
Contents

(Garfinkel/Mahoney book)

1

Introduction to the NeXTSTEP Graphical User Interface 1

What Makes NeXTSTEP So Special?	1
An Overview of the Graphical User Interface	2
Basic Principles of the NeXTSTEP User Interface	4
The Mouse	5
<i>Mouse Action Paradigms</i>	6
Cursors	7
Window Types	8
<i>Standard Windows</i>	8
<i>Panels</i>	9
<i>Menus</i>	11
<i>Pop-Up Lists</i>	11
<i>Pull-Down Lists</i>	12
<i>Minowindows</i>	12
<i>Freestanding and Docked Icons</i>	13
<i>Main and Key Windows</i>	14
<i>Window Order</i>	15
Controls	16
<i>Buttons</i>	17
<i>Menu Commands</i>	17
<i>Text Fields</i>	17
<i>Sliders and Scrollers</i>	17
<i>Browsers and Selection Lists</i>	18
<i>Color Wells</i>	18
The Workspace Manager Application	19
<i>The File Viewer</i>	19
<i>The Workspace Menu Structure</i>	21
<i>Workspace Manager Panels</i>	23
Configuring Your Workspace, Step by Step	23
Menu Guidelines and Keyboard Alternatives	29
<i>Main Menu</i>	29

<i>Info Submenu</i>	30
<i>Document (or File) Submenu</i>	30
<i>Edit Submenu</i>	31
<i>Find Submenu</i>	31
<i>Format Submenu</i>	31
<i>Font Submenu</i>	32
<i>Windows Submenu</i>	32
<i>Services Submenu</i>	33
Working with the File System, Step by Step	33
New Features in the NeXTSTEP 3.0 Interface	43
<i>Workspace Manager 3.0 Menus: New Commands</i>	43
<i>Workspace Manager 3.0 Panels: New Features</i>	44
<i>NextApps, NextDeveloper/Apps Folders in NeXTSTEP 3.0</i>	44
<i>Preferences Application: New Features</i>	46
Summary	46

2

NeXTSTEP Development Tools 47

The Terminal Application	48
Edit – NeXTSTEP’s Mouse-Based Editor	51
Using Edit and Librarian, Step by Step	52
HeaderView	58
Creating Icons	59
<i>Creating an Icon with IconBuilder in NeXTSTEP 3.0</i>	59
<i>Creating an Icon with the Icon Application in NeXTSTEP 2.1</i>	61
<i>Using an Icon to Represent a Directory in the File Viewer</i>	62
make – The UNIX Program Building Utility	62
gdb – Debugging Programs	63
<i>gdb Commands</i>	64
<i>gdb’s “view” Command</i>	64
<i>gdb’s “where” Command</i>	64
<i>gdb’s “print” Command</i>	65
<i>Using gdb in NeXTSTEP 3.0</i>	66
User Interface Design	67
Summary	68

3

Creating a Simple Application with Interface Builder 69

- Getting Started with Interface Builder 71
- Adding Objects to Your Application 76
 - Adding a Button Object to Your Window* 76
 - Giving Your Button a Funky Sound* 79
 - Giving Your Button an Image* 79
- Objects, Messages, and Targets 80
 - Objects and Classes* 80
 - Targets, Actions, and Connections* 82
- Congratulations! 85

4

Creating an Application Without Interface Builder 87

- The Tiny.m Program 87
- Tiny.m: line-by-line 91
- Classes and Objects, Methods and Messages 93
 - The Circle Class Example* 95
 - Creating and Destroying Objects* 96
 - Class Interface Definitions* 97
- The NeXTSTEP Classes Used by Tiny.m 99
 - Inheritance* 100
- Back to Tiny.m 101
 - Windows, Menus, and the demo() Function* 103
 - Views* 107
 - Drawing with PostScript Inside a View Object* 108
- Responding to Events 110
- An Introduction to Event Handling 111
 - Tiny2: A Demonstration of Event Handling* 111
- NeXTSTEP Capitalization Conventions 117
 - General Rules* 117
 - Objective-C Class Names* 118
 - Objective-C Methods* 118
 - UNIX Variables and Functions* 118
- Summary 118

5

Building a Project: A Four Function Calculator 121

Getting Started Building the Calculator Project	122
<i>Building a Project in NeXTSTEP 2.1</i>	124
Building the Calculator's User Interface	126
<i>Customizing the Main Window</i>	126
<i>Adding Controls in a Window</i>	128
<i>Matrix Dragging Options</i>	130
Building the Calculator's Controller Class	131
<i>Designing the Controller Class</i>	131
<i>Creating the Controller Class</i>	132
<i>Outlets and Connections</i>	133
<i>Adding Outlets to an Object</i>	134
<i>Making the Connection</i>	134
<i>Adding Actions to the Controller</i>	136
<i>Generating the Controller Class Files</i>	137
<i>Adding Code to Make the Controller Class Work</i>	138
Customizing Buttons and Making Connections	142
<i>Instantiating (Creating an Instance of) the Controller Class</i>	142
<i>Setting Up Titles and Tags for the Keypad Buttons</i>	143
<i>Making the Connections</i>	146
Compiling and Running a Program	149
<i>Compiling and Running a Program from Project Builder</i>	149
<i>Compiling and Running a Program in a Terminal Shell Window</i>	150
<i>Compiler Error Messages</i>	151
Tags and the enterDigit: Action Method	153
Adding the Four Calculator Functions	154
Adding Unary Minus to the Controller Class	156
<i>Using IB's Parse Command with a New Action Method</i>	156
The Files in a Project	159
<i>The Project Builder-generated Makefile</i>	161
<i>The PB-generated main() Program File</i>	164
<i>Other PB-generated Files</i>	166
Summary	166

6

Nibs and Icons 167

- Managing Multiple Nibs 167
 - Modifying the Controller Class* 168
 - Modifying the Main Calculator Nib* 171
 - Creating the Info Panel's Nib in NeXTSTEP 3.0* 174
 - Creating the Info Panel's Nib in NeXTSTEP 2.1* 179
- Adding Icons to Your Application 182
 - Creating Icons* 182
- Changing the Calculator's Application Icon 184
 - Changing an Application Icon in NeXTSTEP 2.1* 187
- NeXTSTEP Images 189
- Adding Other Images to Your Application 189
- Summary 191

7

Delegation and Resizing 193

- Handling Different Bases 193
 - Modifying the Controller Class* 195
- Delegation 198
 - Did and Will Delegate Methods* 199
 - Specifying an Object's Delegate* 201
 - Setting Up a Delegate Outlet in the Nib* 202
 - Adding the Delegate Method* 203
- Disabling Buttons for Better Multi-Radix Input 205
 - Using a List Object to Directly Access Matrix Cells* 205
 - Coherence in Object Oriented Programming* 208
- Resizing Windows Programmatically 209
 - Modifying the Calculator's Interface* 209
 - Modifying the Controller Class* 211
- The Window and View Classes 215
 - The Window Class* 215
 - The View Class* 217
- Strong Typing with Objective-C 218
- Summary 220

8

Events and Responders 221

- Events and the Responder Chain 221
 - What Is An Event?* 222
 - The NXEvent Data Structure* 222
- Events and the Application Object 225
 - Responders and the Responder Chain* 226
 - Mouse Event Handling* 227
 - Keyboard Event Handling* 229
 - An Event Handling Example* 230
 - Action Messages and the Responder Chain* 231
 - Other Kinds of Events* 234
- The Event Loop 234
 - The Application Event Loop* 235
- Catching Keyboard Events for the Calculator 236
 - Subclassing the Window Class* 237
 - Hash Tables* 238
 - Searching for Button Titles in the Calculator Window* 239
 - Finishing Off the CalcWindow Class Implementation* 242
 - Changes in the Controller Class* 245
- Summary 246

9

Mach and the Window Server 247

- The Mach Operating System 247
 - The Microprocessor* 248
 - The Mach Microkernel* 248
 - The Mach System Support Environment* 249
 - Username and UIDs* 250
 - Processes, PIDs, and UIDs* 250
- Mach-O: The Mach Object File Format 251
- The Window Server and Display PostScript 252
 - PostScript* 253
 - The Application Kit and the Window Server* 253
- Seeing All the Processes 255

10 *MathPaper and Multiple Windows* 259

- MathPaper 259
 - The Structure of MathPaper* 260
- The Evaluator Back End 261
 - Lex and Yacc* 263
- Building the Back End 263
- Building the MathPaper Front End 268
 - Setting up the Main Nib* 269
 - Setting Up the paperwindow Nib* 273
 - Setting Up the paperwindow.nib Connections* 278
 - Window Delegates* 280
 - Controlling the MathPaper Application* 282
 - Controlling a Piece of MathPaper* 284
 - Testing MathPaper* 286
- Summary 286

11 *Spawning Multiple Processes and the Text Object* 287

- Spawning Processes with the Process Class 287
- The ScrollView and Text Classes 292
 - Text Class Basics* 294
- Changes to the PaperControl Class 295
 - Spawning the Subprocesses* 295
 - Displaying Data Received From the Process* 297
 - Outlet Setting Methods* 299
 - Text Delegates and Field Editors* 300
 - Getting Out of Hand* 305
- Summary 306

12 *Text and Rich Text* 307

- What Is Rich Text? 308
 - Manipulating a Text Object with Explicit Commands* 309

<i>Rich Text Format (RTF)</i>	310
<i>Exploring Rich Text Commands</i>	311
<i>Understanding the Rich Text Format</i>	313
A Rich Text Object	316
Integrating the Rich Text Object with MathPaper	320
Summary	321

13 *Saving, Loading, and Printing* 323

Saving to a File	324
<i>File Name Extensions</i>	324
<i>Adding a Document File Icon and Extension to MathPaper</i>	327
<i>Writing the Save Methods</i>	328
<i>Testing the Save Features</i>	332
Loading From a File	335
<i>Exception Handling</i>	338
Marking a Document Window as Edited	340
Terminating an Application Properly	343
Enabling and Disabling Menu Cells	345
<i>Window Updates</i>	346
<i>Making MathPaper Update Its Menus</i>	347
Adding Printing Capability	349
Summary	352

14 *Drawing with Display PostScript* 353

Animation in an Info Panel	353
pswrap: Wrapping PostScript into a C Function	354
<i>Passing Arguments to pswrap Functions</i>	358
<i>Wraps that Return a Value</i>	358
<i>Buffering</i>	359
<i>A PostScript Wrap for MathPaper</i>	359
Drawing in a View with PostScript	361
Adding the Animated Info Panel to MathPaper	362
<i>Adding the Info Panel Menu Cell, Method, and Connection</i>	364
<i>InfoView's Animation Code</i>	365

<i>Testing the Animation</i>	368
Displaying Pictures in a View	368
<i>Compositing</i>	369
<i>The Dissolving Animation</i>	370
Display PostScript Timed Entries	373
<i>Adding a Timed Entry</i>	373
<i>The Timed Entry Handler</i>	374
<i>Removing a Timed Entry</i>	374
<i>Using Timed Entries in Animation</i>	374
Summary	380

15

Draw Yourself: All About NeXTSTEP Views 381

The Advantages of View's drawSelf:: Method	382
BlackView: A View That Paints Itself Black	382
A Closer Look at the View Class	385
<i>View Coordinate Systems</i>	385
<i>Moving and Resizing Views</i>	387
<i>Flipping</i>	388
<i>The View Hierarchy</i>	388
<i>Opaque and Non-Rectangular Views</i>	389
<i>Displaying Views</i>	390
<i>Controlling Redisplay</i>	390
<i>Resizing</i>	391
BarView: A View with a Scaled Coordinate System	392
PolygonView: A Non-Opaque View	395
<i>Changing the PolygonView's Size</i>	398
<i>Placing a View Inside a ScrollView</i>	401
Responding to Events in a View	402
<i>Getting a Mouse-Down Event</i>	403
Autosizing Multiple Views in a Window	404
Summary	406

16

GraphPaper: A Multi-threaded Application with a Display List 407

- GraphPaper's Design 408
 - The Interface* 408
 - Connecting to the Back End* 408
 - Why Use a Display List?* 410
- Working with Multiple Threads 410
 - The Difficulty of Using Threads* 411
 - C Threads* 413
 - Using C Threads* 413
 - Threads and Objective-C* 414
- Building the GraphPaper Application 415
 - Changes to the Evaluator Back End* 415
 - Building GraphPaper's Interface* 416
 - The GraphView Class Interface File* 419
 - The GraphView Class Implementation File* 422
 - The Data Stuffer Methods* 423
 - Stopping a Running Graph* 425
 - The Data Stuffer Function* 425
 - Constructing the Graph* 427
 - The drawSelf:: Method* 431
 - The clear and awakeFromNib Methods* 433
 - The Segment Class* 435
 - The Process Class* 439
 - Testing GraphPaper* 439
- Extending the Display List 441
 - Adding Axes* 441
 - Adding Labeling* 442
 - Using the Label Class* 447
- Summary 448

17

Color 449

- Colors and Color Objects 449
 - Colors From a Programmer's Point of View* 450
 - Colors From a User's Point of View* 452
 - Programming with Colors* 454

Adding Color to GraphPaper	454
<i>Creating a Preferences Panel</i>	455
<i>The Controller Class</i>	456
<i>Creating the Preferences Nib, Panel, and Controller</i>	458
<i>PrefController Class Implementation</i>	459
ColorGraphView	461
<i>Adding Color to GraphPaper</i>	461
<i>Changes to the GraphView Class</i>	463
<i>The ColorGraphView Class Implementation</i>	463
<i>Setting the Colors</i>	464
<i>Setting the Initial Color</i>	466
Changes to the Segment and Label Classes	466
<i>Testing GraphPaper's Color</i>	468
Summary	468

18 *View Resizing and Mouse Tracking 471*

More About Resizing	471
<i>Keeping Your Windows Big Enough</i>	475
Tracking the Mouse	476
<i>Tracking Rectangles</i>	476
<i>The Window Event Mask</i>	478
Adding Mouse Tracking to GraphPaper	479
<i>Changes to the GraphPaper Interface</i>	480
<i>Changes to the TrackingGraphView Class Files</i>	481
Changing the Cursor	486
<i>Adding a Cursor to the Nib</i>	487
<i>Changes to the TrackingGraphView Class</i>	487
Summary	489

19 *Zooming and Saving Graphics Files 491*

Adding a Zoom Button To GraphPaper	491
<i>The ScrollView Class Revisited</i>	493

<i>Changes to GraphPaper.nib</i>	493
<i>Changes to ZoomScrollView</i>	495
<i>Testing the Zoom Button</i>	498
<i>Bug Fixes</i>	499
Saving a View as Encapsulated PostScript	501
<i>Changes to GraphPaper.nib</i>	501
<i>Changes to the Controller Class</i>	502
<i>Testing the Save Image Menu Command</i>	503
Saving a View in Tag Image File Format	504
Creating an Accessory View	506
<i>Changes to the Controller Class Implementation</i>	507
<i>Testing the “EPS or TIFF” Save Feature</i>	509
Summary	509

20 *The Pasteboard and Services 511*

Cut, Copy, and Paste with the Pasteboard	511
<i>Providing Data to the Pasteboard</i>	513
Using the Pasteboard in GraphPaper	515
<i>Providing Data Through Lazy Evaluation</i>	517
<i>Implementing the Cut Command</i>	519
<i>Testing GraphPaper’s Copy and Cut Commands</i>	520
Services	520
<i>How Services Work</i>	522
Creating Your Own Service	524
<i>Adding a Graph Service to GraphPaper</i>	525
<i>Testing GraphPaper’s Service</i>	529
Summary	531

21 *Preferences and Defaults 533*

How Preference Defaults Work	534
<i>Accessing the Information Stored in the Database</i>	534
Adding Defaults to GraphPaper	537
<i>Reading Values from the Defaults Database</i>	538

Making the Preferences Panel Work with Defaults 540*Modifying the Preference Panel 541**Changes to PrefController Class 542**Testing the Updated Preferences Panel 544***Setting Up a Multi-View Panel 545***Multi-View Panels 545**Changes to the Interface 548**Defaults for Initial Values 550***Summary 553**

