxiii
Bulk Sales Information	xxxiii
Obtaining the X Window System Software	xxxiii
Obtaining Motif	xxxiv
Obtaining the Example Programs	xxxiv
FTP	xxxiv
FTPMAIL	xxxv
BITFTP	xxxv
UUCP	xxxvi
Compiling the Example Programs	xxxvii
Acknowledgments	xxxvii
Chapter 1 Introduction to the X Window System	3
1.1 The Server and Client	6
1.2 The Software Hierarchy	8
1.3 Event-driven Programming	10
1.4 The Window Manager	11
1.5 Extensions to X	12
Chapter 2 Introduction to the X Toolkit and Motif	15
2.1 Programming with Widgets	15
2.1.1 About Widget Sets	18
2.1.2 Widget Classes and Instances	20
2.1.3 Widget Configurability with Resources	23
2.1.4 Widget Independence	25
2.1.5 Widget-Application Interaction	27
2.1.6 Xt and Object-oriented Programming (OOP)	29
2.1.6.1 The Object	29

2.1.6.2	Methods	30
2.1.6.3	Messages	30
2.1.6.4	Encapsulation	30
2.2	Structure of Motif Applications	31
2.3	A Simple X Toolkit Application	32
2.3.1	The Code	33
2.3.2	Compiling the Application	36
2.3.3	The App-defaults File	36
2.3.4	To Hardcode or Not to Hardcode	40
2.4	Connecting Widgets to Application Code	41
2.4.1	Callbacks	42
2.5	More About Resources	45
2.5.1	Setting and Getting Resources from the Application	46
2.5.1.1	Setting Resources with the Varargs Interfaces	47
2.5.1.2	Setting Resources with the ArgList Interfaces	49
2.5.1.3	Getting a Resource Value	50
2.5.2	Core Resources	51
2.5.3	Other Inherited Resources	54
2.6	Advice on X Programming	57
2.7	Debugging Xt Applications	58
 Chapter 3 More Techniques for Using Widgets		61
3.1	Using Composite Widgets	61
3.1.1	Setting Resources for an Instance Hierarchy	64
3.1.2	Geometry Management in Practice	65
3.2	Using Constraint Widgets	69
3.3	The Standard Motif Instance Hierarchy	72
3.3.1	Motif Widget Creation Routines	74
3.3.2	Building a Main Window	74
3.4	Using Popups	77
3.4.1	Creating a Basic Menu	77
3.4.2	Creating a Basic Dialog Box	80
3.4.3	Popup Window Hierarchy	84
3.5	More About Callbacks	85
3.5.1	Passing Data to Callback Functions	86
3.5.1.1	The client_data Argument	86
3.5.1.2	The call_data Argument	88
3.5.2	Callback Lists	89
3.6	Application Resources	90
3.6.1	The Application Data Structure	91
3.6.2	The Resource List	91
3.6.3	Getting the Resources	95
3.7	Command-line Options	96
3.7.1	Standard Command-line Options	97
3.7.2	Defining Your Own Command-line Options	98
3.8	Preventing User Customization of Widget Resources	103

3.8.1 Using the Varargs Interfaces	103
3.8.2 Using the Argument List Interfaces	105
3.8.2.1 Another Way to Set Arguments	107
3.8.2.2 Merging Argument Lists	108
3.9 More About Application Contexts	109

Chapter 4 An Example Application 113

4.1 xbitmap1: Bitmap Editor Using a BitmapEdit Widget	114
4.1.1 XBitmap1 App-defaults File	119
4.2 xbitmap2: Adding Graphics to Display the Bitmap	119
4.2.1 Exposure Strategy	121
4.2.2 Graphics from the Application	124
4.2.3 Writing a Bitmap File	128
4.3 xbitmap3: Another Way to Create a Custom Window	129
4.3.1 Actions	129
4.3.1.1 The Actions Table	131
4.3.1.2 Format of an Action Function	132
4.3.2 Translations	133
4.3.2.1 The Translation Table	134
4.3.2.2 Hardcoding Translations	136
4.3.2.3 Action Parameters	138
4.3.3 Adding Actions to Existing Widgets	138
4.4 xbitmap4: A Bitmap Editor Without a BitmapEdit Widget	138

Chapter 5 More About Motif 151

5.1 The Remaining Motif Widgets and Gadgets	152
5.1.1 Geometry Managing Widgets	157
5.1.2 Dialog Widgets	158
5.2 Widget Creation Functions	159
5.3 Compound Strings	162
5.3.1 Simple Compound Strings	162
5.3.2 Strings with Multiple Fonts	164
5.3.3 Manipulating Compound Strings	166
5.3.4 Converting Compound Strings to Text	167
5.3.5 Rendering Compound Strings	168
5.4 Pixmap and Image Caching Functions	168
5.5 Dynamic Resource Defaulting	170
5.6 Resolution Independence	170
5.7 Keyboard Traversal and Focus	171
5.8 Motif Virtual Keyboard Bindings	173
5.9 Drag and Drop	174
5.9.1 Drag Protocol	174
5.9.2 Drop Protocol	175
5.9.3 Operations	175

5.9.4 Application Programming Interface	176
5.9.5 Summary: Drag and Drop in Applications	176
5.10 Tear-off Menus	177
5.11 The Motif User Interface Language	177

Chapter 6 Inside a Widget 185

6.1 Widget Source File Organization	186
6.2 The Private Header File—BitmapEdiP.h	187
6.2.1 Parts and Records	187
6.2.2 Class Part and Class Record	188
6.2.3 Instance Part and Instance Record	189
6.3 The Widget Implementation File—BitmapEdit.c	191
6.3.1 Obligatory Include Files	192
6.3.2 Defining the Resource List	193
6.3.3 The Translation Table and Actions Table	196
6.3.4 Declaring Methods	198
6.3.5 Initializing the Class Record	199
6.3.5.1 The Core Class Part	199
6.3.5.2 Initializing the Core Methods	202
6.3.6 Description of Core Methods	204
6.3.7 Packaging the Class Record for Application Use	206
6.3.8 A Sample Method	206
6.4 The Public Header File—BitmapEdit.h	209
6.5 The Process of Widget Writing	211
6.6 Summary of Conventions	212

Chapter 7 Basic Widget Methods 217

7.1 The X Graphics Model Inside Widgets	218
7.2 The initialize Method	219
7.2.1 Creating GCs	221
7.3 The expose Method	224
7.4 The set_values Method	227
7.5 The resize Method	230
7.6 The query_geometry Method	233
7.7 The destroy Method	236
7.8 Actions in the Widget Framework	237

Chapter 8 Events, Translations, and Accelerators 243

8.1 Translation Table Syntax	244
8.1.1 The Directive	245
8.1.2 Selecting the Events to Translate	245
8.1.3 Details in Keyboard Events	248

8.1.4	Details in Other Event Types	249
8.1.5	Modifiers	251
8.1.5.1	Physical Keys Used as Modifiers	252
8.1.5.2	Default Interpretation of the Modifier List	254
8.1.5.3	Prohibiting a Modifier	254
8.1.5.4	Requiring an Exact Match	255
8.1.5.5	Paying Attention to the Case of Keysyms	255
8.1.6	Event Sequences	256
8.1.6.1	Special Considerations Involving Motion Events	257
8.1.6.2	Modifiers and Event Sequences	258
8.1.6.3	Using Modifiers to Specify Button Event Sequences	258
8.1.6.4	Key Event Sequences	258
8.1.7	Interactions Between Translations	259
8.1.7.1	Translations in Multiple Resource Files	259
8.1.7.2	Order of Translations	259
8.1.7.3	Event Sequences Sharing Initial Events	260
8.1.7.4	Event Sequences Sharing Noninitial Events	260
8.2	Accelerators and Mnemonics	261
8.2.1	Xt Accelerators	261
8.2.2	Event Propagation	263
8.2.3	Installing Accelerators in Multiple Widgets	265
8.2.4	Defining the Accelerator Table in the Code	266
8.2.5	Motif Accelerators and Mnemonics	267
8.2.5.1	Mnemonics	267
8.2.6	The <code>display_accelerators</code> Method	268
 Chapter 9 More Input Techniques		271
9.1	Event Handlers	272
9.1.1	Adding Event Handlers	275
9.1.2	Adding Nonmaskable Event Handlers	276
9.1.3	Removing Event Handlers	277
9.1.4	Adding Raw Event Handlers	278
9.2	Writing Routines That Use Specific Event Data	278
9.2.1	Event Types and Structure Names	280
9.3	File, Pipe, and Socket Input	282
9.3.1	Getting File Input	282
9.3.2	Getting Pipe Input	284
9.4	Timeouts	285
9.5	Visibility Interest	288
9.6	Work Procedures	290
9.7	Low-level Management of the Event Queue	292
9.7.1	<code>XtPending</code> and <code>XtPeekEvent</code>	292
9.7.2	Event Filters	293
9.7.3	Input Sensitivity	294

Chapter 10 Resource Management and Type Conversion	297
10.1 Review of Resource Fundamentals	298
10.2 How Xt's Resource Manager Works	301
10.2.1 Basic Syntax of Resource Specifications	302
10.2.2 Wildcarding Resource Component Names	304
10.2.3 Merging of Resource Files	304
10.2.4 Syntax of Environment Variables	307
10.2.5 Including Files in a Resource File	308
10.2.6 The Language String	309
10.2.7 Screen-specific Resource Strings and Databases	310
10.2.8 Fallback Resources	312
10.2.9 Resource Matching Algorithm	313
10.2.10 Resource Precedence Rules	315
10.2.11 Customized Resource Files	319
10.2.12 The XtNbaseTranslations Resource	321
10.3 Type Conversion	322
10.3.1 Conversions from XmRString	322
10.3.2 Other Built-in Type Conversions	324
10.3.3 Special Resource Defaults That Do Not Use Conversion	325
10.3.4 Registering Type Converters	327
10.3.4.1 Passing Arguments to a Type Converter	330
10.3.5 Explicitly Invoking a Converter	332
10.3.6 Writing a Type Converter	334
10.3.6.1 Defining the Default Value	335
10.4 Subparts and Subresources	336
10.4.1 The Hook Methods	337
10.4.2 Managing Subresources	338
Chapter 11 Interclient Communications	341
11.1 Window Manager Interactions	342
11.1.1 Shell Subclasses	343
11.1.2 Setting Shell Resources	343
11.1.3 Screen Space	346
11.1.4 Input Model	347
11.1.5 Colormaps	349
11.1.6 Icons	350
11.1.7 Window Manager Decorations	352
11.1.8 Interacting With the Motif Window Manager	352
11.1.8.1 WM_PROTOCOLS	353
11.1.8.2 _MOTIF_WM_MESSAGES	354
11.1.8.3 _MOTIF_WM_HINTS	354
11.2 Selections: Widget-to-Widget Communication	354
11.2.1 How Atomic Selection Works	356
11.2.2 Highlighting the Selected Data (Owner)	359
11.2.3 Making the Selection with XtOwnSelection (Owner)	364

11.2.4	Requesting the Selection (Requestor)	365
11.2.4.1	Possible Target Type Atoms	366
11.2.4.2	The Paste Action from BitmapEdit	368
11.2.5	Converting the Selection (Owner)	368
11.2.6	Finally Pasting the Selection (Requestor)	370
11.2.7	If the Selection is Lost (Owner)	372
11.2.8	When the Selection Transfer is Complete (Owner)	373
11.2.9	ICCCM Compliance	373
11.2.9.1	Xmu Atom Caching	375
11.2.9.2	Converting the Standard Selections	375
11.2.10	How Incremental Selection Works	377
11.2.11	Miscellaneous Selection Routines	379
11.3	Motif Cut and Paste Functions and the Clipboard	379

Chapter 12 Geometry Management 383

12.1	How Composite Management Works	384
12.1.1	Initial Geometry Negotiation	386
12.1.2	User Resizes the Application	389
12.1.3	Widget Desires a Size Change	389
12.1.4	Application Resizes a Widget	391
12.2	Writing a Composite Widget	392
12.2.1	Basic Core Methods in Composite Widgets	394
12.2.2	Laying Out Child Widgets	396
12.2.3	The change_managed Method	398
12.2.4	The geometry_manager Method	398
12.2.5	XtQueryGeometry and the query_geometry Method	399
12.2.6	XtMakeGeometryRequest and the geometry_manager Method	400
12.2.7	The set_values_almost Method	402
12.2.8	The insert_child and delete_child Methods	403
12.3	How Constraint Management Works	404
12.4	Writing a Constraint Widget	404
12.4.1	The Core Resource List	405
12.4.2	The Constraint Resource List	406
12.4.3	Class Structure Initialization	408
12.4.4	The Constraint initialize Method	410
12.4.5	The class_part_init Method	411
12.4.6	The geometry_manager Method	412
12.4.7	The resize Method	416
12.4.8	The Core and Constraint set_values Methods	418
12.4.9	The change_managed Method	418
12.4.10	The query_geometry Method	419
12.4.11	Delaying Geometry Recalculation	420
12.5	Compound Widgets	420
12.6	Stacking Order	421

Chapter 13 Menus, Gadgets, and Cascaded Popups	425
13.1 Menu Styles and Implementation	427
13.2 Using Motif Menus	428
13.2.1 Popup Menu	429
13.2.2 Cascaded Menus	430
13.3 Basic Xt Popup Support	431
13.3.1 A Spring-loaded Menu: Pointer Grabbing	432
13.3.2 A Pulldown Menu	440
13.3.3 Cascaded Menus	443
13.3.4 Using the R4 SimpleMenu Widget	448
13.3.5 Delayed Popup Creation	452
13.4 About Dialog Boxes	453
13.5 Gadgets	454
13.5.1 Inside a Gadget	457
13.5.2 Private Header File	458
13.5.3 The Gadget Source File	459
13.5.4 The Public Header File	460
13.5.5 The Gadget Parent	460
Chapter 14 Miscellaneous Toolkit Programming Techniques	467
14.1 Errors and Warnings	467
14.2 Objects	470
14.3 Macros For Getting Information	471
14.4 The accept_focus Method and the Keyboard Focus	473
14.5 Keyboard Interpretation	474
14.6 Memory Allocation	475
14.7 Action Hooks and Calling Actions Directly	476
14.8 Xt Grabbing Functions	476
14.9 File Finding and Internationalization	477
14.10 Application Contexts	477
14.10.1 Multiple Application Contexts	479
14.10.2 Rewriting XtAppMainLoop for Multiple Application Contexts	479
14.10.3 Functions Used with Multiple Application Contexts	480
14.11 Multiple Top-level Shells	480
14.12 Connecting to Multiple Servers	481
14.13 Class Extension Structures	481
14.14 Using Editres in Xt Programming	483
14.14.1 A Tour of editres	483
14.14.2 editres as a Programmer's Tool	490
14.14.3 Adding editres Support to an Xt Application	490
14.14.4 Using editres to Understand the Structure of an Application	491
14.14.5 Using editres to Test or Debug a Widget	492
14.14.5.1 Testing a Widget's set_values Procedures	492
14.14.5.2 Testing a Widget's geometry_manager	493
14.15 Internationalization in the X Toolkit	494

14.15.1 String Encoding and Locale Dependencies in Xt	494
14.15.2 Establishing Locale in an Xt Application	494
14.15.3 XFontSet Resources	496
14.15.4 Other Xt Changes for Internationalization	496
14.15.5 Internationalization in Motif 1.2	497
14.15.6 Text Output	497
14.15.7 Text Input	498
14.15.8 Compound Text Conversion	498
14.15.9 <code>_XmGetLocalized()</code>	498
14.15.10 Summary	498

Appendix A Athena, OPEN LOOK, and Motif 503

A.1 The AT&T OPEN LOOK Widgets	507
A.1.1 Application Controls	508
A.1.1.1 Command Buttons	508
A.1.1.2 Exclusive and Nonexclusive Settings	510
A.1.1.3 Analog Controls	512
A.1.2 Composite Widgets	512
A.1.2.1 Menus and Control Areas	512
A.1.2.2 General Purpose Composite Widgets	513
A.1.2.3 Scrollbars and Scrollable Windows	513
A.1.3 Popups	515
A.1.4 Text Widgets	517
A.1.5 Drawing Areas	517
A.2 The OSF/Motif Widgets	517
A.2.1 Application Controls	520
A.2.1.1 Command Buttons	520
A.2.1.2 Analog Controls	521
A.2.2 Composite Widgets	521
A.2.2.1 Menus and Control Areas	521
A.2.2.2 General Purpose Composite Widgets	523
A.2.2.3 Scrollable Windows	524
A.2.3 Popups	526
A.2.4 Text Widgets	526
A.2.5 Drawing Areas	527

Appendix B Specifying Fonts and Colors 531

B.1 Color Specification	531
B.1.1 Color Names	531
B.1.2 Hexadecimal Color Specification	532
B.1.2.1 The RGB Color Model	533
B.1.2.2 How Many Colors are Available?	534
B.2 Font Specification	536
B.2.1 Font Naming Conventions	537

B.2.2	Font Name Wildcarding	538
B.2.3	Font Name Aliasing	540
B.2.4	Making the Server Aware of Aliases	541
B.2.5	The fonts.dir Files	542
B.3	Font Service	542
B.4	Scalable Fonts	544
B.4.1	Finding Scalable Fonts	545
B.4.2	Finding Derived Instances of Scalable Fonts	546
B.4.3	Using Scalable Fonts	548
B.5	Window Geometry	550

Appendix C Naming Conventions 555

Appendix D Release Notes 559

D.1	R4 to R5	559
D.2	Motif 1.1 to 1.2	560
D.2.1	Widget Changes	561
D.2.2	New Display and Screen Objects	562
D.2.3	Default Colors	562
D.2.4	Insensitive Visuals	563
D.2.5	Baseline Alignment	563
D.2.6	Geometry Management	563
D.2.7	Traversal	563
D.2.8	Representation Type Convenience Functions	563
D.2.9	caddr_t Changed to XtPointer	564
D.2.10	Mwm	564
D.2.11	UIL and Mrm	564

Appendix E The xbitmap Application 567

E.1	The BitmapEdit Widget	567
E.2	The BitmapEdiP.h Private Header File	580
E.3	The BitmapEdit.h Public Header File	582
E.4	xbitmap5	583

Appendix F Sources of Additional Information 593

F.1	Getting the X Software	593
F.1.1	Bug Fixes	594
F.1.1.1	Notes	596
F.1.1.2	Fairness	596

F.2 Netnews	597
F.3 Training, Consulting, and Support	597
F.4 The X Consortium	598
F.5 Finding Out for Yourself	599

Glossary	603
-----------------------	-----

Master Index	625
---------------------------	-----